

TPC Benchmark™ C
Full Disclosure Report
for
IBM® @server® xSeries® 366
using
DB2® Universal Database 8.2
and
Microsoft® Windows® Server 2003
Enterprise x64 Edition

TPC-C Version 5.3

Submitted for Review
March 28, 2005



First Edition - March 2005

THE INFORMATION CONTAINED IN THIS DOCUMENT IS DISTRIBUTED ON AN AS IS BASIS WITHOUT ANY WARRANTY EITHER EXPRESSED OR IMPLIED. The use of this information or the implementation of any of these techniques is the customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used.

This publication was produced in the United States. IBM may not offer the products, services, or features discussed in this document in other countries, and the information is subject to change without notice. Consult your local IBM representative for information on products and services available in your area.

© Copyright International Business Machines Corporation 2005. All rights reserved.

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

U.S. Government Users - Documentation related to restricted rights: Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Trademarks

IBM, the IBM logo, DB2, xSeries, ServeRAID, eServer and the eServer logo are trademarks or registered trademarks of International Business Machines Corporation.

The following terms used in this publication are trademarks of other companies as follows: TPC Benchmark, tpmC, and \$/tpmC trademark of Transaction Processing Performance Council; Intel and Xeon are trademarks or registered trademarks of Intel Corporation; Microsoft and Windows are trademarks or registered trademarks of Microsoft Corporation. Other company, product, or service names, which may be denoted by two asterisks (**), may be trademarks or service marks of others.

Notes

¹ GHz and MHz only measures microprocessor internal clock speed, not application performance. Many factors affect application performance.

² When referring to hard disk capacity, GB, or gigabyte, means one thousand million bytes. Total user-accessible capacity may be less.

Abstract

IBM Corporation conducted the TPC Benchmark™ C on the IBM® @server® xSeries® 366 configured as a client/server system. This report documents the full disclosure information required by the TPC Benchmark C Standard Specification, Revision 5.3, including the methodology used to achieve the reported results. All testing fully complied with this revision level.

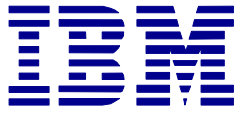
The software used on the xSeries 366 system includes Microsoft® Windows® Server 2003 Enterprise x64 Edition operating system and IBM DB2 Universal Database V8.2 Enterprise Edition database.

Two standard metrics, transactions per minute-C (tpmC) and price per tpmC (\$/tpmC), are reported as required by the TPC Benchmark C Standard Specification.

The benchmark results are summarized in the following table.

| Hardware | Software | Total System Cost | tpmC | \$/tpmC | Total Solution Availability Date |
|----------------------------|---|-------------------|---------|---------------|----------------------------------|
| IBM @server xSeries 366 | DB2 UDB 8.2 Microsoft Windows Server 2003 Enterprise x64 Edition | \$903,089 USD | 150,704 | \$5.99 USD | August 20, 2005 |

The results of the benchmark and test methodology used were audited by Bradley J. Askins of InfoSizing, Inc. The auditor's attestation letter is contained in Section 9 of this report.

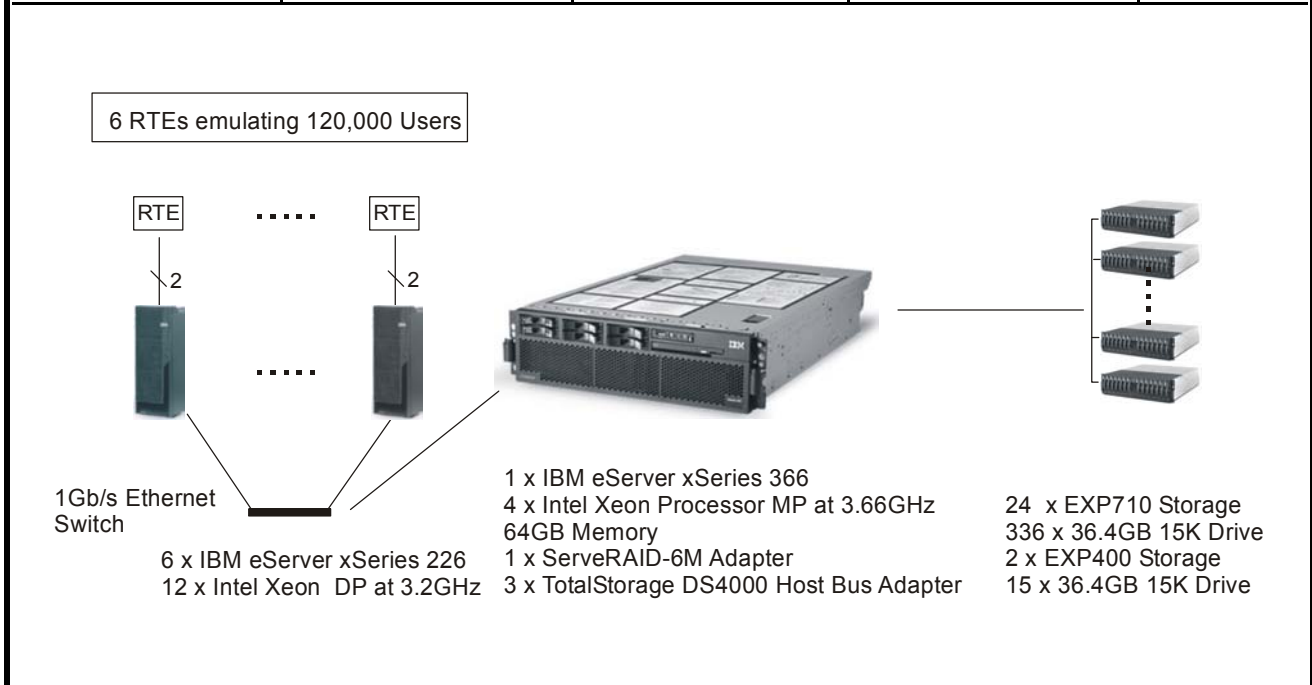


**IBM® eServer® xSeries® 366c/s
and
DB2® UDB 8.2**

TPC-C Rev. 5.3

Report Date: March 28, 2005

| | | | | |
|---|--------------------|--|--|----------------------|
| Total System Cost | | TPC-C Throughput | Price/Performance | Availability Date |
| \$903,089 USD | | 150,704 tpmC | \$5.99 USD / tpmC | Aug. 20, 2005 |
| Processors | Database Manager | Operating System | Other Software | Number of Users |
| Server: 4 64-Bit Intel® Xeon™ Processor MP at 3.66GHz Clients: 12 Intel Xeon at 3.2GHz | DB2 UDB 8.2 | Microsoft Windows® Server 2003 Enterprise x64 Edition | Microsoft Visual C++ 6.0 Microsoft COM+ | 120,000 |



| System Component | Qty | Server: | Qty | Each of Six Clients: |
|------------------|-----|---|-----|-------------------------|
| Processors | 4 | Intel Xeon Processor MP at 3.66GHz/1MB L2 Cache | 2 | Intel Xeon DP at 3.2GHz |
| Cache | | | | 1MB L2 Cache |
| Memory | 16 | 4GB ECC RDIMM | 4 | 512MB |
| | | | 2 | 256MB |
| Disk Controllers | 1 | ServeRAID-6M Adapter | 1 | Ultra320 SCSI Interface |
| Disk Drives | 351 | 36.4GB (15000 rpm) | 1 | 36.4GB (15000 rpm) |
| Total Storage | | 11662GB | | |

| IBM Corporation | IBM @server xSeries 366 c/s | | | TPC-C Revision 5.3 | | | |
|---|-----------------------------|-------------------|---------|--|-----------------|----------------|--------------------|
| | DB2 UDB 8.2 | | | Report Date: March 28, 2005 | | | |
| Description | Part Number | Third Party Brand | Pricing | Unit Price | Quantity | Extended Price | 3-Yr. Maint. Price |
| Server Hardware | | | | | | | |
| xSeries 366 with 1 x 3.6GHz/1MB L2 Cache Intel Xeon Processor MP | 8863-2RU | IBM | 1 | 9,999 | 1 | 9,999 | |
| xSeries 3.6GHz/1MB L2 Cache Intel Xeon Processor MP | 13N0695 | IBM | 1 | 1,999 | 3 | 5,997 | |
| 8GB (2x4GB) PC2-3200 CL3 2RX4 ECC DDR2 SDRAM RDIMM | 30R5145 | IBM | 1 | 17,879 | 8 | 143,032 | |
| Active Memory™ 4-Slot Memory Expansion Card | 13M7409 | IBM | 1 | 499 | 3 | 1,497 | |
| ServeRAID-6M Ultra320 SCSI Adapter | 32P0033 | IBM | 1 | 999 | 1 | 999 | |
| E54 15" (13.8" Viewable) Color Monitor | 633147N | IBM | 1 | 139 | 1 | 139 | |
| IBM Preferred Pro Full-Size Keyboard PS/2 | 31P7415 | IBM | 1 | 29 | 1 | 29 | |
| IBM Sleek 2-Button Mouse | 28L3673 | IBM | 1 | 15 | 1 | 15 | |
| ServicePac for 3-Year 24x7x4 Support (x366) | 96P2253 | IBM | 1 | 900 | 1 | | 900 |
| ServicePac for 3-Year 24x7x4 Support (Monitor) | 30L9183 | IBM | 1 | 90 | 1 | | 90 |
| Discount on xSeries Hardware (15%) | | | | | | 24,256 | |
| Discount on ServicePacs (20%) | | | | | | | 198 |
| | | | | | Subtotal | 137,451 | 792 |
| Server Storage | | | | | | | |
| IBM TotalStorage DS4000 Host Bus Adapter | 24P0960 | IBM | 1 | 1,485 | 3 | 4,455 | |
| IBM TotalStorage DS4500 Disk Subsystem | 174290U | IBM | 1 | 66,500 | 3 | 199,500 | |
| IBM DS4000 Mini Hub | 19K1269 | IBM | 1 | 899 | 6 | 5,394 | |
| IBM Short Wave SFP Module | 19K1271 | IBM | 1 | 499 | 103 | 51,397 | |
| IBM 1m LC-LC Fibre Channel Cable | 19K1247 | IBM | 1 | 79 | 48 | 3,792 | |
| IBM 5m LC-LC Fibre Channel Cable | 19K1248 | IBM | 1 | 129 | 9 | 1,161 | |
| IBM TotalStorage DS4000 EXP710 Storage Exp. Unit | 1740710 | IBM | 1 | 6,000 | 24 | 144,000 | |
| IBM TotalStorage SAN Fibre Channel Switch Model H16 | 2005H16 | IBM | 1 | 11,755 | 1 | 11,755 | |
| 2Gbps FC 36.4GB 15K Hot-Swap HDD | 06P5772 | IBM | 1 | 1,115 | 336 | 374,640 | |
| IBM EXP400 Rack Storage Exp. Enclosure | 17331RU | IBM | 1 | 3,099 | 2 | 6,198 | |
| 36.4GB 15K Ultra320 SCSI Drive | 90P1318 | IBM | 1 | 349 | 15 | 5,235 | |
| Netfinity 4.2M Ultra2 SCSI Cable | 03K9310 | IBM | 1 | 75 | 2 | 150 | |
| IBM UPS 750TLV | 21301TX | IBM | 1 | 299 | 1 | 299 | |
| IBM S2 42U Standard Rack | 93074SX | IBM | 1 | 1,489 | 3 | 4,467 | |
| ServicePac for 3-Year 24x7x4 Support (EXP710) | 41L2768 | IBM | 1 | 760 | 24 | | 18,240 |
| ServicePac for 3-Year 24x7x4 Support (DS4500) | 96P2062 | IBM | 1 | 1,087 | 3 | | 3,261 |
| ServicePac for 3-Year 24x7x4 Support (EXP400) | 41L2768 | IBM | 1 | 760 | 2 | | 1,520 |
| ServicePac for 3-Year 24x7x4 Support (Rack) | 41L2760 | IBM | 1 | 300 | 3 | | 900 |
| ServicePac for 3-Year 24x7x4 Support (Switch) | 29R5130 | IBM | 1 | 2,460 | 1 | | 2,460 |
| Discount on Fibre Channel Disks (30%) | | | | | | 112,392 | |
| Discount on Fibre Channel Storage (20%) | | | | | | 84,291 | |
| Discount on xSeries Storage (15%) | | | | | | 2,452 | |
| Discount on ServicePacs (20%) | | | | | | | 5,276 |
| | | | | | Subtotal | 613,308 | 21,105 |
| Server Software | | | | | | | |
| DB2 UDB ESE 8.2 for Windows Operating Systems on 64-Bit Extended Systems - SW License and Maintenance 12 Months | | IBM | 1 | 22,260 | 4 | 89,040 | |
| SW Maintenance Renewal - 1 Year | | IBM | 1 | 1,060 | 8 | | 8,480 |
| Microsoft Windows Server 2003 Enterprise x64 Edition* | P72-00264 | Microsoft | 2 | 2,399 | 1 | 2,399 | |
| Microsoft Problem Resolution Services | | Microsoft | 2 | 245 | 1 | | 245 |
| | | | | | Subtotal | 91,439 | 8,725 |
| Client Hardware | | | | | | | |
| x226 with 3.2GHz/1MB Xeon DP, 512MB (2x256MB) Memory | 8648-2AU | IBM | 1 | 1,515 | 6 | 9,090 | |
| 3.2GHz/1MB Xeon DP Processor Upgrade | 13N0673 | IBM | 1 | 799 | 6 | 4,794 | |
| 1GB (2x512MB) PC-3200 DDR2 ECC SDRAM RDIMM | 73P3522 | IBM | 1 | 399 | 12 | 4,788 | |
| 36.4GB 15K Ultra320 SCSI Drive | 90P1318 | IBM | 1 | 349 | 6 | 2,094 | |
| PRO/1000 MT Dual-Port Server Adapter | 73P2701 | Intel | 1 | 249 | 18 | 4,482 | |
| E54 15" (13.8" Viewable) Color Monitor | 633147N | IBM | 1 | 139 | 6 | 834 | |
| ServicePac for 3-Year 24x7x4 Support (x226) | 96P2250 | IBM | 1 | 586 | 6 | | 3,516 |
| ServicePac for 3-Year 24x7x4 Support (Monitor) | 30L9183 | IBM | 1 | 90 | 6 | | 540 |
| Discount on xSeries Hardware (15%) | | | | | | 3,912 | |
| Discount on ServicePacs (20%) | | | | | | | 811 |
| | | | | | Subtotal | 22,170 | 3,245 |
| Client Software | | | | | | | |
| Microsoft Windows 2000 Server with COM+* | C11-00821 | Microsoft | 2 | 738 | 6 | 4,428 | |
| Microsoft Visual C++ Professional 6.0 | 254-00170 | Microsoft | 2 | 109 | 1 | 109 | |
| | | | | | Subtotal | 4,537 | |
| Network Components | | | | | | | |
| NETGEAR 8-Port Gigabit Ethernet Switch | GS108 | | 3 | 100 | 3 | 300 | |
| 14ft Ethernet Cable (including 2 spares) | | | 3 | 2 | 9 | 18 | |
| | | | | | Subtotal | 318 | |
| | | | | | Total | 869,223 | 33,867 |
| Pricing: 1- IBM; 2 - Microsoft (*See Quote for Discounts); 3 - www.newegg.com | | | | Three-Year Cost of Ownership USD: \$903,089 | | | |
| Notes: Volume Discounts Based on IBM Direct Guidance (Call 1-919-254-0367 for price verification.) | | | | tpmC: 150,704 | | | |
| Audited by Bradley J. Askins of InfoSizing, Inc. | | | | \$ USD/tpmC: \$5.99 | | | |
| Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. | | | | | | | |
| Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find that stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. | | | | | | | |

| Numerical Quantities Summary | | | |
|--|------------------------|----------------|-----------------------|
| MQTh, Computed Maximum Qualified Throughput: 150,704 tpmC | | | |
| Response Times (in seconds) | 90th Percentile | Average | Maximum |
| New-Order | 0.30 | 0.20 | 1.58 |
| Payment | 0.40 | 0.30 | 3.45 |
| Delivery | 0.30 | 0.29 | 1.06 |
| Stock Level | 0.64 | 0.41 | 1.94 |
| Order Status | 0.30 | 0.23 | 1.72 |
| Delivery (Deferred) | 0.58 | 0.33 | 3.53 |
| Menu | 0.29 | 0.20 | 1.16 |
| Response Time Delay Added for Emulated Components: 0.1 Seconds | | | |
| Transaction Mix (in percent of total transactions) | | | Percent |
| New-Order | | | 44.94 |
| Payment | | | 43.03 |
| Delivery | | | 4.01 |
| Stock-Level | | | 4.01 |
| Order Status | | | 4.01 |
| Keying/Think Times (in seconds) | Minimum | Average | Maximum |
| New Order | 18.00 / 0.00 | 18.00 / 12.03 | 18.00 / 120.31 |
| Payment | 3.00 / 0.00 | 3.00 / 12.04 | 3.02 / 120.31 |
| Delivery | 2.00 / 0.00 | 2.00 / 5.04 | 2.00 / 50.31 |
| Stock Level | 2.00 / 0.00 | 2.00 / 5.04 | 2.00 / 50.31 |
| Order Status | 2.00 / 0.00 | 2.00 / 10.03 | 2.00 / 100.31 |
| Test Duration | | | |
| Ramp-up time | | | 42 minutes 30 seconds |
| Measurement interval | | | 120 minutes |
| Number of checkpoints | | | NA |
| Checkpoint interval | | | NA |
| Number of transactions (all types) completed in measurement interval | | | 40,239,816 |

Table of Contents

| | |
|---|----|
| Abstract | 3 |
| Numerical Quantities Summary | 5 |
| Preface | 12 |
| General Items | 13 |
| Application Code Disclosure and Definition Statements | 13 |
| Benchmark Sponsor | 13 |
| Parameter Settings | 13 |
| Configuration Diagrams | 13 |
| Clause 1: Logical Database Design Related Items | 15 |
| Table Definitions | 15 |
| Physical Organization of the Database | 15 |
| Insert and Delete Operations | 15 |
| Horizontal or Vertical Partitioning | 15 |
| Replication | 15 |
| Table Attributes | 15 |
| Clause 2: Transaction and Terminal Profiles Related Items | 16 |
| Random Number Generation | 16 |
| Screen Layout | 16 |
| Terminal Verification | 16 |
| Intelligent Terminals | 16 |
| Transaction Profiles | 16 |
| Deferred Delivery Mechanism | 17 |
| Clause 3: Transaction and System Properties Related Items | 18 |
| Atomicity Requirements | 18 |
| Consistency Requirements | 18 |
| Isolation Requirements | 19 |
| Durability Requirements | 19 |
| Clause 4: Scaling and Database Population Related Items | 21 |
| Cardinality of Tables | 21 |
| Distribution of Tables and Logs | 21 |
| Database Model Implemented | 26 |
| Partitions/Replications Mapping | 26 |
| 60-Day Space Requirement | 26 |
| Clause 5: Performance Metrics and Response Time Related Items | 27 |
| Measured tpmC | 27 |
| Response Times | 27 |
| Keying/Think Times | 27 |
| Response Time Frequency Distribution Curves | 28 |
| Performance Curve for Response Time vs. Throughput | 30 |
| New Order Think Time Distribution | 31 |
| Steady State Methodology | 32 |
| Work Performed during Steady State | 32 |
| Measurement Interval | 33 |
| Transaction Mix | 33 |
| Percentage of Total Mix | 33 |
| Number of Checkpoints | 34 |
| Clause 6: SUT, Driver and Communication Definition Related Items | 35 |
| Description of RTE | 35 |
| Emulated Components | 35 |
| Benchmarked and Targeted System Configuration Diagrams | 35 |
| Network Configuration | 35 |
| Network Bandwidth | 35 |
| Operator Intervention | 35 |

| | |
|--|-----|
| Clause 7: Pricing Related Items | 36 |
| Hardware and Software Components | 36 |
| Availability Date | 36 |
| Measured tpmC | 36 |
| Country-Specific Pricing | 36 |
| Usage Pricing | 36 |
| System Pricing | 37 |
| Clause 9: Audit Related Items | 38 |
| Auditor | 38 |
| Availability of the Full Disclosure Report | 38 |
| <i>Attestation letter</i> | 39 |
| Appendix A: Client Server Code | 41 |
| A.1 Client/Terminal Handler Code | 41 |
| <i>makefile.config</i> | 41 |
| <i>include/tpccapp.h</i> | 44 |
| <i>include/tpccdbg.h</i> | 44 |
| <i>Src.Common/Makefile</i> | 45 |
| <i>Src.Common/tpccctx.sqc</i> | 46 |
| <i>Src.Common/tpccdbg.c</i> | 47 |
| <i>Src.Cli/Makefile</i> | 51 |
| <i>Src.Cli/tpcccli.sqc</i> | 52 |
| <i>nullDB/nullDB.h</i> | 57 |
| <i>nullDB/nullDB.cpp</i> | 57 |
| <i>nullDB/stdafx.h</i> | 60 |
| <i>nullDB/stdafx.cpp</i> | 60 |
| <i>tpccIsapi/htmlPhraser.h</i> | 60 |
| <i>tpccIsapi/resource.h</i> | 61 |
| <i>tpccIsapi/StdAfx.h</i> | 61 |
| <i>tpccIsapi/tpcc.h</i> | 61 |
| <i>tpccIsapi/tpccIsapi.def</i> | 71 |
| <i>tpccIsapi/tpccIsapi.hpp</i> | 71 |
| <i>tpccIsapi/htmlPhraser.cpp</i> | 72 |
| <i>tpccIsapi/StdAfx.cpp</i> | 74 |
| <i>tpccIsapi/tpccIsapi.cpp</i> | 74 |
| A.2 Client Transaction Code | 99 |
| <i>Makefile.config</i> | 99 |
| <i>tpccenv.bat</i> | 100 |
| <i>include/db2tpcc.h</i> | 100 |
| <i>include/lval.h</i> | 102 |
| <i>include/tpccapp.h</i> | 102 |
| <i>include/tpccdbg.h</i> | 103 |
| <i>Src.Common/Makefile</i> | 103 |
| <i>Src.Common/tpccctx.sqc</i> | 104 |
| <i>Src.Common/tpccdbg.c</i> | 105 |
| <i>Src.Common/tpccmisc.c</i> | 109 |
| <i>Src.Srv/Makefile</i> | 109 |
| <i>Src.Srv/cat-func.ddl</i> | 111 |
| <i>Src.Srv/cat-proc.ddl</i> | 116 |
| <i>Src.Srv/tpcc_all_sql.sqc</i> | 117 |
| <i>Src.Srv/unecat_func.ddl</i> | 142 |
| <i>Src.Srv/unecat_proc.ddl</i> | 142 |
| <i>Src.Srv/rpctpcc.def</i> | 142 |
| <i>utils/EXPLAIN.ddl</i> | 142 |
| <i>utils/UNEXPLAIN.ddl</i> | 147 |
| <i>tpccCom/comreg.h</i> | 147 |

| | |
|--|-----|
| <i>tpccCom/dlldatax.h</i> | 148 |
| <i>tpccCom/Resource.h</i> | 148 |
| <i>tpccCom/stdafx.h</i> | 148 |
| <i>tpccCom/tpccCom.h</i> | 148 |
| <i>tpccCom/tpcc_com.h</i> | 150 |
| <i>tpccCom/tpccCom.def</i> | 151 |
| <i>tpccCom/tpccCom.idl</i> | 151 |
| <i>tpccCom/tpcc_com.rgs</i> | 151 |
| <i>tpccCom/comreg.cpp</i> | 152 |
| <i>tpccCom/stdafx.cpp</i> | 152 |
| <i>tpccCom/tpccCom.cpp</i> | 152 |
| <i>tpccCom/tpcc_com.cpp</i> | 152 |
| <i>tpccCom/dlldata.c</i> | 156 |
| <i>tpccCom/dlldatax.c</i> | 156 |
| <i>tpccCom/tpccCom_i.c</i> | 157 |
| <i>tpccCom/tpccCom_p.c</i> | 157 |
| <i>TpccDB2Glue/stdafx.h</i> | 168 |
| <i>tpccDB2Glue/tpccDB2glue.h</i> | 168 |
| <i>tpccDB2Glue/stdafx.cpp</i> | 168 |
| <i>tpccDB2Glue/tpccDB2glue.cpp</i> | 168 |
| <i>NullDB.cpp</i> | 173 |
| <i>NullDB.h</i> | 176 |
| <i>Stdafx.cpp</i> | 176 |
| <i>Stdafx.h</i> | 176 |
| <i>Stdafx.cpp</i> | 177 |
| <i>StdAfx.h</i> | 177 |
| <i>TpccComClient.cpp</i> | 177 |
| <i>HtmlPhraser.cpp</i> | 178 |
| <i>HtmlPhraser.h</i> | 180 |
| <i>Resource.h</i> | 182 |
| <i>StdAfx.cpp</i> | 182 |
| <i>StdAfx.h</i> | 182 |
| <i>Tpcc.h</i> | 182 |
| <i>TpccIsapi.cpp</i> | 193 |
| <i>TpccIsapi.def</i> | 220 |
| <i>TpccIsapi.hpp</i> | 220 |
| <i>TpccIsapi.rc</i> | 222 |
| Appendix B: Database Design Scripts | 223 |
| <i>create_tablespace.ddl</i> | 223 |
| <i>alter_tablespace.ddl</i> | 226 |
| <i>alter_bufferpool.ddl</i> | 226 |
| <i>create_bufferpool.ddl</i> | 226 |
| <i>create_database.ddl</i> | 227 |
| <i>alittbsp_pf_0.ddl</i> | 227 |
| <i>alittbsp_pf_1024.ddl</i> | 227 |
| <i>crconst_customer.ddl</i> | 228 |
| <i>crconst_new_ordera.ddl</i> | 228 |
| <i>crconst_new_orderb.ddl</i> | 229 |
| <i>crconst_order_line.ddl</i> | 229 |
| <i>crconst_stock.ddl</i> | 229 |
| <i>cridx_cust_idxb.ddl</i> | 230 |
| <i>cridx_ordr_idxb.ddl</i> | 230 |
| <i>crtb_customer.ddl</i> | 230 |
| <i>crtb_item.ddl</i> | 232 |
| <i>crtb_district.ddl</i> | 232 |

| | |
|---------------------------------------|-----|
| <i>crtb_orders.ddl</i> | 232 |
| <i>crtb_order_line.ddl</i> | 233 |
| <i>crtb_new_ordera.ddl</i> | 234 |
| <i>crtb_new_orderb.ddl</i> | 235 |
| <i>crtb_stock.ddl</i> | 236 |
| <i>crtb_history.ddl</i> | 238 |
| <i>crtb_warehouse.ddl</i> | 238 |
| <i>crvw_customer.ddl</i> | 238 |
| <i>crvw_new_order.ddl</i> | 238 |
| <i>crvw_order_line.ddl</i> | 238 |
| <i>crvw_stock.ddl</i> | 239 |
| <i>gen_customer.bat</i> | 239 |
| <i>gen_district.bat</i> | 239 |
| <i>gen_history.bat</i> | 239 |
| <i>gen_item.bat</i> | 239 |
| <i>gen_new_order.bat</i> | 239 |
| <i>gen_orders.bat</i> | 239 |
| <i>gen_stock.bat</i> | 239 |
| <i>gen_warehouse.bat</i> | 240 |
| <i>load_customer_all.ddl</i> | 240 |
| <i>load_district_all.ddl</i> | 240 |
| <i>load_history_all.ddl</i> | 240 |
| <i>load_item_all.ddl</i> | 240 |
| <i>load_new_order_all.ddl</i> | 240 |
| <i>load_order_line_all.ddl</i> | 241 |
| <i>load_orders_all.ddl</i> | 241 |
| <i>load_stock_all.ddl</i> | 241 |
| <i>load_warehouse_all.ddl</i> | 242 |
| <i>rnst_customer.ddl</i> | 242 |
| <i>rnst_district.ddl</i> | 242 |
| <i>rnst_history.ddl</i> | 242 |
| <i>rnst_item.ddl</i> | 242 |
| <i>rnst_new_ordera.ddl</i> | 242 |
| <i>rnst_new_orderb.ddl</i> | 243 |
| <i>rnst_order_line.ddl</i> | 243 |
| <i>rnst_orders.ddl</i> | 243 |
| <i>rnst_stock.ddl</i> | 243 |
| <i>rnst_warehouse.ddl</i> | 243 |
| <i>DBGEN</i> | 244 |
| <i>dbgen\gendata.c</i> | 244 |
| <i>dbgen\makefile</i> | 251 |
| <i>dbgen\tpccrnd.c</i> | 252 |
| <i>dbgen\include\db2tpcc.h</i> | 254 |
| <i>dbgen\include\lval.h</i> | 257 |
| <i>dbgen\include\platform.h</i> | 257 |
| <i>dbgen\include\tpccrnd.h</i> | 259 |
| <i>dbgen\makefile.config</i> | 259 |
| <i>dbgen\Src.Common\makefile</i> | 259 |
| <i>dbgen\Src.Common\tpccmisc.c</i> | 260 |
| <i>dbgen\tpccenv.bat</i> | 261 |
| Appendix C: Tunable Parameters | 262 |
| IBM DB2 UDB | 262 |
| <i>Database Manager Configuration</i> | 262 |
| <i>Database Configuration</i> | 263 |
| <i>DB2set Parameters</i> | 264 |

| | |
|--|-----|
| Microsoft Windows Server 2003 Enterprise x64 Edition | 265 |
| <i>Server Configuration Parameters</i> | 265 |
| <i>System Information Report</i> | 265 |
| ServeRAID-6M Disk Controller Configuration Parameters | 296 |
| DS4500 Disk Subsystem Configuration | 301 |
| Client Configuration | 366 |
| <i>Microsoft Windows 2000 Client System Information Report</i> | 366 |
| <i>Client Configuration Parameters</i> | 390 |
| <i>Microsoft Windows 2000 Client Registry Parameters</i> | 390 |
| RTE Input Parameters | 392 |
| Appendix D: 60-Day Space | 394 |
| Appendix E: Third-Party Quotations | 395 |

Preface

The TPC Benchmark™ C was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark C Standard Specification Version 5.3.

The TPC describes this benchmark in Clause 0.1 of the specification as follows:

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

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

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

Despite the fact that this benchmark offers a rich environment that emulates many OLTP 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 any other environment 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 benchmarking when critical capacity planning and/or product evaluation decisions are contemplated.

General Items

Benchmark Sponsor

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

This benchmark was sponsored by International Business Machines Corporation.

Application Code Disclosure and Definition Statements

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

Appendix A contains all source code implemented in this benchmark.

Parameter Settings

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

- *Database tuning options*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and application configuration parameters.*
- *Compilation and linkage options and run-time optimizations used to create/install applications, OS, and/or databases.*

This requirement can be satisfied by providing a full list of all parameters and options.

Appendix C contains the tunable parameters for the database, the operating system, and the transaction monitor.

Configuration Diagrams

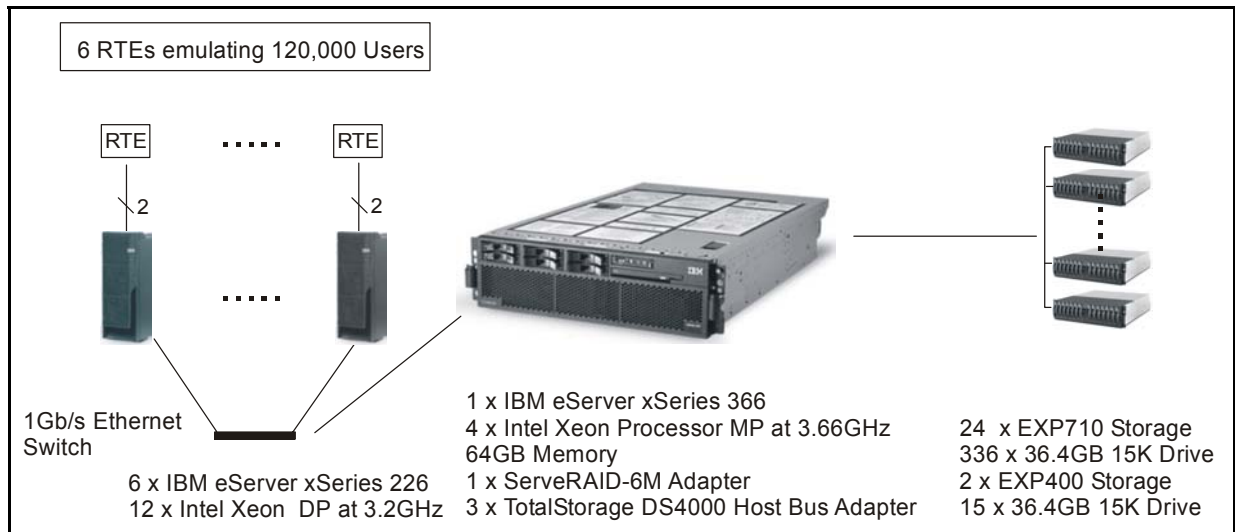
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

The configuration diagrams for the tested and priced systems are provided on the following pages.

The Remote Terminal Emulator (RTE) used for these TPC Benchmark C tests is an IBM proprietary RTE. Under Version 5.3, the components of the configuration being emulated by the RTE are the workstations and the Ethernet hubs. Appendix C contains a listing of the RTE scripts and inputs used in the benchmark testing.

The benchmarked configuration used six IBM eServer xSeries 226 systems, each configured with two 3.2GHz Intel Xeon processors, as the clients, which executed the terminal I/O and submitted transactions to COM+ servers, which are also running on the clients. These COM+ servers forwarded the transaction requests to the server, and returned the results to the RTE. DB2 UDB 8.2 is the DBMS executing on the server

Measured Configuration



The measured and priced configurations were identical. For the priced configuration, see the Executive Summary.

Clause 1: Logical Database Design Related Items

Table Definitions

Listings 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.

Physical Organization of the Database

The physical organization of tables and indexes within the database must be disclosed. Physical space was allocated to DB2 UDB on the server disks as detailed in Figure 4-2.

Insert and Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restriction in the SUT 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 during the running of the benchmark. The space required for an additional 5 percent of the initial table cardinality was allocated to DB2 UDB and priced as static space.

Horizontal or Vertical Partitioning

While there are few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C benchmark (see Clause 1.6), any such partitioning must be disclosed.

Only Stock, Orderline, Customer and New-order tables were horizontally partitioned into multiple tables. Stock, Orderline, Customer and New-order tables were partitioned into seven tables of 1,800 warehouses each. For each partitioned table, a view was created over all table partitions to provide full transparency of data manipulation. The rest of the tables were not partitioned.

Replication

Replication tables, if used, must be disclosed (see Clause 1.4.6).

Replication was not used in this benchmark.

Table Attributes

Additional and/or duplicated attributes in any table must be disclosed, along with a statement on the impact on performance (see Clause 1.4.7).

No additional attributes were used in this benchmark.

Clause 2: Transaction and Terminal Profiles Related Items

Random Number Generation

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

The seeds and offsets for the random number generator were collected and verified to be different for each driver. The auditor selected samples of the generated numbers from the database. The samples were verified to have no discernible patterns.

Screen Layout

The actual layouts of the terminal input/out screens must be disclosed.

All screen layouts followed the TPC Benchmark C Standard Specification.

Terminal 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 must for the demonstration in 8.1.3.3 must be disclosed and commercially available (including supporting software and maintenance).

The auditor verified terminal features by direct experimentation. The benchmarked configuration uses Microsoft Internet Explorer 6.0 SP1 and HTML scripts as the terminal interface.

Intelligent Terminals

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

The terminals emulated in the priced configuration are IBM PC desktop computer systems. All processing of the input/output screens was handled by the xSeries 226 clients. The screen input/output was managed via HTML strings that comply with the HTML Version 2.0 specification. A listing of the code used to implement the intelligent terminals is provided in Appendix A. All data manipulation was handled by the xSeries 366 database server.

Transaction Profiles

The percentage of home and remote order-lines in the New-Order transactions must be disclosed.

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

The number of items per orders entered by New-Order transactions must be disclosed. The percentage of home and remote Payment transactions must be disclosed. The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the database must be disclosed.

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

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

Table 2-1. Transaction Statistics

| New Order | Value (%) |
|--|------------------|
| Home warehouse order lines | 99.00 |
| Remote warehouse order lines | 1.00 |
| Rolled back transactions | 1.00 |
| Average number of items per order | 10.00 |
| Payment | |
| Home warehouse payment transactions | 0.85 |
| Remote warehouse payment transactions | 0.15 |
| Non-Primary Key Access | |
| Payment transactions using C_LAST | 59.99 |
| Order-Status transactions using C_LAST | 60.02 |
| Delivery | |
| Delivery transactions skipped | 0 |
| Transaction Mix | |
| New-Order | 44.94 |
| Payment | 43.03 |
| Delivery | 4.01 |
| Stock Level | 4.01 |
| Order Status | 4.01 |

Deferred Delivery Mechanism

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

The Delivery transaction was submitted to an ISAPI queue that is separate from the COM+ queue that the other transactions used. This queue is serviced by a variable amount of threads that are separate from the worker threads inside the web server. Web server threads are able to complete the on-line part of the Delivery transaction and immediately return successful queuing responses to the drivers. The threads servicing the queue are responsible for completing the deferred part of the transaction asynchronously.

The source code is listed in Appendix A.

Clause 3: Transaction and System Properties Related Items

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

Atomicity Requirements

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

All ACID tests were conducted according to specification.

Completed Transactions

The following steps were performed to verify the Atomicity of completed transactions.

1. The balance was retrieved from the CUSTOMER table for a random Customer, District and Warehouse, giving BALANCE_1.
2. The Payment transaction was executed for the Customer, District and Warehouse used in step 1.
3. The balance was retrieved again for the Customer used in step 1 and step 2, giving BALANCE_2. It was verified that BALANCE_1 was greater than BALANCE_2 by AMT.

Aborted Transactions

The following steps were performed to verify the Atomicity of the aborted Payment transaction:

1. The Payment application code was changed to execute a rollback of the transaction instead of performing the commit.
2. Using the balance, BALANCE_2, from the CUSTOMER table retrieved for the completed transaction, the Payment transaction was executed for the Customer, District and Warehouse used in step 1 of section 3.1.1. The transaction rolled back due to the change in the application code from step 1.
3. The balance was retrieved again for the Customer used for step 2, giving BALANCE_3. It was verified that BALANCE_2 was equal to BALANCE_3.

Consistency Requirements

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

Consistency conditions one through four were tested using a bat file to issue queries to the database. The results of the queries demonstrated that the database was consistent for all four tests.

Isolation Requirements

Sufficient conditions must be enabled at either the system or the application level to ensure that the required isolation defined in Clause 3.4.1 is obtained.

Isolation tests one through seven were run using the bat files to issue queries to the database. Each file included timestamps to demonstrate the concurrency of operations. The results of the queries were captured and placed in files. The auditor reviewed the results and verified that the isolation requirements had been met.

In addition, the phantom tests and the stock-level tests were run and verified.

Case A was followed for Isolation test seven.

Durability Requirements

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

- *Permanent irrecoverable failure of any single durable medium containing TPC-C database tables or recovery log data (this test includes failure of all or part of memory)*
- *Instantaneous interruption (system crash/system hang) in processing that requires system reboot to recover*
- *Failure of all or part of memory (loss of contents)*

Loss of Data Test

The following steps were successfully performed to pass the Durability test of failure of a disk unit with database tables:

1. The contents of the database were backed up to several database dump devices during the initial database load. There were no dump devices on the disk array from which a drive was removed as part of this test.
2. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID for all rows in the district table giving SUM1.
3. A test was started with 12,000 users submitting transactions.
4. A disk containing a portion of each of the tables in the tpcc database was removed causing DB2 to report errors accessing that device.
5. The run was aborted and DB2 was stopped.
6. The failed disk was replaced with a spare disk and was recovered.
7. The database was recovered and restored from the backup dump devices. Afterwards, the transaction log was applied to the database.
8. Step 2 was repeated to obtain the current count of the total number of orders giving SUM2.
9. It was verified that the sum of D_NEXT_O_ID after the database is recovered is greater than or equal to the sum of D_NEXT_O_ID before the run, plus all new order transactions completed during the run minus any rollback transactions.
10. Consistency Condition 3 was verified.

Loss of Log and Loss of System (Instantaneous Interruption and Loss of Memory)

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID for all rows in the district table giving SUM1.
2. This test was executed on a full scale benchmark run with 120,000 users.
3. The test continued to run for 5 minutes after all users were connected to the server.
4. One disk from the log array was removed. Since the disk was RAID-1 mirrored, DB2 continued to process transactions without interruption.
5. The test continued to run for another 5 minutes.

6. The server under test was powered off, which removed power from the system and the memory.
7. The server was powered on again.
8. DB2 was allowed to recover.
9. Step 1 was repeated to obtain the current count of the total number of orders giving SUM2.
10. It was verified that the sum of D_NEXT_O_ID after the database recovered was greater than or equal to the sum of D_NEXT_O_ID before the run, plus all new order transactions completed during the run minus any rollback transactions.

Clause 4: Scaling and Database Population Related Items

Cardinality of Tables

The cardinality (e.g., the number of rows) of each table, as it existed at the start of the benchmark run (see Clause 4.2), must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table were deleted (see Clause 4.2.2), the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed.

The database was built with 12,600 warehouses and the audited run used 12,000 warehouses. The warehouses over 12,000 were deleted each time the database was restored from backups.

Table 4-1. Initial Cardinality of Tables

| Table Name | Rows |
|---------------------|---------------|
| Warehouse | 12,600 |
| District | 126,000 |
| Item | 100,000 |
| New Order | 113,400,000 |
| History | 378,000,000 |
| Orders | 378,000,000 |
| Customer | 378,000,000 |
| Order Line | 3,780,124,350 |
| Stock | 1,260,000,000 |
| Inactive Warehouses | 600 |

Distribution of Tables and Logs

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

The logs are configured as a RAID-10 disk array with 14 physical disks of 36.4GB each attached via an internal ServeRAID-6M adapter.

For the database tables, there is a total of 336 disk drives. Each physical disk has a capacity of 36.4GB. A total of three Fibre Channel storage adapters connect these 336 disks.

Each disk volume, as seen by the Windows operating system, is configured as a RAID-0 disk array with 28 physical disks. There is a total of 12 such disk volumes. Each disk volume is further partitioned into 17 partitions. Each partition corresponds to a DB2 container.

Each partition within a partitioned table is made of either 2 DB2 containers (STOCK, CUSTOMER, ORDERLINE, NEWORDER) or 12 DB2 containers (the seventh partition of STOCK, CUSTOMER, ORDERLINE and NEWORDER), so that the corresponding view spans all 3 adapters (i.e., all 336 disks).

The WAREHOUSE, DISTRICT, ITEM, HISTORY and ORDER tables, which are not partitioned, are made up of 12 DB2 containers and also span all 3 adapters.

Figure 4-2 depicts the database configuration of the tested and priced systems to meet the 8-hour steady state requirement.

Figure 4-2. Data Distribution for the Benchmarked Configuration

| Disk # | Drives | Partition | Size | Use |
|--------|---|--|--------------------|--------------------------|
| 1 | 14 - 36.4GB 1 - 36.4GB EXP400 Enclosure | L: C: | 237.31GB 36.4GB | Database Log OS Drive |
| 2 | 28 - 36.4GB EXP710 Enclosure | C:\Containers\STK\001 C:\Containers\WAR\001 C:\Containers\DIS\001 C:\Containers\CST\001 C:\Containers\NEWA\001 C:\Containers\TM\001 C:\Containers\CSTI\001 C:\Containers\ORD\001 C:\Containers\ORDI\001 C:\Containers\OLN\001 C:\Containers\HST\001 C:\Containers\NEWB\001 C:\Containers\CST\013 C:\Containers\OLN\013 C:\Containers\NEWA\013 C:\Containers\NEWB\013 C:\Containers\CSTI\013 C:\Containers\STK\013 | 935.23GB | Database Backup Files |
| 3 | 28 - 36.4GB EXP710 Enclosure | C:\Containers\STK\002 C:\Containers\WAR\002 C:\Containers\DIS\002 C:\Containers\CST\002 C:\Containers\NEWA\002 C:\Containers\TM\002 C:\Containers\CSTI\002 C:\Containers\ORD\002 C:\Containers\ORDI\002 C:\Containers\OLN\002 C:\Containers\HST\002 C:\Containers\NEWB\002 C:\Containers\CST\014 C:\Containers\OLN\014 C:\Containers\NEWA\014 C:\Containers\NEWB\014 C:\Containers\CSTI\014 C:\Containers\STK\014 | 935.23GB | Database Backup Files |
| 4 | 28 - 36.4GB EXP710 Enclosure | C:\Containers\STK\003 C:\Containers\WAR\003 C:\Containers\DIS\003 C:\Containers\CST\003 C:\Containers\NEWA\003 C:\Containers\TM\003 C:\Containers\CSTI\003 C:\Containers\ORD\003 C:\Containers\ORDI\003 C:\Containers\OLN\003 C:\Containers\HST\003 C:\Containers\NEWB\003 C:\Containers\CST\015 C:\Containers\OLN\015 C:\Containers\NEWA\015 C:\Containers\NEWB\015 C:\Containers\CSTI\015 C:\Containers\STK\015 | 935.23GB | Database Backup Files |

| Disk # | Drives | Partition | Size | Use |
|--------|---------------------------------|--|----------|--------------------------|
| 5 | 28 - 36.4GB EXP710 Enclosure | C:\Containers\STK\004 C:\Containers\WAR\004 C:\Containers\DIS\004 C:\Containers\CST\004 C:\Containers\NEWA\004 C:\Containers\TM\004 C:\Containers\CSTI\004 C:\Containers\ORD\004 C:\Containers\ORDI\004 C:\Containers\OLN\004 C:\Containers\HST\004 C:\Containers\NEWB\004 C:\Containers\CST\016 C:\Containers\OLN\016 C:\Containers\NEWA\016 C:\Containers\NEWB\016 C:\Containers\CSTI\016 C:\Containers\STK\016 | 935.23GB | Database Backup Files |
| 6 | 28 - 36.4GB EXP710 Enclosure | C:\Containers\STK\005 C:\Containers\WAR\005 C:\Containers\DIS\005 C:\Containers\CST\005 C:\Containers\NEWA\005 C:\Containers\TM\005 C:\Containers\CSTI\005 C:\Containers\ORD\005 C:\Containers\ORDI\005 C:\Containers\OLN\005 C:\Containers\HST\005 C:\Containers\NEWB\005 C:\Containers\CST\017 C:\Containers\OLN\017 C:\Containers\NEWA\017 C:\Containers\NEWB\017 C:\Containers\CSTI\017 C:\Containers\STK\017 | 935.23GB | Database Backup Files |
| 7 | 28 - 36.4GB EXP710 Enclosure | C:\Containers\STK\006 C:\Containers\WAR\006 C:\Containers\DIS\006 C:\Containers\CST\006 C:\Containers\NEWA\006 C:\Containers\TM\006 C:\Containers\CSTI\006 C:\Containers\ORD\006 C:\Containers\ORDI\006 C:\Containers\OLN\006 C:\Containers\HST\006 C:\Containers\NEWB\006 C:\Containers\CST\018 C:\Containers\OLN\018 C:\Containers\NEWA\018 C:\Containers\NEWB\018 C:\Containers\CSTI\018 C:\Containers\STK\018 | 935.23GB | Database Backup Files |

| Disk # | Drives | Partition | Size | Use |
|--------|---------------------------------|--|----------|--------------------------|
| 8 | 28 - 36.4GB EXP710 Enclosure | C:\Containers\STK\007 C:\Containers\WAR\007 C:\Containers\DIS\007 C:\Containers\CST\007 C:\Containers\NEWA\007 C:\Containers\TM\007 C:\Containers\CSTI\007 C:\Containers\ORD\007 C:\Containers\ORDI\007 C:\Containers\OLN\007 C:\Containers\HST\007 C:\Containers\NEWB\007 C:\Containers\CST\019 C:\Containers\OLN\019 C:\Containers\NEWA\019 C:\Containers\NEWB\019 C:\Containers\CSTI\019 C:\Containers\STK\019 | 935.23GB | Database Backup Files |
| 9 | 28 - 36.4GB EXP710 Enclosure | C:\Containers\STK\008 C:\Containers\WAR\008 C:\Containers\DIS\008 C:\Containers\CST\008 C:\Containers\NEWA\008 C:\Containers\TM\008 C:\Containers\CSTI\008 C:\Containers\ORD\008 C:\Containers\ORDI\008 C:\Containers\OLN\008 C:\Containers\HST\008 C:\Containers\NEWB\008 C:\Containers\CST\020 C:\Containers\OLN\020 C:\Containers\NEWA\020 C:\Containers\NEWB\020 C:\Containers\CSTI\020 C:\Containers\STK\020 | 935.23GB | Database Backup Files |
| 10 | 28 - 36.4GB EXP710 Enclosure | C:\Containers\STK\009 C:\Containers\WAR\009 C:\Containers\DIS\009 C:\Containers\CST\009 C:\Containers\NEWA\009 C:\Containers\TM\009 C:\Containers\CSTI\009 C:\Containers\ORD\009 C:\Containers\ORDI\009 C:\Containers\OLN\009 C:\Containers\HST\009 C:\Containers\NEWB\009 C:\Containers\CST\021 C:\Containers\OLN\021 C:\Containers\NEWA\021 C:\Containers\NEWB\021 C:\Containers\CSTI\021 C:\Containers\STK\021 | 935.23GB | Database Backup Files |

| Disk # | Drives | Partition | Size | Use |
|--------|---------------------------------|--|----------|--------------------------|
| 11 | 28 - 36.4GB EXP710 Enclosure | C:\Containers\STK\010 C:\Containers\WAR\010 C:\Containers\DIS\010 C:\Containers\CST\010 C:\Containers\NEWA\010 C:\Containers\TM\010 C:\Containers\CSTI\010 C:\Containers\ORD\010 C:\Containers\ORDI\010 C:\Containers\OLN\010 C:\Containers\HST\010 C:\Containers\NEWB\010 C:\Containers\CST\022 C:\Containers\OLN\022 C:\Containers\NEWA\022 C:\Containers\NEWB\022 C:\Containers\CSTI\022 C:\Containers\STK\022 | 935.23GB | Database Backup Files |
| 12 | 28 - 36.4GB EXP710 Enclosure | C:\Containers\STK\011 C:\Containers\WAR\011 C:\Containers\DIS\011 C:\Containers\CST\011 C:\Containers\NEWA\011 C:\Containers\TM\011 C:\Containers\CSTI\011 C:\Containers\ORD\011 C:\Containers\ORDI\011 C:\Containers\OLN\011 C:\Containers\HST\011 C:\Containers\NEWB\011 C:\Containers\CST\023 C:\Containers\OLN\023 C:\Containers\NEWA\023 C:\Containers\NEWB\023 C:\Containers\CSTI\023 C:\Containers\STK\023 | 935.23GB | Database Backup Files |
| 13 | 28 - 36.4GB EXP710 Enclosure | C:\Containers\STK\012 C:\Containers\WAR\012 C:\Containers\DIS\012 C:\Containers\CST\012 C:\Containers\NEWA\012 C:\Containers\TM\012 C:\Containers\CSTI\012 C:\Containers\ORD\012 C:\Containers\ORDI\012 C:\Containers\OLN\012 C:\Containers\HST\012 C:\Containers\NEWB\012 C:\Containers\CST\024 C:\Containers\OLN\024 C:\Containers\NEWA\024 C:\Containers\NEWB\024 C:\Containers\CSTI\024 C:\Containers\STK\024 | 935.23GB | Database Backup Files |

Database Model Implemented

A statement must be provided that describes:

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

The database manager used for this testing was DB2 UDB 8.2, which is a relational database. DB2 remote stored procedures and embedded SQL statements were used. The DB2 stored procedures were invoked via SQL CALL statements. Both the client application and stored procedures were written in embedded C code.

Partitions/Replications Mapping

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

The Stock, Customer, Orderline and New-order tables were horizontally partitioned into multiple tables. The specifics of the distribution of partitioned and non-partitioned tables across the physical media are describe in Table 4-2. The database was not replicated.

60-Day Space Requirement

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 (Order, Order-Line, and History) must be disclosed (see Clause 4.2.3).

See Appendix D for details about how the 60-day space requirements were calculated.

Clause 5: Performance Metrics and Response Time Related Items

Measured tpmC

Measured tpmC must be reported.

Measured tpmC: 150,704 tpmC

Price per tpmC: \$5.99 USD per tpmC

Response Times

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

The TPC-C requirements for the average response time and the 90th percentile were met. Table 5-1 provides the response times for each of the transaction types and the menu for the measured system.

Table 5-1. Response Times in Seconds

| Transaction Type | Average | Maximum | 90 %-tile |
|---------------------|---------|---------|-----------|
| New-Order | 0.20 | 1.58 | 0.30 |
| Payment | 0.30 | 3.45 | 0.40 |
| Delivery | 0.29 | 1.06 | 0.30 |
| Stock Level | 0.41 | 1.94 | 0.64 |
| Order Status | 0.23 | 1.72 | 0.30 |
| Delivery (Deferred) | 0.33 | 3.53 | 0.58 |
| Menu | 0.20 | 1.16 | 0.29 |

Keying/Think Times

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

Table 5-2 lists the keying/think times for the measured system.

Table 5-2. Keying/Think Times

| Transaction Type | Average | Minimum | Maximum |
|------------------|---------------|--------------|---------------|
| New-Order | 18.00 / 12.03 | 18.00 / 0.00 | 18.00/ 120.31 |
| Payment | 3.00 / 12.04 | 3.00 / 0.00 | 3.02/ 120.31 |
| Delivery | 2.00 / 5.04 | 2.00 / 0.00 | 2.00/ 50.31 |
| Stock Level | 2.00 / 5.04 | 2.00 / 0.00 | 2.00/ 50.31 |
| Order Status | 2.00 / 10.03 | 2.00 / 0.00 | 2.00/ 100.31 |

Response Time Frequency Distribution Curves

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

Figure 5-1. New-Order Transaction - Response Time Frequency Distribution

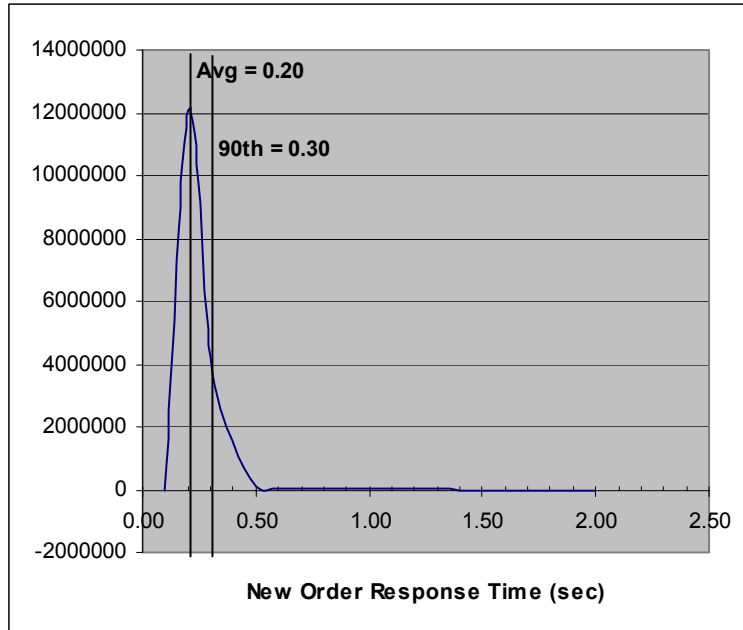


Figure 5-2. Payment Transaction - Response Time Frequency Distribution

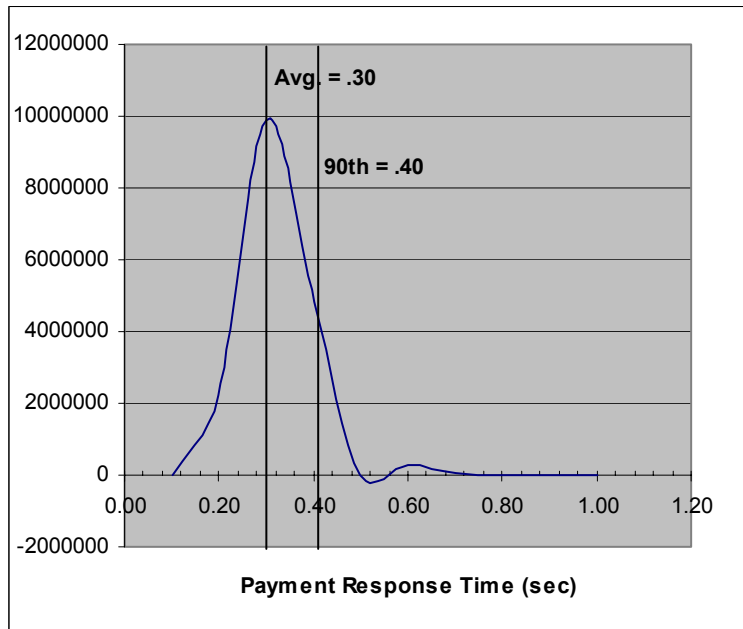


Figure 5-3. Order-Status Transaction - Response Time Frequency Distribution

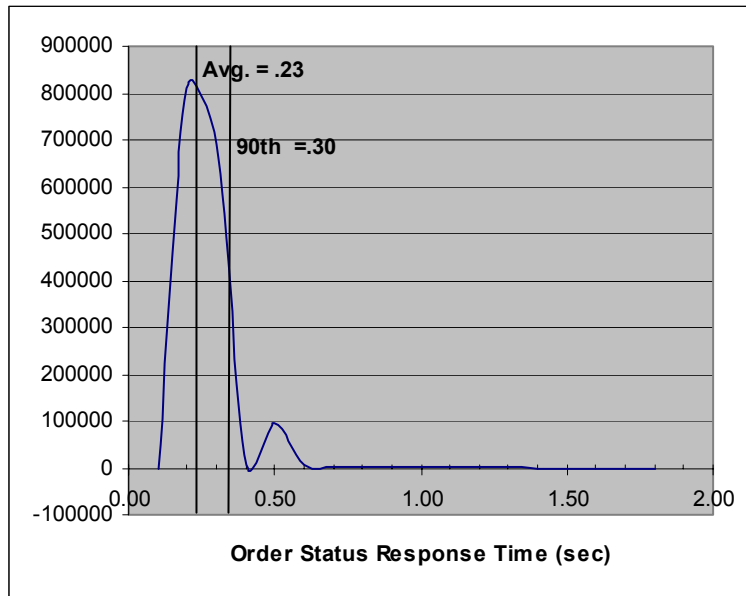


Figure 5-4. Delivery Transaction - Response Time Frequency Distribution

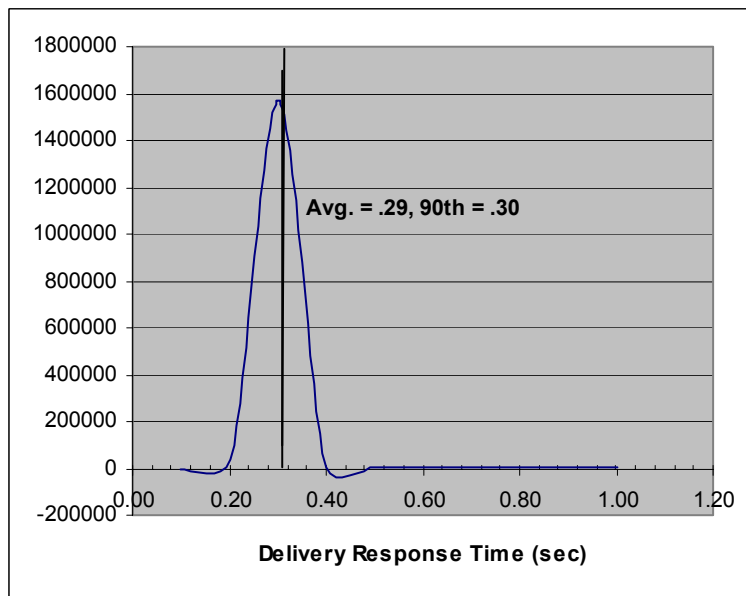
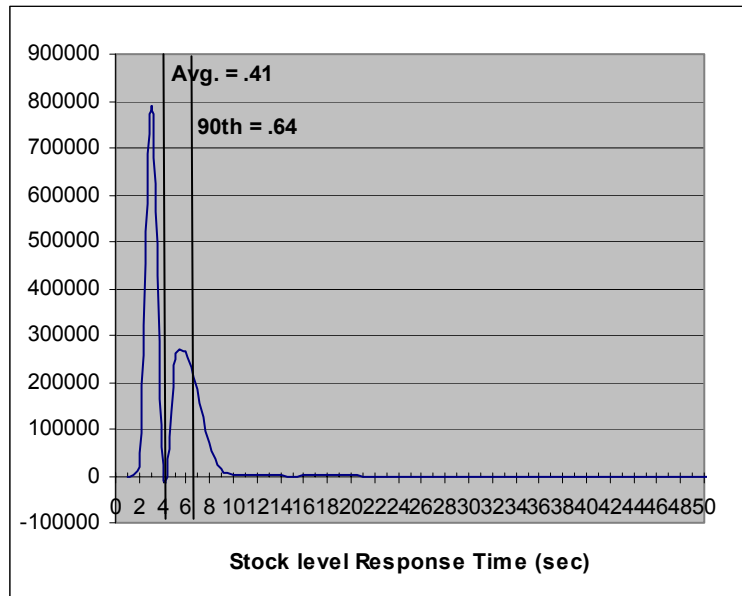


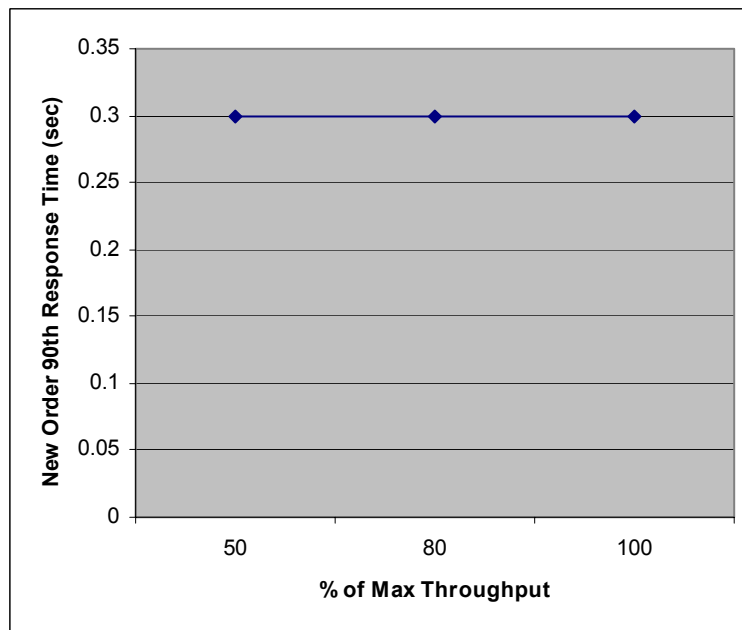
Figure 5-5. Stock-Level Transaction - Response Time Frequency Distribution



Performance Curve for Response Time vs. Throughput

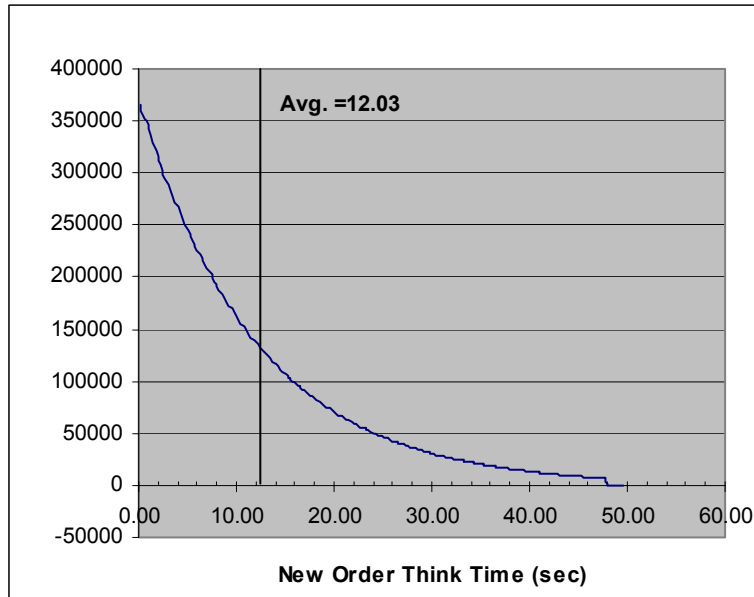
The performance curve for response time vs. throughput (see Clause 5.6.2) must be reported for the New-Order transaction.

Figure 5-6. New-Order Response Time vs. Throughput



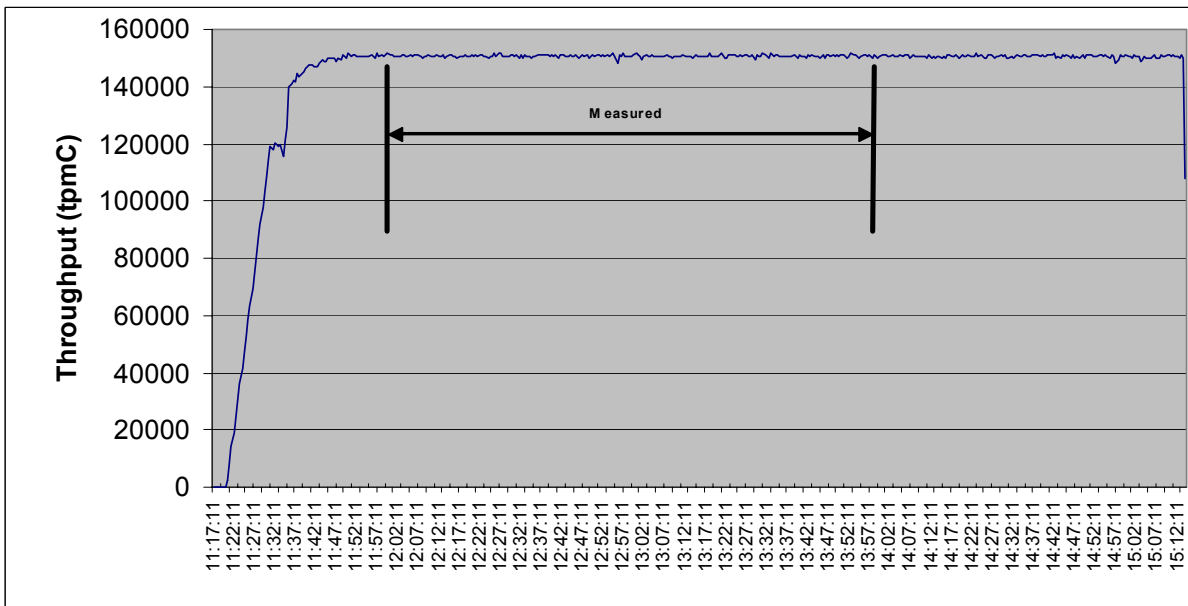
New Order Think Time Distribution

Figure 5-7. New-Order Think Time Distribution



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

Figure 5-8. New-Order Throughput vs. Elapsed Time



Steady State Methodology

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

Figure 5-8 shows that the system was in steady state at the beginning of the measurement interval.

Work Performed during Steady State

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

Transaction Flow

The RTE generated the required input data to choose a transaction from the menu. This data was time-stamped. The response for the requested transaction was verified and time-stamped 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 input screen. The transmission was time-stamped. The return of the screen with the required response data was time-stamped. The difference between these two time-stamps 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 starting at selecting another transaction from the menu.

The RTE transmissions were sent to application processes running on the client machines through Ethernet LANs. Each of the 4 (non-delivery) transactions is serviced by two individual programs, Internet Information Service 5.1 (IIS) and a Microsoft COM+ 1.0 Queued Component Server, used as the transaction manager (COM+). Both programs are running on the client system:

- The initial HTML 1.0 request is serviced by an ISAPI custom-written handler running on Internet Information Service 5.1(IIS). IIS is responsible for handling all HTML requests. The web server communicates to the COM+ server through a Microsoft COM+ api interface.
- COM+ communicates with the Server system over Ethernet and handles all database operations, using DB2 embedded SQL calls. When the COM+ server boots up, it creates a configurable amount of connections to the Server (listed in application settings). COM+ routes the transaction and balances the load according to the options defined in the Component Services GUI for the COM+ server application and settings in the Windows 2000 Registry. The configuration file and registry variables are listed in Appendix B.2. At the beginning, each TPC-C user sends a pair of HTML 1.0 requests submitting the its unique warehouse and district to the IIS ISAPI handler. Upon successful validation of the user's login, IIS the displays an HTML form that encapsulates the TPC-C transaction menu.

The transaction flow is described below:

1. The TPC-C user requests the transaction type's HTML form and proceeds to generate (fill in) a GET request with the required files for the transaction.
2. IIS accepts the filled-in GET request, parses, and validates all values entered by the user.
3. It then proceeds to transmit those values to the COM+ server through an transaction type-specific COM+ api interface.
4. The COM+ Pool Manager receives the request and first decides if there is a connection object in the pool available to service it. If so, the connection is used to send the transaction request to the Server. If no connection is available, the request will enter a COM+ internal queue and will be serviced by the next available connection.
5. Once the connection is available to be used, a COM+ pool thread receives the transaction and calls a TPC-C back-end DB2 client api to execute all database operations related to the transaction type. (All the transaction information entered on the HTML form is available in a data structure provided by the ISAPI caller.)
6. The transaction is committed and the DB2 back-end client returns control back to the COM pool thread.

7. COM pool thread returns control to the ISAPI caller. (All transaction results are inside the data structure that the ISAPI caller provided to the COM+ api in the parameter list.)
- 8 ISAPI caller returns control to the "screen application" by doing a PUT request.

Measurement Interval

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.

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. (8.1.6.13)

The RTE was given a weighted random distribution, which was not adjusted during the run. See Table 5-3.

Percentage of Total Mix

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

See Table 5-3.

Table 5-3. Transaction Statistics and Transaction Mix

| New Order | Value (%) |
|--|------------------|
| Home warehouse order lines | 99.00 |
| Remote warehouse order lines | 1.00 |
| Rolled back transactions | 1.00 |
| Average number of items per order | 10.00 |
| Payment | |
| Home warehouse payment transactions | 0.85 |
| Remote warehouse payment transactions | 0.15 |
| Non-Primary Key Access | |
| Payment transactions using C_LAST | 59.99 |
| Order-Status transactions using C_LAST | 60.02 |
| Delivery | |
| Delivery transactions skipped | 0 |
| Transaction Mix | |
| New-Order | 44.94 |
| Payment | 43.03 |
| Delivery | 4.01 |
| Stock Level | 4.01 |
| Order-Status | 4.01 |

Number of Checkpoints

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.

DB2 UDB uses a write-ahead-logging protocol to guarantee recovery. This protocol uses “soft” checkpoint to write least-recently-used database pages to disk independent of transaction commit. However, enough log information to redo/undo the change to a database pages is committed to disk before the database page itself is written. This protocol, therefore, renders checkpoint unnecessary for DB2 UDB.

For a more detailed description of the general principles of the write-ahead-logging protocol, see the IBM research paper, “ARIES: A Transaction Recovery Method Supporting Fine Granularity Locking and Partial Rollbacks Using Write-Ahead Logging,” by C. Mohan, Database Technology Institute, IBM Almaden Research Center.

<http://portal.acm.org/citation.cfm?id=128770&coll=portal&dl=ACM&CFID=10343790&CFTOKEN=42047146>

Clause 6: SUT, Driver and Communication Definition Related Items

Description of RTE

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

The RTE used is IBM-developed proprietary software. The RTE input is listed in Appendix C.

Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to that of the priced system. The results of the test described in Clause 6.6.3.4 must be disclosed.

No components were emulated.

Benchmarked and Targeted System Configuration Diagrams

A complete functional diagram of both the benchmarked configuration and the configuration of the proposed (target) system must be disclosed. A detailed list of all software and hardware functionality being performed on the Driver System, and its interface to the SUT must be disclosed (see Clause 6.6.3.6).

The driver RTE generated the transaction input data and transmitted it to the client in HTML format. The driver RTE received the output from the System Under Test, time-stamped it, and forwarded it to the Master RTE for post-test processing. No other functionality was included on the driver RTE.

Detailed diagrams of the benchmarked and priced configurations are provided in the section called “General Items” at the beginning of this document.

Network Configuration

The network configurations of both the tested services and the proposed (target) services which are being represented and a thorough explanation of exactly which parts of the proposed configuration are being replaced with the Driver System must be disclosed (see Clause 6.6.4).

See the measured and priced configuration diagrams for details about the network configuration.

Network Bandwidth

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

The Ethernet used in the LAN connecting the clients and driver RTEs complies with the IEEE.802.3 standard. The Ethernet LAN had a bandwidth of 1Gbps. The LAN that connected the clients to the server complies with the IEEE.802.3 standard. The Ethernet LAN had a bandwidth of 1Gbps.

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 did not require any operator intervention to sustain the reported throughput.

Clause 7: Pricing Related Items

Hardware and Software Components

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

Pricing source(s) and effective date(s) must also be reported.

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

A detailed list of all hardware and software, including the 3-year price, is provided in the Executive Summary at the front of this report. All third-party quotations are included in Appendix E at the end of this document.

Availability Date

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

The total solution as priced will be generally available August 20, 2005.

Measured tpmC

A statement of the measured tpmC, as well as the respective calculations for the 3-year pricing, price/performance (price/tpmC) and the availability date must be included.

- Maximum Qualified Throughput: 150,704 tpmC
- Price per tpmC: \$5.99 USD per tpmC
- Three-year cost of ownership: \$903,089 USD

Country-Specific Pricing

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

The configuration is priced for the United States of America.

Usage Pricing

For any usage pricing, the sponsor must disclose:

- *Usage level at which the component was priced.*
- *A statement of the company policy allowing such pricing.*

The component pricing based on usage is shown below:

- 1 Microsoft Windows Server 2003, Enterprise x64 Edition
- 6 Microsoft Windows 2000 Server
- 4 DB2 UDB 8.2 (based on per-processor price)

- 3-year support for hardware components (except for components for which a minimum of 2 or 10 percent spares are provided)

System Pricing

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

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

A detailed list of all hardware and software, including the 3-year price, is provided in the Executive Summary at the front of this report. All third-party quotations are included in Appendix E at the end of this document.

Clause 9: Audit Related Items

Auditor

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

This implementation of the TPC-C benchmark was audited by Bradley J. Askins of InfoSizing, Inc. The auditor's attestation letter is provided in this section.

Availability of the Full Disclosure Report

The Full Disclosure Report must be readily available to the public at a reasonable charge, similar to the charges for similar documents by the test sponsor. The report must be made available when results are made public. In order to use the phrase "TPC BenchmarkTMC," the Full Disclosure Report must have been submitted to the TPC Administrator as well as written permission obtained to distribute same.

The TPC Benchmark C Full Disclosure Report can be obtained from www.tpc.org.

Benchmark Sponsor: Berni Schiefer
 IBM DB2 Performance
 8200 Warden Avenue
 Markham, Ontario L6G1C7

March 22, 2005

I verified the TPC Benchmark™ C performance for the following Client/Server configuration:

Platform: **IBM @server xSeries 366 c/s**
 Operating system: **Microsoft Windows Server 2003 Enterprise x64 Edition**
 Database Manager: **DB2 UDB 8.2**
 Transaction Manager: **Microsoft COM+**

The results were:

| CPU's Speed | Memory | Disks | NewOrder 90% Response Time | tpmC |
|--|---|---------------|----------------------------|-------------------|
| Server: IBM @server xSeries 366 | | | | |
| 4 x Xeon MP (3.6GHz) | 64 GB Main (1MB L2 Cache) | 351 x 36.4 GB | 0.30 Seconds | 150,704.91 |
| Clients: Six (6) IBM @server xSeries 226 (Specification for each) | | | | |
| 2 x Xeon DP (3.2 GHz) | 2.5GB Main (1 MB L2 Cache per processor) | 1 x 36.4 GB | n/a | n/a |

In my opinion, these performance results were produced in compliance with the TPC's requirements for the benchmark. The following verification items were given special attention:

- The database records were the proper size
- The database was properly scaled and populated
- The required ACID properties were met

- The transactions were correctly implemented
- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured.
- All 90% response times were under the specified maximums
- At least 90% of all delivery transactions met the 80 Second completion time limit
- The reported measurement interval was 120 minutes (7200 seconds)
- The reported measurement interval was representative of steady state conditions
- Write-ahead-logging was active during the measurement interval
- The 60 day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

None.

Respectfully Yours,

A handwritten signature in black ink, appearing to read "François Raab". The signature is fluid and cursive, with a long horizontal stroke extending to the right.

François Raab, President

A handwritten signature in black ink, appearing to read "Bradley J. Askins". The signature is cursive and includes a large, stylized initial "B".

Bradley J. Askins, Auditor

Appendix A: Client Server Code

A.1 Client/Terminal Handler Code

makefile.config

```
#####  
#####  
## Licensed Materials - Property of IBM  
##  
## Governed under the terms of the International  
## License Agreement for Non-Warranted Sample Code.  
##  
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2005  
## All Rights Reserved.  
##  
## US Government Users Restricted Rights - Use, duplication or  
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
#####  
#####  
  
#  
# Makefile.config - NT/Win2000 Makefile Configuration  
#  
  
# Make Configuration (MSVC)  
MAKE=nmake.exe  
  
# Compiler Configuration (MSVC).  
# CFLAGS_DEBUG may be set to "-Zi -Od", "-DDEBUGIT" "-Zi -Od  
-DDEBUGIT" or left blank  
CC=cl.exe  
CFLAGS_OS=-DSQLWINT -MT -DWIN32 -J -Zp8 -DREG_KIT_METHOD  
CFLAGS_OUT=/Fo  
CFLAGS_DEBUG=  
  
# Linker Configuration (MSVC)  
LD_EXEC=link.exe  
LD_STORP=link.exe  
LDFLAGS_EXEC=  
LDFLAGS_SHLIB=/DLL  
LDFLAGS_STORP=$(LDFLAGS_SHLIB) /DEF:rpctpcc.def  
LDFLAGS_LIB=/LIBPATH:$(TPCC_SQLLIB)\lib /LIBPATH:"C:\Program  
Files\Microsoft Visual Studio\VC98\Lib" db2api.lib winmm.lib  
LDFLAGS_OUT=/OUT:  
  
# Library Configuration  
AR=lib.exe  
ARFLAGS=  
ARFLAGS_LIB=  
ARFLAGS_OUT=/OUT:  
  
# OS Commands  
ERASE=del /F  
ERASEDIR=rmdir /S  
MOVE=MOVE  
COPY=COPY  
  
# OS File Extensions & Path Separator  
OBJEXT=.obj  
LIBEXT=.lib  
SHLIBEXT=.dll  
BINEXT=.exe  
SLASH=\  
CMDSEP=&
```

```
tpccenv.bat  
  
@REM  
*****  
*****  
@REM Licensed Materials - Property of IBM  
@REM  
@REM Governed under the terms of the International  
@REM License Agreement for Non-Warranted Sample Code.  
@REM  
@REM (C) COPYRIGHT International Business Machines Corp. 1996 - 2005  
@REM All Rights Reserved.  
@REM  
@REM US Government Users Restricted Rights - Use, duplication or  
@REM disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
@REM  
*****  
*****  
@REM  
@REM tpccenv.bat - Windows Environment Setup  
@REM  
  
@REM The Kit Version  
set TPCC_VERSION=CK041012  
  
@REM The DB2 Instance Name (for DB2)  
set DB2INSTANCE=DB2  
  
@REM The OS being used (i.e. "UNIX", "WINDOWS")  
set PLATFORM=WINDOWS  
  
@REM The type of make command and slash used by the OS  
@REM (i.e. UNIX - "/", WINDOWS - "\")  
@REM These are referenced all over the kit.  
set SLASH=  
set MAKE=nmake  
  
set TPCC_SPTYPE=SPGENERAL  
  
set DB2VERSION=v8  
  
@REM The schema name is typically the SQL authorization ID (or username).  
@REM This is required for runstats and EEE.  
set TPCC_SCHEMA=%USERNAME%  
  
@REM DB2 EE/EEE Configuration  
set DB2EDITION=EE  
set DB2NODE=0  
set DB2NODES=1  
  
set HOME=C:  
set TPCC_DBNAME=TPCC  
set TPCC_ROOT=c:\tpcckit\tpc-c.ibm  
set TPCC_SQLLIB=c:\SQLLIB  
set TPCC_RUNDATA=c:\tpcckit\tpccdata  
  
set TPCC_DEBUGDIR=c:\temp  
  
@REM Specifies where stored procedures should be placed and if they should  
@REM be fenced.  
set TPCC_SPDIR=%TPCC_SQLLIB%\function  
set TPCC_FENCED=NO  
include/db2tpcc.h  
/*****  
*****  
** Licensed Materials - Property of IBM  
**  
** Governed under the terms of the International
```

```

** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****
***/

/*
 * db2tpcc.h - Macros and Miscellany
 */

#ifndef __DB2TPCC_H
#define __DB2TPCC_H

#include <sys/types.h>
typedef __int16 int16_t;
typedef __int32 int32_t;
typedef __int64 int64_t;

#include "lval.h"

/*
*****
***** */
/* Transaction Return Codes (s_transtatus) */
/*
*****
***** */

#define INVALID_ITEM 100
#define TRAN_OK 0
#define FATAL_SQLERROR -1

/*
*****
***** */
/* Definition of Unused and Bad Items */
/*
*****
***** */
/* Define unused item ID to be 0. This allows the SUT to determine the
 * number of items in the order as required by 2.4.1.3 and 2.4.2.2 since
 * the assumption that any item with OL_I_ID = 0 is unused will be true.
 * This in turn requires that the value used for an invalid item is
 * equal to ITEMS + 1.
 */
/*
*****
***** */

#define INVALID_ITEM_ID (2 * ITEMS) + 1
#define UNUSED_ITEM_ID 0

#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES

/*
*****
***** */
/* NURand Constants */
/* C_C_LAST_RUN and C_C_LAST_LOAD must adhere to clause 2.1.6.
 */
/*
*****
***** */
#define C_C_LAST_RUN 88
#define C_C_LAST_LOAD 173
#define C_C_ID 319

```

```

#define C_OL_I_ID 3849
#define A_C_LAST 255
#define A_C_ID 1023
#define A_OL_I_ID 8191

/*
*****
***** */
/* Transaction Type Identifiers */
/*
*****
***** */

#define CLIENT_SQL 0
#define NEWORD_SQL 1
#define PAYMENT_SQL 2
#define ORDSTAT_SQL 3
#define DELIVERY_SQL 4
#define STOCKLEV_SQL 5

#define SPGENERAL_PAD 3
#define SPGENERAL_ADJUST sizeof(int16_t)

struct in_neword_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    struct in_items_struct {
        int32_t s_OL_I_ID;
        int32_t s_OL_SUPPLY_W_ID;
        int16_t s_OL_QUANTITY;
        int16_t pad1[3];
    } in_item[15];
    int64_t s_O_ENTRY_D_time; /* init by SUT */
    int32_t s_C_ID;
    int32_t s_W_ID;
    int16_t s_D_ID;
    int16_t s_O_OL_CNT; /* init by SUT */
    int16_t s_all_local;
    int16_t duplicate_items;
};

struct out_neword_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    struct items_struct {
        int32_t s_I_PRICE;
        int32_t s_OL_AMOUNT;
        int16_t s_S_QUANTITY;
        int16_t pad2;
        char s_I_NAME[25];
        char s_brand_generic;
    } item[15];
    int64_t s_O_ENTRY_D_time;
    int32_t s_W_TAX;
    int32_t s_D_TAX;
    int32_t s_C_DISCOUNT;
    int32_t s_total_amount;
    int32_t s_O_ID;
    int16_t s_O_OL_CNT;
    int16_t s_transtatus;
    int16_t deadlocks;
    char s_C_LAST[17];
    char s_C_CREDIT[3];
};

struct in_payment_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_H_DATE_time; /* init by SUT */
    int64_t s_H_AMOUNT;

```

```

int32_t s_W_ID;
int32_t s_C_W_ID;
int32_t s_C_ID;
int16_t s_C_D_ID;
int16_t s_D_ID;
char s_C_LAST[17];
};

struct out_payment_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int64_t s_H_DATE_time;
int64_t s_C_SINCE_time;
int64_t s_C_CREDIT_LIM;
int64_t s_C_BALANCE;
int32_t s_C_DISCOUNT;
int32_t s_C_ID;
int16_t s_transtatus;
int16_t deadlocks;
char s_W_STREET_1[21];
char s_W_STREET_2[21];
char s_W_CITY[21];
char s_W_STATE[3];
char s_W_ZIP[10];
char s_D_STREET_1[21];
char s_D_STREET_2[21];
char s_D_CITY[21];
char s_D_STATE[3];
char s_D_ZIP[10];
char s_C_FIRST[17];
char s_C_MIDDLE[3];
char s_C_LAST[17];
char s_C_STREET_1[21];
char s_C_STREET_2[21];
char s_C_CITY[21];
char s_C_STATE[3];
char s_C_ZIP[10];
char s_C_PHONE[17];
char s_C_CREDIT[3];
char s_C_DATA[201];
};

struct in_ordstat_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_C_ID;
int32_t s_W_ID;
int16_t s_D_ID;
int16_t pad1[3];
char s_C_LAST[17];
};

struct out_ordstat_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int64_t s_C_BALANCE;
int64_t s_O_ENTRY_D_time;
int32_t s_C_ID;
int32_t s_O_ID;
int16_t s_O_CARRIER_ID;
int16_t s_ol_cnt;
int16_t pad1[2];
struct oitems_struct {
int64_t s_OL_DELIVERY_D_time;
int32_t s_OL_AMOUNT;
int32_t s_OL_I_ID;
int32_t s_OL_SUPPLY_W_ID;
int16_t s_OL_QUANTITY;
int16_t pad2;
} item[15];
int16_t s_transtatus;
int16_t deadlocks;
char s_C_FIRST[17];
char s_C_MIDDLE[3];
char s_C_LAST[17];
};

struct in_delivery_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int64_t s_O_DELIVERY_D_time; /* init by SUT */
int32_t s_W_ID;
int16_t s_O_CARRIER_ID;
};

struct out_delivery_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_O_ID[10];
int16_t s_transtatus;
int16_t deadlocks;
};

struct in_stocklev_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_threshold;
int32_t s_W_ID;
int16_t s_D_ID;
};

struct out_stocklev_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_low_stock;
int16_t s_transtatus;
int16_t deadlocks;
};

/*
***** */
/* Transaction Prototypes */
/*
***** */

#ifdef __cplusplus
extern "C" {
#endif

extern int neword_sql(struct in_neword_struct*, struct out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct out_stocklev_struct*);

#ifdef __cplusplus
}
#endif

/*
***** */
/* DB2 Connect/Disconnect & Thread Context Wrappers */

```

```

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

#ifdef __cplusplus
extern "C" {
#endif

extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*, char*, char*);
extern int disconnect_from_TM(void);

extern int create_context(void);
extern int destroy_context(void);
extern int get_context(void**);
extern int attach_context(void**);
extern int detach_context(void**);

#ifdef __cplusplus
}
#endif

#endif // __DB2TPCC_H

include/tpccapp.h

** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****

/*
* tpccapp.h - Application Macros
*/

#ifdef __TPCCAPP_H
#define __TPCCAPP_H

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#include "sqlenv.h"
#define daricall __stdcall

#include "sqlca.h"
#include "sqlcodes.h"

#ifdef SWAP_ENDIAN
#define SWAP_BYTE(Var) SwapEndian((void*)&Var, sizeof(Var))

/*
*****
***** */
FUNCTION: SwapEndian
PURPOSE: Swap the byte order of a structure
EXAMPLE: int I=0x12345678; SWAP_BYTE(I); I => 0x78563412;
IMPLEMENTATION: Fold Addr in half, swap header & tail by XOR op
e.g.: *a = 0x12 [ Addr + 0];

```

```

*b = 0x78 [ Addr + 4 - 0 - 1 = Addr+3];
*a ^= *b; // sets *a to 0x6A
*b ^= *a; // sets *b to 0x12
*a ^= *b; // sets *a to 0x78

Now *a => 0x78 && *b => 0x12
*****
*****/

void SwapEndian(void *Addr, int nb)
{
int i;
for (i=0; i<nb/2; i++)
{
char *a = (char*)Addr+i;
char *b = (char*)Addr+(nb-i-1);

*a ^= *b;
*b ^= *a;
*a ^= *b;
}
}
#endif //SWAP_ENDIAN

/*
*****
*****/
/* SQLCODE Macros */
/*
*****
*****/

#define DLCHK(a) \
if (sqlca.sqlcode == SQL_RC_E911) { goto a; }

#define NACOMPCHK(last) \
if (sqlca.sqlcode != SQL_RC_E1339) { last = -1; } \
else { int a = ((sqlca.sqlerrmc[4] == 0x20) ? 0 : sqlca.sqlerrmc[4]-0x30); \
int b = ((sqlca.sqlerrmc[5] == 0x20) ? 0 : sqlca.sqlerrmc[5]-0x30); \
if (b == 0) { last = a; } else { last = a * 10 + b; } \
}

#endif // __TPCCAPP_H

include/tpccdbg.h

/*
*****
***** */
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****

/*
* tpccdbg.h - Debugging Macros
*/

#ifdef __TPCCDBG_H
#define __TPCCDBG_H

#ifdef __cplusplus

```

```

extern "C" {
#endif

extern void sqlerror (int tranType, char *msg, char *file, int line,
    SQL_STRUCTURE sqlca *psqlca);

extern void new_debug (struct out_neword_struct *neword_ptr,
    struct in_neword_struct *in_neword_ptr,
    char *msg);
extern void pay_debug (struct out_payment_struct *payment_ptr,
    struct in_payment_struct *in_payment_ptr,
    char *msg);
extern void ord_debug (struct out_ordstat_struct *ordstat_ptr,
    struct in_ordstat_struct *in_ordstat_ptr,
    char *msg);
extern void del_debug (struct out_delivery_struct *delivery_ptr,
    struct in_delivery_struct *in_delivery_ptr,
    char *msg);
extern void stk_debug (struct out_stocklev_struct *stocklev_ptr,
    struct in_stocklev_struct *in_stocklev_ptr,
    char *msg);

extern void new_print (struct out_neword_struct *neword_ptr,
    struct in_neword_struct *in_neword_ptr,
    char *filename,
    char *msg);
extern void pay_print (struct out_payment_struct *payment_ptr,
    struct in_payment_struct *in_payment_ptr,
    char *filename,
    char *msg);
extern void ord_print (struct out_ordstat_struct *ordstat_ptr,
    struct in_ordstat_struct *in_ordstat_ptr,
    char *filename,
    char *msg);
extern void del_print (struct out_delivery_struct *delivery_ptr,
    struct in_delivery_struct *in_delivery_ptr,
    char *filename,
    char *msg);
extern void stk_print (struct out_stocklev_struct *stocklev_ptr,
    struct in_stocklev_struct *in_stocklev_ptr,
    char *filename,
    char *msg);

#ifdef __cplusplus
}
#endif

#endif // __TPCCDBG_H
Src.Common/Makefile

#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#####
#####
#
# Makefile - Makefile for Src.Common
#
!include $(TPCC_ROOT)/Makefile.config

```

```

#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####
BND_OPTS =      GRANT PUBLIC \
                MESSAGES $*.bnd.msg
PRP_OPTS =      BINDFILE \
                OPTLEVEL 1 \
                ISOLATION RR \
                MESSAGES $*.prep.msg \
                LEVEL $(TPCC_VERSION) \
                NOLINEMACRO
INCLUDES =      -I$(TPCC_SQLLIB)$(SLASH)include
                -I$(TPCC_ROOT)$(SLASH)include
CFLAGS =        $(CFLAGS_OS) $(CFLAGS_DEBUG) $(INCLUDES) \
                -DSQLA_NOLINES -D$(DB2EDITION)
                -D$(DB2VERSION) \
                -D$(TPCC_SPTYPE)
UTIL_OBJ =      tpcctdbg$(OBJEXT) tpcctcx$(OBJEXT)
#
#####
#####
# User Targets
#
#####
all:            connect $(UTIL_OBJ) disconnect
clean:          - $(ERASE) *$(OBJEXT) *.bnd *.msg tpcctcx.c
#
#####
#####
# Helper Targets
#
#####
connect:        - db2 connect to $(TPCC_DBNAME)
disconnect:     - db2 connect reset
                - db2 terminate

rebind:        db2 bind tpcctcx.bnd $(BND_OPTS)
#
#####
#####
# Build Rules
#
#####
.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc
.sqc.c:        @echo "Prepping $*.sqc"
                -db2 prep $*.sqc $(PRP_OPTS)
                @echo "Binding $*.bnd"
                db2 bind $*.bnd $(BND_OPTS)
#
#####
#####
# Dependencies
#
#####
#####

```

```
# Source
tpccdbg$(OBJEXT): tpccdbg.c
tpccctx$(OBJEXT): tpccctx.c
tpccmisc$(OBJEXT): tpccmisc.c
# Headers
tpccdbg.c: $(TPCC_ROOT)/include/db2tpcc.h
```

Src.Common/tpccctx.sqc

```

/*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****/
/*****/
/*
 * tpccctx.sqc - TPCC context code
 */
#include <stdlib.h>
#include <stdio.h>
#include <sqlutil.h>
#include "db2tpcc.h"
int connect_to_TM(char *in_dbname);
int connect_to_TM_auth(char *in_dbname, char *in_username, char
*in_password);
int disconnect_from_TM(void);
int create_context();
int destroy_context();
int attach_context(void*);
int detach_context(void*);
int get_context(void**);
int connect_to_TM(char *in_dbname)
{
    return connect_to_TM_auth(in_dbname, "", "");
}
int connect_to_TM_auth(char *in_dbname, char *in_username, char
*in_password)
{
    SQL_STRUCTURE sqlca sqlca;
    int ConnectSQLCODE = 0;
    EXEC SQL BEGIN DECLARE SECTION;
    char dbname[9];
    char username[129];
    char password[15];
    EXEC SQL END DECLARE SECTION;
    SQLCODE = create_context();
    if (SQLCODE != 0) { return SQLCODE; }
    strncpy(dbname, in_dbname, 8);
    if (strcmp(in_username, "") == 0)
    {
        EXEC SQL CONNECT TO :dbname IN SHARE MODE;
    } else {
        strncpy(username, in_username, 128);
        strncpy(password, in_password, 14);
        EXEC SQL CONNECT TO :dbname IN SHARE MODE USER :username
USING :password;
    }
    ConnectSQLCODE = SQLCODE;
    if (ConnectSQLCODE != 0)

```

```

{
    sqlerror( CLIENT_SQL, "CONNECT", __FILE__, __LINE__, &sqlca);
    SQLCODE = destroy_context();
    if (SQLCODE != 0) { return SQLCODE; }
    return ConnectSQLCODE;
}
}
return 0;
}
int disconnect_from_TM(void)
{
    SQL_STRUCTURE sqlca sqlca;
    int DisconnectSQLCODE = 0;
    EXEC SQL CONNECT RESET;
    DisconnectSQLCODE = SQLCODE;
    if (DisconnectSQLCODE != 0) {
        sqlerror( CLIENT_SQL, "DISCONNECT", __FILE__, __LINE__, &sqlca);
    }
    SQLCODE = destroy_context();

    if (SQLCODE != 0) { return SQLCODE; }
    if (DisconnectSQLCODE) {
        return DisconnectSQLCODE;
    }
    return 0;
}
int create_context(void)
{
    SQL_STRUCTURE sqlca sqlca;
    void *ctx;
    sqlcSetTypeCtx(SQL_CTX_MULTI_MANUAL);
    sqlcBeginCtx(&ctx, SQL_CTX_BEGIN_ALL, NULL, &sqlca);
    if (SQLCODE != 0) {
        sqlerror( CLIENT_SQL, "CREATE", __FILE__, __LINE__, &sqlca);
        return SQLCODE;
    }
    return 0;
}
int attach_context(void *ctx)
{
    SQL_STRUCTURE sqlca sqlca;
    sqlcAttachToCtx(ctx, NULL, &sqlca);
    if (SQLCODE != 0) {
        sqlerror( CLIENT_SQL, "ATTACH", __FILE__, __LINE__, &sqlca);
        return SQLCODE;
    }
    return 0;
}
int detach_context(void *ctx)
{
    SQL_STRUCTURE sqlca sqlca;

    sqlcDetachFromCtx(ctx, NULL, &sqlca);
    if (SQLCODE != 0) {
        sqlerror( CLIENT_SQL, "DETACH", __FILE__, __LINE__, &sqlca);
        return SQLCODE;
    }
    return 0;
}
int destroy_context(void)
{
    SQL_STRUCTURE sqlca sqlca;
    void *ctx;
    SQLCODE = get_context(&ctx);
    if (SQLCODE) { return SQLCODE; }
    sqlcEndCtx(&ctx, SQL_CTX_END_ALL, NULL, &sqlca);
    if (SQLCODE != 0) {

```

```

    sqlerror( CLIENT_SQL, "DESTROY", __FILE__, __LINE__, &sqlca);
    return SQLCODE;
}
return 0;
}
int get_context(void **ctx)
{
    SQL_STRUCTURE sqlca sqlca;
    sqlGetCurrentCtx(ctx, NULL, &sqlca);
    if (SQLCODE != 0) {
        sqlerror( CLIENT_SQL, "GETCTX", __FILE__, __LINE__, &sqlca);
        return SQLCODE;
    }
    return 0;
}

```

Src.Common/tpccdbg.c

```

/*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****
*****
*/
/*
 * tcdbg.c - Debugging Routines
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <time.h>
#include "sqlca.h"
#include "sql.h"
#include "db2tpcc.h"
#include "tpccdbg.h"
#define DEBUG_FILENAME_SZ 128
#define DEBUG_PATH_SIZE 128
void del_print();
void new_print();
void ord_print();
void pay_print();
void stk_print();
void current_tmstamp(char *buf);
static int debugInit = 0;
static char debugPath[DEBUG_PATH_SIZE] = "";
/*-----*/
/* InitializeDebug */
/*-----*/
inline void InitializeDebug(void) {
    if (debugInit == 0) {
        char *p = getenv("TPCC_DEBUGDIR");
        if (p) {
            strncpy(debugPath, p, DEBUG_PATH_SIZE);
        } else {
            strcpy(debugPath, "C:\\temp");
        }
        strcat(debugPath, "\\");
    }
    debugInit = 1;
}

```

```

}
/*-----*/
/* sqlerror */
/*-----*/
void sqlerror(int tranType, char *msg, char *file, int line, SQL_STRUCTURE
sqlca *psqlca)
{
    FILE *err_fp = NULL;
    char err_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];
    char tranName[16];
    int j,k;
    char timeStamp[27];
    char errStr[512] = "";
    InitializeDebug();
    strncpy(err_fn, debugPath, DEBUG_PATH_SIZE);
    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;
    switch(tranType)
    {
        case NEWORD_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "new.err.out");
            strcpy(tranName, "NEW_ORDER");
            break;
        case DELIVERY_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "del.err.out");
            strcpy(tranName, "DELIVERY");
            break;
        case PAYMENT_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "pay.err.out");
            strcpy(tranName, "PAYMENT");
            break;
        case ORDSTAT_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "ord.err.out");
            strcpy(tranName, "ORDER_STAT");
            break;
        case STOCKLEV_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "stk.err.out");
            strcpy(tranName, "STOCK_LVL");
            break;
        case 0:
            strcat(err_fn, "cli.err.out");
            strcpy(tranName, "CLIENT");
            break;
        default:
            return;
    }
    /* Generate Formatted Error Message */
    sqlaintp(errStr, 512, 78, psqlca);
    err_fp = fopen(err_fn, "a+");
    fprintf(err_fp, "-----\n");
    fprintf(err_fp, "Transaction: %s (%s)\n", tranName, msg);
    fprintf(err_fp, "FILE %s (%u)\n", file, line);
    fprintf(err_fp, "SQLCODE %d ", psqlca->sqlcode);
    fprintf(err_fp, "TIME %s\n", timeStamp);
    fprintf(err_fp, "-----\n");
    fprintf(err_fp, "%s", errStr);
    fprintf(err_fp, "-----\n");
    if (psqlca->sqlerrmc[0] != ' ' || psqlca->sqlerrmc[1] != ' ')
    {
        fprintf(err_fp, "slerrmc: ");
        for(j = 0; j < 5; j++)
        {
            for(k = 0; k < 16; k++) {

```

```

int pos = j * 16 + k;
if (pos < 70) fprintf(err_fp, "%02x ", psqlca->sqlerrmc[pos]);
else fprintf(err_fp, " ");
}
fprintf(err_fp, "|");
for(k = 0; k < 16; k++) {
int pos = j * 16 + k;
char c = ' ';
if (pos < 70) {
c = psqlca->sqlerrmc[pos];
if (!isprint(c)) c = ' ';
}
fprintf(err_fp, "%c", c);
}
fprintf(err_fp, "\n");
if (j < 4) fprintf(err_fp, " ");
}
}
fprintf(err_fp, "sqlerrp: ");
for(j = 0; j < 8; j++)
fprintf(err_fp, "%c", psqlca->sqlerrp[j]);
fprintf(err_fp, "\n");
fprintf(err_fp, "sqlerrd: ");
for(j = 0; j < 6; j++)
fprintf(err_fp, " %d", psqlca->sqlerrd[j]);
fprintf(err_fp, "\n");
if (psqlca->sqlwarn[0] != ' ')
{
fprintf(err_fp, "sqlwarn: ");
for(j = 0; j < 8; j++)
fprintf(err_fp, "%c ", psqlca->sqlwarn[j]);
fprintf(err_fp, "\n");
}
fprintf(err_fp, "\n");
fclose(err_fp);
}

/*-----*/
/* del_debug */
/*-----*/
void del_debug (struct out_delivery_struct *delivery_ptr,
struct in_delivery_struct *in_delivery,
char *msg)
{
char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

InitializeDebug();
strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
strcat(debug_fn, "del.debug.out");
del_print(delivery_ptr, in_delivery, debug_fn, msg);
}
/*-----*/
/* del_print */
/*-----*/
void del_print (struct out_delivery_struct *delivery_ptr,
struct in_delivery_struct *in_delivery,
char *filename,
char *msg)
{
FILE *debug_fp;
char timeStamp[27];
int j;
current_tmstamp(&timeStamp[0]);
timeStamp[19] = (char)NULL;
if ((debug_fp = fopen(filename, "a+")) == NULL)
{
return;
}
}

```

```

fprintf(debug_fp, "Delivery debug information follows %s (%s)\n",
timeStamp, msg);

fprintf(debug_fp, "\n=====
=====\\n");
fprintf(debug_fp, "in_delivery_struct {\n");
fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
in_delivery->s_W_ID, in_delivery->s_W_ID);
fprintf(debug_fp, "ts_O_CARRIER_ID = %d (%X)\n",
in_delivery->s_O_CARRIER_ID, in_delivery->s_O_CARRIER_ID);
fprintf(debug_fp, "ts_O_DELIVERY_D = %lld (%lX)\n",
in_delivery->s_O_DELIVERY_D_time,
in_delivery->s_O_DELIVERY_D_time);
fprintf(debug_fp, "}\n");
fprintf(debug_fp, "out_delivery_struct {\n");
fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
delivery_ptr->s_transtatus, delivery_ptr->s_transtatus);
fprintf(debug_fp, "tdeadlocks = %d (%X)\n",
delivery_ptr->deadlocks, delivery_ptr->deadlocks);
for (j = 0; j < 10; j++) {
fprintf(debug_fp, "\tts_O_ID[%d] = %d\n",
j, delivery_ptr->s_O_ID[j]);
}
fprintf(debug_fp, "\t}\n");
fclose(debug_fp);
}

/*-----*/
/* new_debug */
/*-----*/
void new_debug (struct out_neword_struct *neword_ptr,
struct in_neword_struct *in_neword,
char *msg)
{
char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

InitializeDebug();
strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
strcat(debug_fn, "new.debug.out");
new_print(neword_ptr, in_neword, debug_fn, msg);
}
/*-----*/
/* new_print */
/*-----*/
void new_print (struct out_neword_struct *neword_ptr,
struct in_neword_struct *in_neword,
char *filename,
char *msg)
{
FILE *debug_fp;
char timeStamp[27];
int j, items;
current_tmstamp(&timeStamp[0]);
timeStamp[19] = (char)NULL;
if ((debug_fp = fopen(filename, "a+")) == NULL)
{
return;
}
fprintf(debug_fp, "New order debug information follows %s (%s)\n",
timeStamp, msg);

fprintf(debug_fp, "\n=====
=====\\n");
fprintf(debug_fp, "in_neword_struct {\n");
fprintf(debug_fp, "ts_C_ID = %d (%X)\n",
in_neword->s_C_ID, in_neword->s_C_ID);
fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
in_neword->s_W_ID, in_neword->s_W_ID);
}

```



```

fprintf(debug_fp, "ts_D_ID      = %d (%X)\n",
        in_neword->s_D_ID, in_neword->s_D_ID);
fprintf(debug_fp, "ts_O_OL_CNT    = %d (%X)\n",
        in_neword->s_O_OL_CNT, in_neword->s_O_OL_CNT);
fprintf(debug_fp, "ts_all_local  = %d (%X)\n",
        in_neword->s_all_local, in_neword->s_all_local);
fprintf(debug_fp, "ts_O_ENTRY_D  = %lld (%lX)\n",
        in_neword->s_O_ENTRY_D_time, in_neword->s_O_ENTRY_D_time);
// fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
//         in_neword->s_transtatus, in_neword->s_transtatus);
// fprintf(debug_fp, "tduplicate_items= %d (%X)\n",
//         in_neword->duplicate_items, in_neword->duplicate_items);
fprintf(debug_fp, "titems {n}");
items = in_neword->s_O_OL_CNT;
for (j=0; j<items; j++) {
    if(j != 0)
        fprintf(debug_fp, "\n");
    fprintf(debug_fp, "ts_OL_I_ID[%d]    = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_I_ID,
            in_neword->in_item[j].s_OL_I_ID);
    fprintf(debug_fp, "ts_OL_SUPPLY_W_ID[%d] = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_SUPPLY_W_ID,
            in_neword->in_item[j].s_OL_SUPPLY_W_ID);
    fprintf(debug_fp, "ts_OL_QUANTITY[%d]  = %d (%X)\n",

            j, in_neword->in_item[j].s_OL_QUANTITY,
            in_neword->in_item[j].s_OL_QUANTITY);
}
fprintf(debug_fp, "t}\n}\n");
fprintf(debug_fp, "out_neword_struct {n}");
fprintf(debug_fp, "ts_C_LAST      = %s\n",
        neword_ptr->s_C_LAST);
fprintf(debug_fp, "ts_C_CREDIT   = %s\n",
        neword_ptr->s_C_CREDIT);
fprintf(debug_fp, "ts_W_TAX      = %d\n",
        neword_ptr->s_W_TAX);
fprintf(debug_fp, "ts_D_TAX      = %d\n",
        neword_ptr->s_D_TAX);
fprintf(debug_fp, "ts_C_DISCOUNT = %d\n",
        neword_ptr->s_C_DISCOUNT);
fprintf(debug_fp, "ts_O_ID       = %d (%X)\n",
        neword_ptr->s_O_ID, neword_ptr->s_O_ID);
fprintf(debug_fp, "ts_O_OL_CNT   = %d (%X)\n",
        neword_ptr->s_O_OL_CNT, neword_ptr->s_O_OL_CNT);
fprintf(debug_fp, "ts_O_ENTRY_D  = %lld (%lX)\n",
        neword_ptr->s_O_ENTRY_D_time,
        neword_ptr->s_O_ENTRY_D_time);
neword_ptr->s_O_ENTRY_D_time);
fprintf(debug_fp, "ts_total_amount = %d\n",
        neword_ptr->s_total_amount);
fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
        neword_ptr->s_transtatus, neword_ptr->s_transtatus);
fprintf(debug_fp, "tdeadlocks   = %d (%X)\n",
        neword_ptr->deadlocks, neword_ptr->deadlocks);
// fprintf(debug_fp, "ts_W_ID      = %d (%X)\n",
//         neword_ptr->s_W_ID, neword_ptr->s_W_ID);
// fprintf(debug_fp, "ts_D_ID      = %d (%X)\n",
//         neword_ptr->s_D_ID, neword_ptr->s_D_ID);
// fprintf(debug_fp, "ts_all_local  = %d (%X)\n",
//         neword_ptr->s_all_local, neword_ptr->s_all_local);
// fprintf(debug_fp, "tduplicate_items= %d (%X)\n",
//         neword_ptr->duplicate_items, neword_ptr->duplicate_items);
fprintf(debug_fp, "titems {n}");
items = neword_ptr->s_O_OL_CNT;
for (j=0; j<items; j++) {
    if(j != 0)
        fprintf(debug_fp, "\n");
    fprintf(debug_fp, "ts_I_NAME[%d]    = %s\n",
            j, neword_ptr->item[j].s_I_NAME);

```

```

fprintf(debug_fp, "ts_I_PRICE[%d]    = %d\n",
        j, neword_ptr->item[j].s_I_PRICE);
fprintf(debug_fp, "ts_OL_AMOUNT[%d]  = %d\n",
        j, neword_ptr->item[j].s_OL_AMOUNT);
fprintf(debug_fp, "ts_S_QUANTITY[%d]  = %d (%X)\n",
        j, neword_ptr->item[j].s_S_QUANTITY,
        neword_ptr->item[j].s_S_QUANTITY);
fprintf(debug_fp, "ts_brand_generic[%d] = %c\n",
        j, neword_ptr->item[j].s_brand_generic);
}
fprintf(debug_fp, "t}\n}\n");
fclose(debug_fp);
}

/*-----*/
/* ord_debug */
/*-----*/
void ord_debug (struct out_ordstat_struct *ordstat_ptr,
               struct in_ordstat_struct *in_ordstat,
               char *msg)
{
    char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
    strcat(debug_fn, "ord.debug.out");
    ord_print(ordstat_ptr, in_ordstat, debug_fn, msg);
}

/*-----*/
/* ord_print */
/*-----*/
void ord_print (struct out_ordstat_struct *ordstat_ptr,
               struct in_ordstat_struct *in_ordstat,
               char *filename,
               char *msg)
{
    FILE *debug_fp;
    char timeStamp[27];
    int j, items;
    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;
    if ((debug_fp = fopen(filename, "a+")) == NULL)
    {
        return;
    }
    fprintf(debug_fp, "Order status debug information follows %s (%s)\n",
            timeStamp, msg);

    fprintf(debug_fp, "\n=====");
    fprintf(debug_fp, "in_ordstat_struct {n}");
    fprintf(debug_fp, "ts_W_ID      = %d (%X)\n",
            in_ordstat->s_W_ID, in_ordstat->s_W_ID);
    fprintf(debug_fp, "ts_D_ID      = %d (%X)\n",
            in_ordstat->s_D_ID, in_ordstat->s_D_ID);
    fprintf(debug_fp, "ts_C_ID      = %d (%X)\n",
            in_ordstat->s_C_ID, in_ordstat->s_C_ID);
    fprintf(debug_fp, "ts_C_LAST    = %s\n",
            in_ordstat->s_C_LAST);

    fprintf(debug_fp, "}\n");
    fprintf(debug_fp, "out_ordstat_struct {n}");
    fprintf(debug_fp, "ts_C_ID      = %d (%X)\n",
            ordstat_ptr->s_C_ID, ordstat_ptr->s_C_ID);
    fprintf(debug_fp, "ts_C_FIRST   = %s\n",
            ordstat_ptr->s_C_FIRST);
    fprintf(debug_fp, "ts_C_MIDDLE  = %s\n",

```

```

    ordstat_ptr->s_C_MIDDLE);
fprintf(debug_fp, "\ts_C_LAST = %s\n",
        ordstat_ptr->s_C_LAST);
fprintf(debug_fp, "\ts_C_BALANCE = %lld\n",
        ordstat_ptr->s_C_BALANCE);
fprintf(debug_fp, "\ts_O_ID = %d (%X)\n",
        ordstat_ptr->s_O_ID, ordstat_ptr->s_O_ID);
fprintf(debug_fp, "\ts_O_ENTRY_D = %lld (%lX)\n",
        ordstat_ptr->s_O_ENTRY_D_time, ordstat_ptr->s_O_ENTRY_D_time);
fprintf(debug_fp, "\ts_O_CARRIER_ID = %d (%X)\n",
        ordstat_ptr->s_O_CARRIER_ID, ordstat_ptr->s_O_CARRIER_ID);
fprintf(debug_fp, "\ts_ol_cnt = %d (%X)\n",
        ordstat_ptr->s_ol_cnt, ordstat_ptr->s_ol_cnt);
fprintf(debug_fp, "\ts_transtatus = %d (%X)\n",
        ordstat_ptr->s_transtatus, ordstat_ptr->s_transtatus);
fprintf(debug_fp, "\ts_deadlocks = %d (%X)\n",
        ordstat_ptr->s_deadlocks, ordstat_ptr->s_deadlocks);
fprintf(debug_fp, "\titems {\n");
items = ordstat_ptr->s_ol_cnt;
for (j = 0; j < items; j++) {
    if (j != 0)
        fprintf(debug_fp, "\n");
    fprintf(debug_fp, "\ts_OL_SUPPLY_W_ID[%d] = %d (%X)\n",
            j, ordstat_ptr->item[j].s_OL_SUPPLY_W_ID,
            ordstat_ptr->item[j].s_OL_SUPPLY_W_ID);
    fprintf(debug_fp, "\ts_OL_I_ID[%d] = %d (%X)\n",
            j, ordstat_ptr->item[j].s_OL_I_ID, ordstat_ptr->item[j].s_OL_I_ID);
    fprintf(debug_fp, "\ts_OL_QUANTITY[%d] = %d (%X)\n",
            j, ordstat_ptr->item[j].s_OL_QUANTITY,
            ordstat_ptr->item[j].s_OL_QUANTITY);
    fprintf(debug_fp, "\ts_OL_AMOUNT[%d] = %d\n",
            j, ordstat_ptr->item[j].s_OL_AMOUNT);
    fprintf(debug_fp, "\ts_OL_DELIVERY_D[%d] = %lld (%lX)\n",
            j, ordstat_ptr->item[j].s_OL_DELIVERY_D_time,
            ordstat_ptr->item[j].s_OL_DELIVERY_D_time);
}
fprintf(debug_fp, "\t}\n");
fclose(debug_fp);
}

/*-----*/
/* pay_debug */
/*-----*/
void pay_debug (struct out_payment_struct *payment_ptr,
               struct in_payment_struct *in_payment,
               char *msg)
{
    char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
    strcat(debug_fn, "pay.debug.out");
    pay_print(payment_ptr, in_payment, debug_fn, msg);
}

/*-----*/
/* pay_print */
/*-----*/
void pay_print (struct out_payment_struct *payment_ptr,
               struct in_payment_struct *in_payment,
               char *filename,
               char *msg)
{
    FILE *debug_fp;
    char timeStamp[27];
    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;

```

```

if ((debug_fp = fopen(filename, "a+")) == NULL)
{
    return;
}
fprintf(debug_fp, "Payment debug information follows %s (%s)\n",
        timeStamp, msg);

fprintf(debug_fp, "\n=====
=====");
fprintf(debug_fp, "in_payment_struct {\n");
fprintf(debug_fp, "\ts_H_AMOUNT = %lld (%lX)\n",
        in_payment->s_H_AMOUNT, in_payment->s_H_AMOUNT);
fprintf(debug_fp, "\ts_C_ID = %d (%X)\n",
        in_payment->s_C_ID, in_payment->s_C_ID);
fprintf(debug_fp, "\ts_W_ID = %d (%X)\n",
        in_payment->s_W_ID, in_payment->s_W_ID);
fprintf(debug_fp, "\ts_D_ID = %d (%X)\n",
        in_payment->s_D_ID, in_payment->s_D_ID);
fprintf(debug_fp, "\ts_C_D_ID = %d (%X)\n",
        in_payment->s_C_D_ID, in_payment->s_C_D_ID);
fprintf(debug_fp, "\ts_C_W_ID = %d (%X)\n",
        in_payment->s_C_W_ID, in_payment->s_C_W_ID);
fprintf(debug_fp, "\ts_C_LAST = %s\n",
        in_payment->s_C_LAST);
fprintf(debug_fp, "\ts_H_DATE = %lld (%lX)\n",
        in_payment->s_H_DATE_time, in_payment->s_H_DATE_time);
fprintf(debug_fp, "\n}\n");
fprintf(debug_fp, "out_payment_struct {\n");
fprintf(debug_fp, "\ts_H_DATE = %lld (%lX)\n",
        in_payment->s_H_DATE_time, in_payment->s_H_DATE_time);
fprintf(debug_fp, "\ts_C_CREDIT_LIM = %lld\n",
        payment_ptr->s_C_CREDIT_LIM);
fprintf(debug_fp, "\ts_C_DISCOUNT = %d\n",
        payment_ptr->s_C_DISCOUNT);
fprintf(debug_fp, "\ts_C_BALANCE = %lld\n",
        payment_ptr->s_C_BALANCE);
fprintf(debug_fp, "\ts_C_ID = %d (%X)\n",
        payment_ptr->s_C_ID, payment_ptr->s_C_ID);
fprintf(debug_fp, "\ts_W_STREET_1 = %s\n",
        payment_ptr->s_W_STREET_1);
fprintf(debug_fp, "\ts_W_STREET_2 = %s\n",
        payment_ptr->s_W_STREET_2);
fprintf(debug_fp, "\ts_W_CITY = %s\n",
        payment_ptr->s_W_CITY);
fprintf(debug_fp, "\ts_W_STATE = %s\n",
        payment_ptr->s_W_STATE);
fprintf(debug_fp, "\ts_W_ZIP = %s\n",
        payment_ptr->s_W_ZIP);
fprintf(debug_fp, "\ts_D_STREET_1 = %s\n",
        payment_ptr->s_D_STREET_1);
fprintf(debug_fp, "\ts_D_STREET_2 = %s\n",
        payment_ptr->s_D_STREET_2);
fprintf(debug_fp, "\ts_D_CITY = %s\n",
        payment_ptr->s_D_CITY);
fprintf(debug_fp, "\ts_D_STATE = %s\n",
        payment_ptr->s_D_STATE);
fprintf(debug_fp, "\ts_D_ZIP = %s\n",
        payment_ptr->s_D_ZIP);
fprintf(debug_fp, "\ts_C_FIRST = %s\n",
        payment_ptr->s_C_FIRST);
fprintf(debug_fp, "\ts_C_MIDDLE = %s\n",
        payment_ptr->s_C_MIDDLE);
fprintf(debug_fp, "\ts_C_LAST = %s\n",
        payment_ptr->s_C_LAST);
fprintf(debug_fp, "\ts_C_STREET_1 = %s\n",
        payment_ptr->s_C_STREET_1);
fprintf(debug_fp, "\ts_C_STREET_2 = %s\n",
        payment_ptr->s_C_STREET_2);

```

```

fprintf(debug_fp, "ts_C_CITY    = %s\n",
        payment_ptr->s_C_CITY);
fprintf(debug_fp, "ts_C_STATE   = %s\n",
        payment_ptr->s_C_STATE);
fprintf(debug_fp, "ts_C_ZIP    = %s\n",
        payment_ptr->s_C_ZIP);
fprintf(debug_fp, "ts_C_PHONE   = %s\n",
        payment_ptr->s_C_PHONE);
fprintf(debug_fp, "ts_C_SINCE   = %lld (%lX)\n",
        payment_ptr->s_C_SINCE_time, payment_ptr->s_C_SINCE_time);
fprintf(debug_fp, "ts_C_CREDIT  = %s\n",
        payment_ptr->s_C_CREDIT);
fprintf(debug_fp, "ts_C_DATA    = %s\n",
        payment_ptr->s_C_DATA);
fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
        payment_ptr->s_transtatus, payment_ptr->s_transtatus);
fprintf(debug_fp, "tdeadlocks  = %d (%X)\n",
        payment_ptr->deadlocks, payment_ptr->deadlocks);
fprintf(debug_fp, "\n\n");
fclose(debug_fp);
}

```

```

/*-----*/
/* stk_debug          */
/*-----*/
void stk_debug (struct out_stocklev_struct *stocklev,
               struct in_stocklev_struct *in_stocklev,
               char *msg)

```

```

{
    char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
    strcat(debug_fn, "stk.debug.out");
    stk_print(stocklev, in_stocklev, debug_fn, msg);
}

```

```

/*-----*/
/* stk_print          */
/*-----*/

```

```

void stk_print (struct out_stocklev_struct *stocklev,
               struct in_stocklev_struct *in_stocklev,
               char *filename,
               char *msg)
{
    FILE *debug_fp;
    char timeStamp[27];
    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;
    if ((debug_fp = fopen(filename, "a+")) == NULL)
    {
        return;
    }
    fprintf(debug_fp, "Stock level debug information follows %s (%s)\n",
            timeStamp, msg);
}

```

```

fprintf(debug_fp, "\n=====
\n");
fprintf(debug_fp, "in_stocklev_struct {\n");
fprintf(debug_fp, "ts_W_ID    = %d (%X)\n",
        in_stocklev->s_W_ID, in_stocklev->s_W_ID);
fprintf(debug_fp, "ts_D_ID    = %d (%X)\n",
        in_stocklev->s_D_ID, in_stocklev->s_D_ID);
fprintf(debug_fp, "ts_threshold = %d (%X)\n",
        in_stocklev->s_threshold, in_stocklev->s_threshold);
fprintf(debug_fp, "}\n\n");
fprintf(debug_fp, "out_stocklev_struct {\n");
fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
        stocklev->s_transtatus, stocklev->s_transtatus);

```

```

fprintf(debug_fp, "tdeadlocks  = %d (%X)\n",
        stocklev->deadlocks, stocklev->deadlocks);
fprintf(debug_fp, "ts_low_stock = %d (%X)\n",
        stocklev->s_low_stock, stocklev->s_low_stock);
fprintf(debug_fp, "}\n\n");
fclose(debug_fp);
}
void current_tmstamp(char *buf)
{
    time_t t = time(NULL);
    strncpy(buf, ctime(&t), 19);
}

```

Src.Cli/Makefile

```

#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#####
#
# Makefile - Makefile for Src.Cli (RTE/Driver Interface)
#
!include $(TPCC_ROOT)/Makefile.config
#
#####
# Preprocessor, Compiler and Linker Flags
#
#####
BND_OPTS =      GRANT PUBLIC \
                MESSAGES $*.bnd.msg
PRP_OPTS =      BINDFILE \
                ISOLATION RR \
                EXPLAIN ALL \
                MESSAGES $*.prep.msg \
                LEVEL $(TPCC_VERSION) \
                NOLINEMACRO
INCLUDES =      -I$(TPCC_SQLLIB)/include -I$(TPCC_ROOT)/include
CFLAGS =        $(CFLAGS_OS) $(INCLUDES) $(CFLAGS_DEBUG) \
                $(UOPTS) -D$(DB2EDITION) -D$(DB2VERSION)
-D$(TPCC_SPTYPE)
OBJS =          $(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT) \
                $(TPCC_ROOT)/Src.Common/tpccdbg$(OBJEXT) \
                $(TPCC_ROOT)/Src.Common/tpccctx$(OBJEXT) \
                tpcccli$(OBJEXT)
LIBS =          tpcccli$(LIBEXT)
#
#####
# User Targets
#
#####
all:            connect $(OBJS) plan $(LIBS) disconnect
                $(AR) $(ARFLAGS) $(ARFLAGS_OUT)tpcccli$(LIBEXT)
$(OBJS) $(ARFLAGS_LIB)
                @echo "-----"
                @echo "Please copy lval.h, db2tpcc.h, and tpcccli$(LIBEXT) to"

```

```

@echo "a place where they can be #included and linked with the"
@echo "RTE/driver code."
@echo "-----"
clean:
- $(ERASE) *.msg *.bnd *.plan *$(OBJEXT) *$(LIBEXT) tpccli.c
#
#####
#####
# Helper Targets
#
#####
#####
connect:
- db2 connect to $(TPCC_DBNAME)
disconnect:
- db2 connect reset
- db2 terminate
plan:
- db2exfmt -d $(TPCC_DBNAME) -e $(TPCC_SCHEMA) -s
$(TPCC_SCHEMA) -w -l -n TPCCLLI -g -# 0 -o TPCCLLI.exfmt.plan
- db2expln -d $(TPCC_DBNAME) -c $(TPCC_SCHEMA) -p
TPCCCLI -s 0 -g -o TPCCLLI.expln.plan
rebind: connect
db2 bind tpccli.bnd $(BND_OPTS) QUERYOPT 7
#
#####
#####
# Build Rules
#
#####
#####
.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc
tpccli.c:
@echo "Prepping $*.sqc"
-db2 prep $*.sqc $(PRP_OPTS) ISOLATION RR
@echo "Binding $*.bnd"
db2 bind $*.bnd $(BND_OPTS) QUERYOPT 7
#
#####
#####
# Dependencies
#
#####
#####
# Client Library:
tpccli$(LIBEXT): $(OBJS)
# Source
tpcc_all_sql$(OBJEXT): tpcc_all_sql.c
# Headers
tpcc_all_sql.c: $(TPCC_ROOT)/include/db2tpcc.h
$(TPCC_ROOT)/include/lval.h

```

Src.Cli/tpccli.sqc

```

/*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

```

```

*****
*****/
/*
* tpccli.sqc - Client/Server code for TPCC
*/
#include <stdlib.h>
#include <errno.h>
#include "db2tpcc.h"
#include "tpccapp.h"
#include "tpccdbg.h"
#include "sqlca.h"
#include "sql.h"
#include "lval.h"
// -----
// New Order CLIENT
// -----
static int itemComparison ( const void * a , const void * b )
{
struct in_items_struct * one = (struct in_items_struct *) a ;
struct in_items_struct * two = (struct in_items_struct *) b ;
// If diff item id then sort on that.
// If real/quasi dup, then sort on warehouse id.
if ( one->s_OL_I_ID != two->s_OL_I_ID )
{
return ( one->s_OL_I_ID - two->s_OL_I_ID ) ;
}
else
{
return ( one->s_OL_SUPPLY_W_ID - two->s_OL_SUPPLY_W_ID ) ;
}
}
int neword_sql ( struct in_neword_struct * in_neword
, struct out_neword_struct * neword )
{
struct sqlca sqlca ;
EXEC SQL BEGIN DECLARE SECTION;
struct vc_new_in
{
short len;
char data[ 270 ] ;
} * pHostvarInput ;
struct vc_new_out
{
short len;
char data[ 662 ] ;
} * pHostvarOutput ;
EXEC SQL END DECLARE SECTION;
int clientRc = TRAN_OK ;
int itemIndex = 0 ;
int actualItemIndex = 0 ;
/* Create Timestamp */
in_neword->s_O_ENTRY_D_time = time(NULL) ;
// Sort the item list ; This helps eliminate duplicates anyway, and since invalid
item
// IDs == 100001 , we will remain compliant with 2.4.2.3 Comment 1.
// !! So DON'T sort or change the order of the items in any subsequent code,
including SQL.
in_neword->s_all_local = 1 ;
for ( itemIndex = 0 ;
itemIndex < 15 && in_neword->in_item[ itemIndex ].s_OL_I_ID !=
UNUSED_ITEM_ID ;
itemIndex++
)
{
if ( in_neword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID !=
in_neword->s_W_ID )
{
in_neword->s_all_local = 0 ;

```

```

}
}
// Pre-duplicate processing count
in_newword->s_O_OL_CNT = itemIndex ;
// Sort the original array
qsort( in_newword->in_item, in_newword->s_O_OL_CNT
      , sizeof ( in_newword->in_item[ 0 ] )
      , itemComparison
      );
// Now purge the duplicates.
actualItemIndex = -1 ;
for ( itemIndex = 0
      ; itemIndex < in_newword->s_O_OL_CNT
      ; itemIndex++ )
{
    // If duplicate, just increment the item entry order count
    if ( actualItemIndex >= 0
        && in_newword->in_item[ actualItemIndex ].s_OL_I_ID ==
in_newword->in_item[ itemIndex ].s_OL_I_ID
        && in_newword->in_item[ actualItemIndex ].s_OL_SUPPLY_W_ID ==
in_newword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID
        )
    {
        in_newword->in_item[ actualItemIndex ].s_OL_QUANTITY +=
in_newword->in_item[ itemIndex ].s_OL_QUANTITY ;
    }
    else
    {
        actualItemIndex ++ ;
        in_newword->in_item[ actualItemIndex ].s_OL_I_ID =
in_newword->in_item[ itemIndex ].s_OL_I_ID ;
        in_newword->in_item[ actualItemIndex ].s_OL_SUPPLY_W_ID =
in_newword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID ;
        in_newword->in_item[ actualItemIndex ].s_OL_QUANTITY =
in_newword->in_item[ itemIndex ].s_OL_QUANTITY ;
    }
}
in_newword->s_O_OL_CNT = actualItemIndex + 1 ;
pHostvarInput = (struct vc_new_in *) in_newword ;
pHostvarInput->len = sizeof(struct in_newword_struct) -
SPGENERAL_ADJUST ;
pHostvarOutput = (struct vc_new_out *) newword ;
pHostvarOutput->len = sizeof(struct out_newword_struct) -
SPGENERAL_ADJUST ;
#ifdef DEBUGIT
    new_debug(newword, in_newword, "Client before SP call");
#endif /* DEBUGIT */
#ifdef SWAP_ENDIAN
    for ( itemIndex=0; itemIndex<in_newword->s_O_OL_CNT; itemIndex++)
    {
        SWAP_BYTE(in_newword->in_item[ itemIndex ].s_OL_I_ID);
        SWAP_BYTE(in_newword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID);
        SWAP_BYTE(in_newword->in_item[ itemIndex ].s_OL_QUANTITY);
    }
    SWAP_BYTE(in_newword->s_O_ENTRY_D_time);
    SWAP_BYTE(in_newword->s_C_ID);
    SWAP_BYTE(in_newword->s_W_ID);
    SWAP_BYTE(in_newword->s_D_ID);
    SWAP_BYTE(in_newword->s_O_OL_CNT);
    SWAP_BYTE(in_newword->s_all_local);
    SWAP_BYTE(in_newword->duplicate_items);
#endif //SWAP_ENDIAN
    EXEC SQL CALL news ( :*pHostvarInput, :*pHostvarOutput );
#ifdef SWAP_ENDIAN
    SWAP_BYTE(in_newword->s_O_ENTRY_D_time);
    SWAP_BYTE(in_newword->s_C_ID);
    SWAP_BYTE(in_newword->s_W_ID);
    SWAP_BYTE(in_newword->s_D_ID);

```

```

    SWAP_BYTE(in_newword->s_O_OL_CNT);
    SWAP_BYTE(in_newword->s_all_local);
    SWAP_BYTE(in_newword->duplicate_items);
    for (itemIndex=0; itemIndex<in_newword->s_O_OL_CNT; itemIndex++)
    {
        SWAP_BYTE(in_newword->in_item[ itemIndex ].s_OL_I_ID);
        SWAP_BYTE(in_newword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID);
        SWAP_BYTE(in_newword->in_item[ itemIndex ].s_OL_QUANTITY);
    }
    SWAP_BYTE(newword->s_O_ENTRY_D_time);
    SWAP_BYTE(newword->s_W_TAX);
    SWAP_BYTE(newword->s_D_TAX);
    SWAP_BYTE(newword->s_C_DISCOUNT);
    SWAP_BYTE(newword->s_total_amount);
    SWAP_BYTE(newword->s_O_ID);
    SWAP_BYTE(newword->s_O_OL_CNT);
    SWAP_BYTE(newword->s_transtatus);
    SWAP_BYTE(newword->deadlocks);
    for (itemIndex=0; itemIndex<in_newword->s_O_OL_CNT; itemIndex++)
    {
        SWAP_BYTE(newword->item[ itemIndex ].s_I_PRICE);
        SWAP_BYTE(newword->item[ itemIndex ].s_OL_AMOUNT);
        SWAP_BYTE(newword->item[ itemIndex ].s_S_QUANTITY);
    }
#endif //SWAP_ENDIAN
    if ( sqlca.sqlcode == 0 )
    {
        double wtax = newword->s_W_TAX / 10000.0 ;
        double dtax = newword->s_D_TAX / 10000.0 ;
        double cdisc = newword->s_C_DISCOUNT / 10000.0 ;
        double factor = (1.0 - cdisc) * (1.0 + wtax + dtax) ;
        // Post process the item set, detecting any bad items , and set or count from
that.
        // Anything that could be deferred from the SP to the client has been.
        newword->s_total_amount = 0 ;
        for ( itemIndex = 0 ;
              itemIndex < in_newword->s_O_OL_CNT ; // from input , not output
              itemIndex++
              )
        {
            if ( newword->item[ itemIndex ].s_I_PRICE > 0 ) // A zero price signifies a
bad item
            {
                newword->item[ itemIndex ].s_OL_AMOUNT = newword->item[
itemIndex ].s_I_PRICE *
                    in_newword->in_item[ itemIndex
].s_OL_QUANTITY ; // reference input value

                newword->s_total_amount += newword->item[ itemIndex
].s_OL_AMOUNT ;
            }
        }
        // s_total_amount gets cast implicitly to a double to do the arithmetic,
// and then cast back to a sqlint32.
        newword->s_total_amount *= factor;
    }
    else
    {
        sqlerror( NEWORD_SQL, "NEW", __FILE__, __LINE__, &sqlca );
        newword->s_transtatus = FATAL_SQLERROR ;
        clientRc = FATAL_SQLERROR ;
    }
}
/* Update Output Structure with Timestamp */
newword->s_O_ENTRY_D_time = in_newword->s_O_ENTRY_D_time ;
#ifdef DEBUGIT
    new_debug(newword, in_newword, "Client after SP call");
#endif /* DEBUGIT */

```

```

if(newword->s_transtatus <= FATAL_SQLERROR)
{
    new_debug(newword, in_newword, "NEW failed");
    clientRc = FATAL_SQLERROR ;
}
if(newword->s_transtatus == INVALID_ITEM)
{
    clientRc = INVALID_ITEM ;
}
return ( clientRc ) ;
}
//-----
// Payment CLIENT
//-----
int payment_sql ( struct in_payment_struct * in_payment
                , struct out_payment_struct * payment )
{
    struct sqlca sqlca ;
    int clientRc = TRAN_OK ;
    EXEC SQL BEGIN DECLARE SECTION;
    // Inputs
    sqlint64 h_amount ;
    sqlint32 in_c_id ;
    struct s_data_type { short len ; char data[ 16 ] ; } c_last_input ;
    sqlint32 w_id ;
    sqlint32 c_w_id ;
    short d_id ;
    short c_d_id ;
    sqlint64 h_date;
    // Outputs
    sqlint32 c_id ;
    sqlint64 c_credit_lim ;
    sqlint32 c_discount ;
    sqlint64 c_balance ;
    char w_street_1 [ 20 ] , w_street_2 [ 20 ] ;
    char w_city [ 20 ] , w_state [ 2 ] , w_zip [ 9 ] ;
    char d_street_1 [ 20 ] , d_street_2 [ 20 ] , d_city [ 20 ] ;
    char d_state [ 2 ] , d_zip [ 9 ] , c_first [ 16 ] ;
    char c_last [ 16 ] ;
    char c_middle [ 2 ] , c_street_1 [ 20 ] ;
    char c_street_2 [ 20 ] , c_city [ 20 ] , c_state [ 2 ] ;
    char c_zip [ 9 ] , c_phone [ 16 ] ;
    char c_credit [ 2 ] ;
    sqlint64 c_since ;
    char c_data [ 200 ] ;
    short c_data_indicator = 0 ;
    struct c_data_prefix_c_last_type { short len ; char data[ 28 ] ; }
c_data_prefix_c_last ;
    struct c_data_prefix_c_id_type { short len ; char data[ 34 ] ; }
c_data_prefix_c_id ;

    EXEC SQL END DECLARE SECTION;

    // Input redirects
#define h_amount in_payment->s_H_AMOUNT
#define in_c_id in_payment->s_C_ID
#define w_id in_payment->s_W_ID
#define d_id in_payment->s_D_ID
#define c_d_id in_payment->s_C_D_ID
#define c_w_id in_payment->s_C_W_ID
#define h_date in_payment->s_H_DATE_time
    // Output redirects
#define c_credit_lim payment->s_C_CREDIT_LIM
#define c_discount payment->s_C_DISCOUNT
#define c_balance payment->s_C_BALANCE
#define c_id payment->s_C_ID
#define c_last payment->s_C_LAST
#define c_first payment->s_C_FIRST

```

```

#define c_middle payment->s_C_MIDDLE
#define c_street_1 payment->s_C_STREET_1
#define c_street_2 payment->s_C_STREET_2
#define c_city payment->s_C_CITY
#define c_state payment->s_C_STATE
#define c_zip payment->s_C_ZIP
#define c_phone payment->s_C_PHONE
#define c_credit payment->s_C_CREDIT
#define c_since payment->s_C_SINCE_time
#define c_data payment->s_C_DATA
#define w_street_1 payment->s_W_STREET_1
#define w_street_2 payment->s_W_STREET_2
#define w_city payment->s_W_CITY
#define w_state payment->s_W_STATE
#define w_zip payment->s_W_ZIP
#define d_street_1 payment->s_D_STREET_1
#define d_street_2 payment->s_D_STREET_2
#define d_city payment->s_D_CITY
#define d_state payment->s_D_STATE
#define d_zip payment->s_D_ZIP
/* Create Timestamp */
in_payment->s_H_DATE_time = (sqlint64) time( NULL ) ;
payment->deadlocks = -1 ;
payment->s_transtatus = TRAN_OK ;
if (c_w_id == 0) { c_w_id = w_id ; }
if (c_d_id == 0) { c_d_id = d_id ; }
#ifdef DEBUGIT
    pay_debug(payment, in_payment, "Client before SQL call");
#endif /* DEBUGIT */
    // Create c_data_prefix strings and copy some elements from
    // in -> out struct outside of retry_tran loop
    if ( in_c_id == 0 )
    {
        // Pre-built c_data prefix for BC does not include c_id in this instance
        // The strange $04.4d.%02.2d printf modifier is to print a (4,2)
        // 0-padded floating-point value -- %f won't 0-pad by default.
        c_data_prefix_c_last.len = sprintf( c_data_prefix_c_last.data, "%2.2d
%6.6d %2.2d %6.6d %04.4d.%02.2d", c_d_id, c_w_id, d_id, w_id ,
(int)(h_amount / 100), (int)(h_amount % 100) );
        // Setup the input c_last varchar
        c_last_input.len = strlen( in_payment->s_C_LAST ) ;
        memcpy( c_last_input.data , in_payment->s_C_LAST , c_last_input.len ) ;
        // Copy to the output structure
        memcpy( payment->s_C_LAST , in_payment->s_C_LAST , sizeof(
payment->s_C_LAST ) ) ;
    } else {
        // Copy c_id to the output structure
        c_id = in_c_id ;

        // Pre-built c_data prefix for BC does include c_id in this instance
        // The strange $04.4d.%02.2d printf modifier is to print a (4,2)
        // 0-padded floating-point value -- %f won't 0-pad by default.
        c_data_prefix_c_id.len = sprintf( c_data_prefix_c_id.data, "%5.5d %2.2d
%6.6d %2.2d %6.6d %04.4d.%02.2d", c_id, c_d_id, c_w_id, d_id, w_id ,
(int)(h_amount / 100), (int)(h_amount % 100) );
    }
    retry_tran:
    payment->deadlocks ++ ;
    if ( in_c_id == 0 )
    {
        EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC
        SELECT W_STREET_1, W_STREET_2, W_CITY, W_STATE,
W_ZIP
        , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
        , C_ID, 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, C_DATA

```

```

        INTO :w_street_1 , :w_street_2 , :w_city , :w_state , :w_zip
            , :d_street_1 , :d_street_2 , :d_city , :d_state , :d_zip
            , :c_id , :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 , :c_data :c_data_indicator

        FROM TABLE ( PAY_C_LAST( :w_id
            , :d_id
            , :c_w_id
            , :c_d_id
            , :c_last_input
            , :h_date
            , :h_amount
            , :c_data_prefix_c_last
        )
        ) AS T ( W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP
            , D_STREET_1, D_STREET_2, D_CITY, D_STATE,
D_ZIP
            , C_ID, 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, C_DATA
        )
        ;
        COMMIT ;
        END COMPOUND ;
    }
    else
    {
        EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC
        SELECT W_STREET_1, W_STREET_2, W_CITY, W_STATE,
W_ZIP
            , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
            , C_LAST, 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, C_DATA

        INTO :w_street_1 , :w_street_2 , :w_city , :w_state , :w_zip
            , :d_street_1 , :d_street_2 , :d_city , :d_state , :d_zip
            , :c_last , :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 , :c_data :c_data_indicator

        FROM TABLE ( PAY_C_ID( :w_id
            , :d_id
            , :c_w_id
            , :c_d_id
            , :in_c_id
            , :h_date
            , :h_amount
            , :c_data_prefix_c_id
        )
        ) AS T ( W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP
            , D_STREET_1, D_STREET_2, D_CITY, D_STATE,
D_ZIP
            , C_LAST, 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, C_DATA
        )
        ;
        COMMIT ;
        END COMPOUND ;
    }
    /* Update Output Structure with Timestamp */
    payment->s_H_DATE_time = in_payment->s_H_DATE_time ;
#ifdef DEBUGIT
    pay_debug(payment, in_payment, "Client after SQL call");
#endif /* DEBUGIT */
    if ( sqlca.sqlcode != 0 )
    {
        DLCHK( retry_tran ) ;
        sqlerror( PAYMENT_SQL , "PAY" , __FILE__ , __LINE__ , &sqlca ) ;
        payment->s_transtatus = FATAL_SQLERROR ;
        clientRc = FATAL_SQLERROR ;
        pay_debug( payment, in_payment, "PAY failed" ) ;
        EXEC SQL ROLLBACK WORK ;
        if ( sqlca.sqlcode != 0 )
        {
            sqlerror( PAYMENT_SQL , "ROLLBACK FAILED" , __FILE__ ,
__LINE__ , &sqlca ) ;
        }
    }
    return ( clientRc ) ;
}
// -----
// Order Status CLIENT
// -----

int ordstat_sql ( struct in_ordstat_struct * in_ordstat
                , struct out_ordstat_struct * ordstat)
{
    struct sqlca sqlca ;
    EXEC SQL BEGIN DECLARE SECTION;
    struct vc_ord_in
    {
        short len ;
        char data[ 42 ] ;
    } * in_ord ;
    struct vc_ord_out
    {
        short len ;
        char data[ 446 ] ;
    } * out_ord ;
    EXEC SQL END DECLARE SECTION;

    int clientRc = TRAN_OK ;
    int itemIndex = 0 ;

    in_ord = (struct vc_ord_in *) in_ordstat ;
    in_ord->len = sizeof(struct in_ordstat_struct) - SPGENERAL_ADJUST ;
    out_ord = (struct vc_ord_out *) ordstat ;
    out_ord->len = sizeof(struct out_ordstat_struct) - SPGENERAL_ADJUST ;
#ifdef DEBUGIT
    ord_debug(ordstat, in_ordstat, "Client before SP call");
#endif /* DEBUGIT */
#ifdef SWAP_ENDIAN
    SWAP_BYTE(in_ordstat->s_C_ID);
    SWAP_BYTE(in_ordstat->s_W_ID);
    SWAP_BYTE(in_ordstat->s_D_ID);
#endif //SWAP_ENDIAN
    EXEC SQL CALL ords ( :*in_ord , :*out_ord ) ;

#ifdef SWAP_ENDIAN
    SWAP_BYTE(in_ordstat->s_C_ID);

```

```

SWAP_BYTE(in_ordstat->s_W_ID);
SWAP_BYTE(in_ordstat->s_D_ID);
SWAP_BYTE(ordstat->s_C_BALANCE);
SWAP_BYTE(ordstat->s_O_ENTRY_D_time);
SWAP_BYTE(ordstat->s_C_ID);
SWAP_BYTE(ordstat->s_O_ID);
SWAP_BYTE(ordstat->s_O_CARRIER_ID);
SWAP_BYTE(ordstat->s_ol_cnt);
SWAP_BYTE(ordstat->s_transtatus);
SWAP_BYTE(ordstat->deadlocks);
for (itemIndex=0; itemIndex<ordstat->s_ol_cnt; itemIndex++)
{
    SWAP_BYTE(ordstat->item[ itemIndex ].s_OL_DELIVERY_D_time);
    SWAP_BYTE(ordstat->item[ itemIndex ].s_OL_AMOUNT);
    SWAP_BYTE(ordstat->item[ itemIndex ].s_OL_I_ID);
    SWAP_BYTE(ordstat->item[ itemIndex ].s_OL_SUPPLY_W_ID);
    SWAP_BYTE(ordstat->item[ itemIndex ].s_OL_QUANTITY);
}
#endif //SWAP_ENDIAN
if ( sqlca.sqlcode == 0 )
{
    // Propagate the field we already knew into the output structure
    // 60% of the time, we already new c_last (input c_id is 0)
    if ( in_ordstat->s_C_ID == 0 )
    {
        memcpy( ordstat->s_C_LAST , in_ordstat->s_C_LAST, sizeof(
ordstat->s_C_LAST ) );
    }
    else
    {
        ordstat->s_C_ID = in_ordstat->s_C_ID ;
    }
}
else
{
    sqlerror( ORDSTAT_SQL, "ORD", __FILE__, __LINE__, &sqlca );
    ordstat->s_transtatus = FATAL_SQLERROR ;
    clientRc = FATAL_SQLERROR ;
}
#endif DEBUGIT
ord_debug(ordstat, in_ordstat, "Client after SP call");
#endif /* DEBUGIT */
if ( ordstat->s_transtatus <= FATAL_SQLERROR )
{
    ord_debug(ordstat, in_ordstat, "ORD failed");
    clientRc = FATAL_SQLERROR ;
}
return ( clientRc ) ;
}
// -----
// Delivery CLIENT
// -----
int delivery_sql( struct in_delivery_struct * in_delivery
, struct out_delivery_struct * delivery )
{
    struct sqlca sqlca ;
    EXEC SQL BEGIN DECLARE SECTION;
    struct vc_del_in
    {
        short len ;
        char data[ 22 ] ;
    } * in_del ;
    struct vc_del_out
    {
        short len;
        char data[ 50 ] ;
    } * out_del ;
    EXEC SQL END DECLARE SECTION;

```

```

int clientRc = TRAN_OK ;
int orderIndex = 0 ;
/* Create Timestamp */
in_delivery->s_O_DELIVERY_D_time = (sqlint64) time( NULL ) ;
in_del = (struct vc_del_in *) in_delivery ;
in_del->len = sizeof(struct in_delivery_struct) - SPGENERAL_ADJUST;
out_del = (struct vc_del_out *) delivery ;
out_del->len = sizeof(struct out_delivery_struct) - SPGENERAL_ADJUST;
#endif DEBUGIT
del_debug(delivery, in_delivery, "Client before SP call");
#endif /* DEBUGIT */
#endif SWAP_ENDIAN
SWAP_BYTE(in_delivery->s_O_DELIVERY_D_time);
SWAP_BYTE(in_delivery->s_W_ID);
SWAP_BYTE(in_delivery->s_O_CARRIER_ID);
#endif //SWAP_ENDIAN
EXEC SQL CALL dels ( : *in_del, : *out_del ) ;
#endif SWAP_ENDIAN
SWAP_BYTE(in_delivery->s_O_DELIVERY_D_time);
SWAP_BYTE(in_delivery->s_W_ID);
SWAP_BYTE(in_delivery->s_O_CARRIER_ID);
for (orderIndex=0; orderIndex<10; orderIndex++) {
    SWAP_BYTE(delivery->s_O_ID[ orderIndex ]);
}
SWAP_BYTE(delivery->s_transtatus);
SWAP_BYTE(delivery->deadlocks);
#endif //SWAP_ENDIAN
#endif DEBUGIT
del_debug(delivery, in_delivery, "Client after SP call");
#endif /* DEBUGIT */
if ( sqlca.sqlcode != 0 )
{
    sqlerror( DELIVERY_SQL, "DEL", __FILE__, __LINE__, &sqlca );
    delivery->s_transtatus = FATAL_SQLERROR ;
    clientRc = FATAL_SQLERROR ;
}
if ( delivery->s_transtatus <= FATAL_SQLERROR )
{
    del_debug(delivery, in_delivery, "DEL failed");
    clientRc = FATAL_SQLERROR ;
}
return ( clientRc ) ;
}
// -----
// Stock CLIENT
// -----
#undef w_id
#undef d_id
int stocklev_sql ( struct in_stocklev_struct * in_stocklev
, struct out_stocklev_struct * stocklev )
{
    struct sqlca sqlca ;
    int clientRc = TRAN_OK ;
    EXEC SQL BEGIN DECLARE SECTION;

    // input
    ##sqlint32 w_id ;
    ##short d_id ;
    sqlint32 threshold ;
    // output

    sqlint32 low_stock ;
    EXEC SQL END DECLARE SECTION;
#define w_id in_stocklev->s_W_ID
#define d_id in_stocklev->s_D_ID
#define threshold in_stocklev->s_threshold
#define low_stock stocklev->s_low_stock

```



```

stocklev->deadlocks = -1 ;
stocklev->s_transtatus = TRAN_OK ;
#ifdef DEBUGIT
stk_debug(stocklev, in_stocklev, "Client before SQL call");
#endif /* DEBUGIT */
retry_tran:
stocklev->deadlocks ++ ;
/*
EXEC SQL
SELECT ITEMS_BELOW_THRESHOLD
INTO :low_stock
FROM TABLE( STOCK_LEVEL( :w_id, :d_id, :threshold ) ) AS T
WITH CS;
*/

EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC
SELECT COUNT( S_I_ID ) INTO :low_stock
FROM ( SELECT DISTINCT S_I_ID
FROM ORDER_LINE , STOCK , DISTRICT
WHERE D_W_ID = :w_id
AND D_ID = :d_id
AND OL_O_ID < d_next_o_id
AND OL_O_ID >= ( d_next_o_id - 20 )
AND OL_W_ID = D_W_ID
AND OL_D_ID = D_ID
AND S_I_ID = OL_I_ID
AND S_W_ID = OL_W_ID
AND S_QUANTITY < :threshold

) OLS

WITH CS
;

COMMIT ;

END COMPOUND ;
#ifdef DEBUGIT
stk_debug(stocklev, in_stocklev, "Client after SQL call");
#endif /* DEBUGIT */
if ( sqlca.sqlcode != 0 )
{
DLCHK( retry_tran ) ;
sqlerror( STOCKLEV_SQL , "STK" , __FILE__ , __LINE__ , &sqlca);
stocklev->s_transtatus = FATAL_SQLERROR ;
clientRc = FATAL_SQLERROR ;
stk_debug( stocklev, in_stocklev, "STK failed" ) ;
EXEC SQL ROLLBACK WORK ;
if ( sqlca.sqlcode != 0 )
{
sqlerror( STOCKLEV_SQL , "ROLLBACK FAILED" , __FILE__ ,
__LINE__ , &sqlca ) ;
}
}
return ( clientRc ) ;
}

```

NULLDB/NULLDB.h

```

// The following ifdef block is the standard way of creating macros which make
exporting
// from a DLL simpler. All files within this DLL are compiled with the
NULLDB_EXPORTS
// symbol defined on the command line. this symbol should not be defined on
any project
// that uses this DLL. This way any other project whose source files include this
file see
// NULLDB_API functions as being imported from a DLL, whereas this DLL
sees symbols

```

```

// defined with this macro as being exported.
#ifdef NULLDB_EXPORTS
#define NULLDB_API __declspec(dllexport)
#else
#define NULLDB_API __declspec(dllimport)
#endif
extern NULLDB_API int dataSet;
extern "C" NULLDB_API int do_nord(struct nord_wrapper *nord, void *ctx);
extern "C" NULLDB_API int do_pymt(struct pymt_wrapper *pymt, void *ctx);
extern "C" NULLDB_API int do_ordr(struct ordr_wrapper *ordr, void *ctx);
extern "C" NULLDB_API int do_dlvv(struct dlvv_wrapper *dlvv, void *ctx);
extern "C" NULLDB_API int do_stok(struct stok_wrapper *stok, void *ctx);
extern "C" NULLDB_API int connect_db(char *dbName, void **ctx);
extern "C" NULLDB_API int disconnect_db(void *ctx);

```

NULLDB/NULLDB.cpp

```

// NULLDB.cpp : Defines the entry point for the DLL application.
//
#include "stdafx.h"
#include "NULLDB.h"
#include "..\tpccIsapi\tpcc.h"
BOOL WINAPIENTRY DllMain( HANDLE hModule,
DWORD ul_reason_for_call,
LPVOID lpReserved
)
{
switch (ul_reason_for_call)
{
case DLL_PROCESS_ATTACH:
case DLL_THREAD_ATTACH:
case DLL_THREAD_DETACH:
case DLL_PROCESS_DETACH:
break;
}
return TRUE;
}
// This is an example of an exported variable
NULLDB_API int dataSet = 0;
extern "C" NULLDB_API int connect_db(char *dbName, void **ctx)
{
return OK;
}
extern "C" NULLDB_API int disconnect_db(void *ctx)
{
return OK;
}
extern "C" NULLDB_API int do_nord(struct nord_wrapper *nord, void *ctx)
{
nord->out_nord.s_transtatus = 0;
if (dataSet == 0)
{
strcpy(nord->out_nord.s_C_LAST, "NOYOLA");
strcpy(nord->out_nord.s_C_CREDIT, "GC");
nord->out_nord.s_W_TAX = 1694;
nord->out_nord.s_D_TAX = 967;
nord->out_nord.s_C_DISCOUNT = 1024;
nord->out_nord.s_O_ID = 3013;
nord->out_nord.s_O_OL_CNT = 4;
nord->out_nord.s_total_amount = 32345;
nord->out_nord.s_O_ENTRY_D_time = 1234567890;
strcpy(nord->out_nord.item[0].s_I_NAME, "98 Toyota Supra Turbo");
nord->in_nord.in_item[0].s_OL_I_ID = 1;
nord->in_nord.in_item[0].s_OL_QUANTITY = 1;
nord->in_nord.in_item[0].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[0].s_I_PRICE = 42000;
nord->out_nord.item[0].s_OL_AMOUNT = 554000;
nord->out_nord.item[0].s_S_QUANTITY = 31;
nord->out_nord.item[0].s_brand_generic = 'G';
}
}

```

```

strepy(nord->out_nord.item[1].s_I_NAME,"HKS Turbo Timer");
nord->in_nord.in_item[1].s_OL_I_ID = 1;
nord->in_nord.in_item[1].s_OL_QUANTITY = 1;
nord->in_nord.in_item[1].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[1].s_I_PRICE = 4500;
nord->out_nord.item[1].s_OL_AMOUNT = 438100;
nord->out_nord.item[1].s_S_QUANTITY = 57;
nord->out_nord.item[1].s_brand_generic = 'G';
strepy(nord->out_nord.item[2].s_I_NAME,"TRD GEN2 Exhaust");
nord->in_nord.in_item[2].s_OL_I_ID = 1;
nord->in_nord.in_item[2].s_OL_QUANTITY = 1;
nord->in_nord.in_item[2].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[2].s_I_PRICE = 6734;
nord->out_nord.item[2].s_OL_AMOUNT = 47173;
nord->out_nord.item[2].s_S_QUANTITY = 42;
nord->out_nord.item[2].s_brand_generic = 'G';
strepy(nord->out_nord.item[3].s_I_NAME,"BLITZ DUAL-SOLENOID");
nord->in_nord.in_item[3].s_OL_I_ID = 1;
nord->in_nord.in_item[3].s_OL_QUANTITY = 1;
nord->in_nord.in_item[3].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[3].s_I_PRICE = 35000;
nord->out_nord.item[3].s_OL_AMOUNT = 12096;
nord->out_nord.item[3].s_S_QUANTITY = 84;
nord->out_nord.item[3].s_brand_generic = 'G';
dataSet = 1;
}
else
{
strepy(nord->out_nord.s_C_LAST,"SIMPSON");
strepy(nord->out_nord.s_C_CREDIT,"GC");
nord->out_nord.s_W_TAX = 913;
nord->out_nord.s_D_TAX = 1519;
nord->out_nord.s_C_DISCOUNT = 958;
nord->out_nord.s_O_ID = 1410;
nord->out_nord.s_O_OL_CNT = 9;
nord->out_nord.s_total_amount = 12345;
nord->out_nord.s_O_ENTRY_D_time = 1234567890;
strepy(nord->out_nord.item[0].s_I_NAME,"97 Toyota Supra NA");
nord->in_nord.in_item[0].s_OL_I_ID = 1;
nord->in_nord.in_item[0].s_OL_QUANTITY = 1;
nord->in_nord.in_item[0].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[0].s_I_PRICE = 30000;
nord->out_nord.item[0].s_OL_AMOUNT = 769600;
nord->out_nord.item[0].s_S_QUANTITY = 97;
nord->out_nord.item[0].s_brand_generic = 'G';
strepy(nord->out_nord.item[1].s_I_NAME,"98 Turbo Stereo");
nord->in_nord.in_item[1].s_OL_I_ID = 1;
nord->in_nord.in_item[1].s_OL_QUANTITY = 1;
nord->in_nord.in_item[1].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[1].s_I_PRICE = 10001;
nord->out_nord.item[1].s_OL_AMOUNT = 192999;
nord->out_nord.item[1].s_S_QUANTITY = 51;
nord->out_nord.item[1].s_brand_generic = 'G';
strepy(nord->out_nord.item[2].s_I_NAME,"XERD Exhaust Header");
nord->in_nord.in_item[2].s_OL_I_ID = 1;
nord->in_nord.in_item[2].s_OL_QUANTITY = 1;
nord->in_nord.in_item[2].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[2].s_I_PRICE = 4000;
nord->out_nord.item[2].s_OL_AMOUNT = 41670;
nord->out_nord.item[2].s_S_QUANTITY = 14;
nord->out_nord.item[2].s_brand_generic = 'G';
strepy(nord->out_nord.item[3].s_I_NAME,"LEXOL Conditioner");
nord->in_nord.in_item[3].s_OL_I_ID = 1;
nord->in_nord.in_item[3].s_OL_QUANTITY = 1;
nord->in_nord.in_item[3].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[3].s_I_PRICE = 1400;
nord->out_nord.item[3].s_OL_AMOUNT = 17213;
nord->out_nord.item[3].s_S_QUANTITY = 90;

nord->out_nord.item[3].s_brand_generic = 'G';
strepy(nord->out_nord.item[4].s_I_NAME,"TRD Sticker 1");
nord->in_nord.in_item[4].s_OL_I_ID = 1;
nord->in_nord.in_item[4].s_OL_QUANTITY = 1;
nord->in_nord.in_item[4].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[4].s_I_PRICE = 1400;
nord->out_nord.item[4].s_OL_AMOUNT = 27232;
nord->out_nord.item[4].s_S_QUANTITY = 75;
nord->out_nord.item[4].s_brand_generic = 'G';
strepy(nord->out_nord.item[5].s_I_NAME,"TRD Sticker 2");
nord->in_nord.in_item[5].s_OL_I_ID = 1;
nord->in_nord.in_item[5].s_OL_QUANTITY = 1;
nord->in_nord.in_item[5].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[5].s_I_PRICE = 4400;
nord->out_nord.item[5].s_OL_AMOUNT = 35808;
nord->out_nord.item[5].s_S_QUANTITY = 22;
nord->out_nord.item[5].s_brand_generic = 'G';
strepy(nord->out_nord.item[6].s_I_NAME,"TRD Sticker 3");
nord->in_nord.in_item[6].s_OL_I_ID = 1;
nord->in_nord.in_item[6].s_OL_QUANTITY = 1;
nord->in_nord.in_item[6].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[6].s_I_PRICE = 5500;
nord->out_nord.item[6].s_OL_AMOUNT = 44392;
nord->out_nord.item[6].s_S_QUANTITY = 21;
nord->out_nord.item[6].s_brand_generic = 'G';
strepy(nord->out_nord.item[7].s_I_NAME,"TRD Sticker 4");
nord->in_nord.in_item[7].s_OL_I_ID = 1;
nord->in_nord.in_item[7].s_OL_QUANTITY = 1;
nord->in_nord.in_item[7].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[7].s_I_PRICE = 8300;
nord->out_nord.item[7].s_OL_AMOUNT = 83410;
nord->out_nord.item[7].s_S_QUANTITY = 35;
nord->out_nord.item[7].s_brand_generic = 'G';
strepy(nord->out_nord.item[8].s_I_NAME,"98 Toyota OEM Bra");
nord->in_nord.in_item[8].s_OL_I_ID = 1;
nord->in_nord.in_item[8].s_OL_QUANTITY = 1;
nord->in_nord.in_item[8].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[8].s_I_PRICE = 10000;
nord->out_nord.item[8].s_OL_AMOUNT = 43160;
nord->out_nord.item[8].s_S_QUANTITY = 73;
nord->out_nord.item[8].s_brand_generic = 'G';
dataSet = 0;
}
}
return OK;
}
extern "C" NULLDB_API int do_pymt(struct paym_wrapper *pymt,void *ctx)
{
pymt->out_paym.s_transtatus = 0;
if (dataSet == 0)
{
pymt->out_paym.s_C_CREDIT_LIM = 5000000;

pymt->out_paym.s_C_DISCOUNT = 1024;
pymt->out_paym.s_C_BALANCE = 17815;
pymt->out_paym.s_C_ID = 89;
pymt->out_paym.s_H_DATE_time = 1234567890;
strepy(pymt->out_paym.s_W_STREET_1,"11501 Burnet Rd");
strepy(pymt->out_paym.s_W_STREET_2,"BLD 905");
strepy(pymt->out_paym.s_W_CITY,"Austin");
strepy(pymt->out_paym.s_W_STATE,"TX");
strepy(pymt->out_paym.s_W_ZIP,"78758");
strepy(pymt->out_paym.s_D_STREET_1,"11900 Hobby Horse");
strepy(pymt->out_paym.s_D_STREET_2,"Apt. 525");
strepy(pymt->out_paym.s_D_CITY,"Valley");
strepy(pymt->out_paym.s_D_STATE,"TX");
strepy(pymt->out_paym.s_D_ZIP,"78559");
strepy(pymt->out_paym.s_C_FIRST,"Jim");
strepy(pymt->out_paym.s_C_MIDDLE,"F");
}
}

```



```

}
extern "C" NULLDB_API int do_dlvly(struct dlvly_wrapper *dlvy,void *ctx)
{
    dlvly->out_dlvly.s_transtatus = 0;
    if (dataSet == 0)
    {
        dataSet = 1;
        for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
            dlvly->out_dlvly.s_O_ID[districtIndex]= 2055;
    }
    else
    {
        for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
            dlvly->out_dlvly.s_O_ID[districtIndex]= 2056;
        dataSet = 0;
    }
    return OK;
}

```

```

extern "C" NULLDB_API int do_stok(struct stok_wrapper *stok,void *ctx)
{
    stok->out_stok.s_transtatus = 0;
    if (dataSet == 0)
    {
        stok->out_stok.s_low_stock = 100;
        dataSet = 1;
    }
    else
    {
        stok->out_stok.s_low_stock = 40;
        dataSet = 0;
    }
    return OK;
}

```

nullDB/stdafx.h

```

// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//
#pragma once

```

```

#define WIN32_LEAN_AND_MEAN // Exclude rarely-used
stuff from Windows headers
// Windows Header Files:
#include <windows.h>
// TODO: reference additional headers your program requires here

```

nullDB/stdafx.cpp

```

// stdafx.cpp : source file that includes just the standard includes
// nullDB.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information
#include "stdafx.h"
// TODO: reference any additional headers you need in STDAFX.H
// and not in this file

```

tpccIsapi/htmlPhraser.h

```

////////////////////////////////////
// htmlPharaser.h
////////////////////////////////////
// Class to decode a html query string
////////////////////////////////////

```

```

#pragma once
#include <memory.h>
////////////////////////////////////
// Definitions
////////////////////////////////////
#define NULL 0
#define COMMAND_ID 0
#define TERM_ID 1
#define W_ID 2
#define D_ID 3
#define C_ID 4
#define C_NAME 5
#define C_W_ID 6
#define C_D_ID 7
#define AMT_PAID 8
#define STK_THRESHOLD 9
#define CARRIER_NUM 10
#define ITEM_LIST_START 11
#define ITEM_LIST_FINISH 11
#define MAX_QUERY_ID 55
#define MAX_FIELD_LEN 256
#define MAX_FIELD_NUM 56
////////////////////////////////////
// Command Codes
////////////////////////////////////
#define NEW_ORDER_CODE 'n'
#define PAYMENT_CODE 'p'
#define ORDER_STATUS_CODE 'o'
#define DELIVERY_CODE 'd'
#define STOCK_CODE 's'
#define EXIT_CODE 'e'
#define MENU_CODE 'm'
#define COMMAND_LOGIN 0
#define COMMAND_NEW_ORDER 1
#define COMMAND_PAYMENT 2
#define COMMAND_ORDER_STATUS 3
#define COMMAND_DELIVERY 4
#define COMMAND_STOCK 5
#define COMMAND_EXIT 6
#define COMMAND_LOGIN_RESULTS 7
#define COMMAND_NEW_ORDER_RESULTS 8
#define COMMAND_PAYMENT_RESULTS 9
#define COMMAND_ORDER_STATUS_RESULTS 10
#define COMMAND_DELIVERY_RESULTS 11
#define COMMAND_STOCK_RESULTS 12
////////////////////////////////////
// Class htmlPhraser
////////////////////////////////////
class htmlPhraser
{
    // Constructors / Destructor
public:
    htmlPhraser(char *queryString);

```

```

~htmlPhraser()
{return;}

// getters
public:
    int      getCommandId();
    int validate(int txnType);

    char *   get_TERM_ID()
{return iQueryValues[TERM_ID];}
    char *   get_W_ID()
{return iQueryValues[W_ID];}
    char *   get_D_ID()
{return iQueryValues[D_ID];}
    char *   get_C_ID()
{return iQueryValues[C_ID];}
    char *   get_C_NAME()
{return iQueryValues[C_NAME];}
    char *   get_C_W_ID()
{return iQueryValues[C_W_ID];}
    char *   get_C_D_ID()
{return iQueryValues[C_D_ID];}
    char *   get_AMT_PAID()
{return iQueryValues[AMT_PAID];}
    char *   get_STK_THRESHOLD()
{return iQueryValues[STK_THRESHOLD];}
    char *   get_CARRIER_NUM()
{return iQueryValues[CARRIER_NUM];}
    char *   get_ITEM_SUPP_W(int item) {return
iQueryValues[(ITEM_LIST_START + 0) + (item * 3)];}
    char *   get_ITEM_ITEM_NUM(int item)
{return iQueryValues[(ITEM_LIST_START + 1) + (item * 3)];}
    char *   get_ITEM_QTY(int item)
{return iQueryValues[(ITEM_LIST_START + 2) + (item * 3)];}

// Class Functions
private:
    char convertQueryToken(char **queryString);
// Class Attributes
private:
    int      iCustomerIdFlag;
    int      iCarrierNumFlag;
    int      iStockThresholdFlag;
    char
iQueryValues[MAX_FIELD_NUM][MAX_FIELD_LEN];
};
///////////////////////////////////////////////////////////////////

```

tpccsapi/resource.h

```

//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by tpccsapi.rc
//
#define IDS_PROJNAME 100

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 201
#define _APS_NEXT_COMMAND_VALUE 32768
#define _APS_NEXT_CONTROL_VALUE 201
#define _APS_NEXT_SYMED_VALUE 101
#endif
#endif

```

tpccsapi/StdAfx.h

```

// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//
#pragma once
#define WIN32_LEAN_AND_MEAN // Exclude rarely-used
stuff from Windows headers
#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS // some
CString constructors will be explicit
#define _ATL_ALL_WARNINGS
// critical error descriptions will only be shown to the user
// in debug builds. they will always be logged to the event log
#ifndef _DEBUG
#define ATL_CRITICAL_ISAPI_ERROR_LOGONLY
#endif

#ifndef _WIN32_WINNT
#define _WIN32_WINNT 0x0403
#endif
// TODO: this disables support for registering COM objects
// exported by this project since the project contains no
// COM objects or typelib. If you wish to export COM objects
// from this project, add a typelib and remove this line
#define _ATL_NO_COM_SUPPORT
#include "resource.h"
#include <atlsrvres.h>
#include <atlisapi.h>
#include <atlstencil.h>
// TODO: reference additional headers your program requires here

```

tpccsapi/tpcc.h

```

// Common defines and structures use internally by client code
// Not to be confused with structures actually passed in transactions
//
// standard includes
#ifndef _COMMON_TPCC
#define _COMMON_TPCC
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/timeb.h>
#include <time.h>
#include <db2tpcc.h>
#include <iostream>
#include <fstream>
#include <process.h>
#include <ios>
///////////////////////////////////////////////////////////////////
// Defines
///////////////////////////////////////////////////////////////////
#define OK
0
#define INVALID_STATUS -1
#define ERR -1
#define INVALID_COM_STATUS -2

#define TXN_MAX_COMMANDS 55
#define MAX_TRANSACTIONS 14
#define MAX_CMD_LENGTH 100
#define INPUT_ITEMS 3
#define MAX_INT_BUFFER 15
#define NORD_ITEMS 15

```

```

#define ITEM_START
11
#define ITEM_END
55
#define MAX_ITEMS
15
#define MAX_STRING_LEN 256
#define MAX_HTML_PAGE_LEN 4096
#define MAX_HTML_HEADER_LEN 512
#define DELIVERY_THREADS_NUM 100
#define DISTRICTS_PER_WAREHOUSE 10
// Transaction Codes
// Transaction Codes
#define TXN_LOGIN
0
#define TXN_NEW_ORDER 1
#define TXN_PAYMENT
2
#define TXN_ORDER_STATUS 3
#define TXN_DELIVERY 4
#define TXN_STOCK
5
#define TXN_EXIT
6
#define TXN_LOGIN_RESULTS 7
#define TXN_NEW_ORDER_RESULTS 8
#define TXN_PAYMENT_RESULTS 9
#define TXN_ORDER_STATUS_RESULTS 10
#define TXN_DELIVERY_RESULTS 11
#define TXN_STOCK_RESULTS
12
#define CMD_NORD
"nord"
#define CMD_PYMT
"pymt"
#define CMD_ORDS
"ords"
#define CMD_DLVY
"dlvy"
#define CMD_STOK
"stok"
#define CMD_EXIT
"exit"
#define CMD_MENU
"menu"
#define APP_NAME
"tpcc.html"
#define HEADER
"Content-Type:text/html\r\nContent-Length: %d\r\nConnection:
Keep-Alive\r\n\r\n"
// URL Commands
// URL Commands
#define CMD_TXN_ID
"00"
#define CMD_TERM_ID
"01"
#define CMD_W_ID
"02"
#define CMD_D_ID
"03"
#define CMD_C_ID
"04"
#define CMD_C_NAME
"05"
#define CMD_C_W_ID
"06"

#define CMD_C_D_ID
"07"
#define CMD_AMT_PAID
"08"
#define CMD_STK_THRESHOLD
"09"
#define CMD_CARRIER_NUM
"10"
#define ITEM01_SUPP_W
"11"
#define ITEM01_ITEM_NUM
"12"
#define ITEM01_OTY
"13"

#define CHAR_FILL ''
#define NUMERIC_FILL ''
#define NEGITIVE_SYMBOL '-'
#define MONEY_SYMBOL '$'
#define DECIMAL_SYMBOL '.'
#define ZERO_SYMBOL '0'
#define ZIP_DELIMITER ':'
#define PHONE_DELIMITER '-'
#define DATE_DELIMITER '-'
#define TIME_DELIMITER ':'
#define DEFAULT_MONEY64_LEN 15
#define DEFAULT_MONEY32_LEN 9
#define DEFAULT_MONEY16_LEN 9
#define DEFAULT_NUMERIC64_LEN 15
#define DEFAULT_NUMERIC32_LEN 9
#define DEFAULT_NUMERIC16_LEN 9
#define DEFAULT_DECIMAL64_LEN 5
#define DEFAULT_DECIMAL32_LEN 5
#define DEFAULT_DECIMAL16_LEN 5
#define DEFAULT_DATETIME_LEN 19
#define DEFAULT_DATE_LEN 11
#define DEFAULT_TIME_LEN 8

#define DEFAULT_STRING_LEN 25
#define DEFAULT_ZIP_LEN 17
#define DEFAULT_PHONE_LEN 18
// String Field Lengths
// String Field Lengths
#define NAME_LEN 24
#define LAST_NAME_LEN 16
#define FIRST_NAME_LEN 16
#define INITIALS_LEN 2
#define CREDIT_LEN 2
#define STREET_LEN 20
#define CITY_LEN 20
#define STATE_LEN 2
#define ZIP_LEN 9
#define PHONE_LEN 16
#define DATA_LEN 200
#define ITEM_LIST 15
#define ORDER_LIST 10
// Type definitions
// Type definitions
typedef __int8 INT8b;
typedef __int16 INT16b;
typedef __int32 INT32b;
typedef __int64 INT64b;
typedef unsigned __int8 UINT8b;
typedef unsigned __int16 UINT16b;
typedef unsigned __int32 UINT32b;
typedef unsigned __int64 UINT64b;

```

```

typedef INT16b          sqlint16;
typedef INT32b          sqlint32;
typedef INT64b          sqlint64;
typedef INT16b          int16_t;
typedef INT32b          int32_t;
typedef INT64b          int64_t;
typedef char            BYTE8b;
typedef double          DOUBLE;
typedef unsigned long   NATURAL;
// Date and time values
// Error codes
#define ERR_INVALID_TXN_TYPE          -1
#define ERR_MISSING_W_ID              -2
#define ERR_NON_NUMERIC_W_ID         -3
#define ERR_MISSING_D_ID              -4
#define ERR_NON_NUMERIC_D_ID         -5
#define ERR_MISSING_C_ID              -6
#define ERR_NON_NUMERIC_C_ID         -7
#define ERR_MISSING_SUPP_W           -8
#define ERR_NON_NUMERIC_SUPP_W       -9
#define ERR_MISSING_ITEM_NUM         -10
#define ERR_NON_NUMERIC_ITEM_NUM     -11
#define ERR_MISSING_ITEM_OTY         -12
#define ERR_NON_NUMERIC_ITEM_QTY     -13
#define ERR_MISSING_CLAST_NAME       -14
#define ERR_NON_NUMERIC_CUST_W_ID    -15
#define ERR_NON_NUMERIC_CUST_D_ID    -16
#define ERR_MISSING_AMOUNT_PAID      -17
#define ERR_NON_NUMERIC_AMOUNT_PAID -18
#define ERR_INVALID_D_ID              "ERROR: Invalid District ID. Try Again."
#define ERR_INVALID_W_ID              "ERROR: Invalid Warehouse ID. Try Again."
#define ERR_INVALID_C_ID              "ERROR: Invalid Customer ID. Try Again."
#define ERR_INVALID_SUPPLY_W_ID       "ERROR: Invalid Item Supply Warehouse. Try Again."
#define ERR_INVALID_ITEM_NUM          "ERROR: Invalid Item Number. Try Again."
#define ERR_INVALID_ITEM_OTY         "ERROR: Invalid Item Qty. Try Again."
#define ERR_MISSING_C_ID_OR_CLAST     "ERROR: Must Enter Customer Id or Customer Last Name. Try Again."
#define ERR_INVALID_PAYMENT_AMOUNT    "ERROR: Invalid Payment Amount. Try Again."
#define ERR_INVALID_CARRIER          "ERROR: Invalid Carrier Number. Try Again."
#define ERR_INVALID_THRESHOLD         "ERROR: Invalid Threshold. Try Again."
#define ERR_INVALID_C_D_ID            "ERROR: Invalid Customer District Id. Try Again."
#define ERR_INVALID_C_W_ID            "ERROR: Invalid Customer Warehouse Id. Try Again."

#define ERR_TERMINAL_FULL              "ERROR: Terminal can not support user. Terminal full."
#define ERR_C_ID_OR_CLAST_ONLY        "ERROR: Either customer id or customer last name can be specified."
#define ERR_UNABLE_TO_OPEN_REG        -50
#define ERR_DLVY_THREAD_FAILED        -51
#define ERR_DLVY_SEMAPHORE_INIT_FAILED -52
#define ERR_DLVY_EVENT_INIT_FAILED    -53
#define ERR_DLVY_QUEUE_EATING_TAIL    -54
#define ERR_INVALID_USERNAME          -70
#define ERR_INVALID_PASSWORD          -71
#define ERR_INVALID_DB_NAME           -72
#define ERR_INVALID_REGISTRY_KEY      -73
#define ERR_DB2_DLL_NOT_LOADED        -74
#define ERR_ORACLE_DLL_NOT_LOADED    -75
#define ERR_CONNECT_ADDRESS_NOT_FOUND -76
#define ERR_NORD_ADDRESS_NOT_FOUND    -77
#define ERR_PYMT_ADDRESS_NOT_FOUND    -78
#define ERR_ORDS_ADDRESS_NOT_FOUND    -79
#define ERR_DLVY_ADDRESS_NOT_FOUND    -80
#define ERR_STOK_ADDRESS_NOT_FOUND    -81
#define ERR_NULL_DLL_NOT_LOADED       -82
#define ERR_UNKNOWN_DB                -83
#define ERR_DISCONNECT_ADDRESS_NOT_FOUND -84
#define ERR_SAVING_CONTEXT            -90
#define ERR_DETACHING_CONTEXT         -91
#define ERR_ATTACHING_CONTEXT         -92
#define ERR_HANDLE_IN_USE             -93
#define ERR_CONNECT_TO_TM_FAILED      -99
#define ERR_DLVY_LOG_OPEN_FAILED      -100
#define ERR_DLVY_QUEUE_FULL           -101

// Registry Definitions
#define REGISTRY_SUB_KEY              "SOFTWARE\TPCC"
#define DELIVERY_THREADS              "dlvyThreads"
#define DELIVERY_QUEUE_LEN            "dlvyQueueLen"
#define DELIVERY_LOG_PATH             "dlvyLogPath"
#define ERROR_LOG_FILE                "errorLogFile"
#define HTML_TRACE_LOG_FILE           "htmlTraceLogFile"
#define DB_NAME                       "dbName"
#define NULL_DB                       "nullDB"
#define COM_NULL_DB                   "comnullDB"
#define CLIENT_NULL_DB                "clientNullDB"
#define NUM_USERS                     "numUsers"
#define DB_TYPE                       "dbType"
#define TXN_MONITOR                   "txn_server"

```

```

#define      COMM_POOL
"comm_pool"
#define      HTML_TRACE
"htmlTrace"
#define      ISAPI_TRACE
"isapi_trace"
#define      DEFAULT_DLVY_THREADS      1
#define      DEFAULT_DLVY_QUEUE_LEN
10
#define      DEFAULT_DLVY_LOG_PATH
"c:\\inetpub\\wwwroot\\tpcc\\dlvy"
#define      DEFAULT_ERROR_LOG_FILE
"c:\\inetpub\\wwwroot\\tpcc\\errorLog.txt"
#define      DEFAULT_HTML_TRACE_LOG_FILE
"c:\\inetpub\\wwwroot\\tpcc\\htmlTrace.txt"
#define      DEFAULT_NUM_USERS
10000
#define      DEFAULT_DB_NAME
"tpcc"
////////////////////////////////////////////////////////////////////
// Structure defines
////////////////////////////////////////////////////////////////////
struct nord_wrapper {
    struct in_neword_struct in_nord;
    struct out_neword_struct out_nord;
};
struct paym_wrapper {
    struct in_payment_struct in_paym;
    struct out_payment_struct out_paym;
};
struct ords_wrapper {
    struct in_ordstat_struct in_ords;
    struct out_ordstat_struct out_ords;
};
struct dlvy_wrapper {
    struct in_delivery_struct in_dlvy;
    struct out_delivery_struct out_dlvy;
};
struct stok_wrapper {
    struct in_stocklev_struct in_stok;
    struct out_stocklev_struct out_stok;
};
typedef struct
{
    int      year;
    int      month;
    int      day;
    int      hour;
    int      minute;
    int      second;
} datetime;
struct NEWORDERDATA
{
    struct in_items_struct {
        int s_OL_I_ID;
        int s_OL_SUPPLY_W_ID;
        short s_OL_QUANTITY;
    } in_item[15];
    long long in_s_O_ENTRY_D_time;      /* init by SUT */
    int      in_s_C_ID;
    int      in_s_W_ID;
    short    in_s_D_ID;
    short    in_s_O_OL_CNT;      /* init by SUT */
    short    in_s_all_local;
    short    in_duplicate_items;
    struct out_items_struct {
        double s_I_PRICE;
        double s_OL_AMOUNT;

```

```

        short    s_S_QUANTITY;
        char    s_I_NAME[25];
        char    s_brand_generic;
    } out_item[15];
    long long out_s_O_ENTRY_D_time;
    double out_s_W_TAX;
    double out_s_D_TAX;
    double out_s_C_DISCOUNT;
    double out_s_total_amount;
    int      out_s_O_ID;
    short    out_s_O_OL_CNT;
    short    out_s_transtatus;
    short    out_deadlocks;
    char    out_s_C_LAST[17];
    char    out_s_C_CREDIT[3];
};
struct PAYMENTDATA
{
    long long in_s_H_DATE_time;
    double in_s_H_AMOUNT;
    int      in_s_W_ID;
    int      in_s_C_W_ID;
    int      in_s_C_ID;
    short    in_s_C_D_ID;
    short    in_s_D_ID;
    char    in_s_C_LAST[17];
    long long out_s_H_DATE_time;
    long long out_s_C_SINCE_time;
    double out_s_C_CREDIT_LIM;
    double out_s_C_BALANCE;
    double out_s_C_DISCOUNT;
    int      out_s_C_ID;
    short    out_s_transtatus;
    short    out_deadlocks;
    char out_s_W_STREET_1[21];
    char out_s_W_STREET_2[21];
    char out_s_W_CITY[21];
    char out_s_W_STATE[3];
    char out_s_W_ZIP[10];
    char out_s_D_STREET_1[21];
    char out_s_D_STREET_2[21];
    char out_s_D_CITY[21];
    char out_s_D_STATE[3];
    char out_s_D_ZIP[10];
    char out_s_C_FIRST[17];
    char out_s_C_MIDDLE[3];
    char out_s_C_LAST[17];
    char out_s_C_STREET_1[21];
    char out_s_C_STREET_2[21];
    char out_s_C_CITY[21];
    char out_s_C_STATE[3];
    char out_s_C_ZIP[10];
    char out_s_C_PHONE[17];
    char out_s_C_CREDIT[3];
    char out_s_C_DATA[201];
};
struct ORDERSTATUSDATA
{
    int in_s_C_ID;
    int in_s_W_ID;
    short in_s_D_ID;
    char in_s_C_LAST[17];

    double out_s_C_BALANCE;
    long long out_s_O_ENTRY_D_time;
    int out_s_C_ID;
    int out_s_O_ID;
    short out_s_O_CARRIER_ID;

```



```

short out_s_ol_cnt;
struct out_oitems_struct {
    long long s_OL_DELIVERY_D_time;
    double s_OL_AMOUNT;
    int s_OL_I_ID;
    int s_OL_SUPPLY_W_ID;
    short s_OL_QUANTITY;
} out_item[15];
short out_s_transtatus;
short out_deadlocks;
char out_s_C_FIRST[17];
char out_s_C_MIDDLE[3];
char out_s_C_LAST[17];
};
struct DELIVERYDATA
{
    long long in_s_O_DELIVERY_D_time;
    int in_s_W_ID;
    short in_s_O_CARRIER_ID;
    int out_s_O_ID[10];
    short out_s_transtatus;
    short outdeadlocks;
};
struct STOCKLEVELDATA
{
    int in_s_threshold;
    int in_s_W_ID;
    short in_s_D_ID;
    int out_s_low_stock;
    short out_s_transtatus;
    short out_deadlocks;
};

// MISCELLANEOUS HELPER FUNCTIONS
inline void appendText(char **string,char *text);
inline void appendText(char **string,char *text,int length,int justify);
inline void appendChar(char **string,char byte);
inline void DEBUGMSG(FILE * debugFile, char * message);
inline void appendSpaces(char **string,int spaces);
inline void calcOutDateTime(const INT64b value,datetime *timestamp);
inline int copyOutPhone(char *buffer,char *value,int len);
inline bool copyInMoney64(const char * value,INT64 *number);
inline int copyInMoney(const char *value);
inline void copyOutMoney64(char *buffer,INT64b value,unsigned int len);
inline int copyOutDateTime(char *buffer,INT64b value);
inline int copyOutDate(char *buffer,INT64b value);
inline int copyOutTime(char *buffer,INT64b value);
inline int copyOutDecimal64(char *buffer,INT64b value,unsigned int len);
inline UINT16b changeOrder16(UINT16b value);
inline UINT32b changeOrder32(UINT32b value);
inline UINT64b changeOrder64(UINT64b value);
inline INT16b changeOrder16(INT16b value);
inline INT32b changeOrder32(INT32b value);
inline INT64b changeOrder64(INT64b value);
//
// Name      : appendText
// Description :
// Append text to string
// Parameters :
// char ** - string point to append to
// char * - text to append
// Returns   :
// None
// Comments  :
//
inline void appendText(char **string,char *text)
{

```

```

while(*text)
{
    *(*string)++ = *text++;
}
**string='\0';
return;
}
//
// Name      : appendText
// Description :
// Append text to string
// Parameters :
// char ** - string point to append to
// char * - text to append
// Returns   :
// None
// Comments  :
//
// int - total field length including
// blank spaces
// int - justify flag
// Returns   :
// None
// Comments  :
//
// right justify
// left justify

inline void appendText(char **string,char *text,int length,int justify)
{
    int byteCount = 0;

    if(justify)
    {
        while(*text)
        {
            *(*string)++ = *text++;
            byteCount++;
        }

        //append blank spaces if text is less than length at end
        for(byteCount;byteCount < length;byteCount++)
            *(*string)++ = ' ';
    }
    else
    {
        long long textLen = strlen(text);
        for(textLen;textLen < length;textLen++)
            *(*string)++ = ' ';
        while(*text)
            *(*string)++ = *text++;
    }
    **string='\0';
}
// Name      : appendChar
// Description :
// Append text to string
// Parameters :
// char ** - string point to append to
// char * - text to append
// Returns   :
// None
// Comments  :
//
inline void appendChar(char **string,char byte)
{
    *(*string)++ = byte;
    **string='\0';
    return;
}

```

```

//
// Name          : appendSpaces
// Description    :
//                appends buffer spaces to result
// Parameters    :
//                **htmlPage
// Returns       :
//                amount of characters
// the function appened
//                to the html page
// Comments      :
//
inline void appendSpaces(char **string,int spaces)
{
    for(int index=0;index<spaces;index++)
    {
        *(*string)++ = ' ';
    }
    **string='\0';
}
//
// Name          : appendCustData
// Description    :
//                appends cust data buffer to result
// Parameters    :
//                **htmlPage
// Returns       :
//                Adds a newline
// character every 50 characters displayed.
// Comments      :
//
inline void appendCustData(char **string,char *text)
{
    short byteCount = 0;
    while(*text)
    {
        *(*string)++ = *text++;
        byteCount++;
        if((byteCount % 50) == 0)
        {
            *(*string)++ = '\n';
            *(*string)++ = ' ';
            *(*string)++ = ' ';
            *(*string)++ = ' ';
            *(*string)++ = ' ';
            *(*string)++ = ' ';
            *(*string)++ = ' ';
            *(*string)++ = ' ';
            *(*string)++ = ' ';
            *(*string)++ = ' ';
        }
        **string='\0';
    }
}
//
// calcOutDateTime
// Title         : Calculate date & time data out of class array
// Parameters    : INT64b - date & time expressed in seconds
//                datetime * - timestamp
// Return Value  : None
// Comments      :
//
inline void calcOutDateTime(const INT64b value,datetime *timestamp)
{

```

```

// fixed days in each month (FEB 29 is special case)
static int daysInMonth[12] =
{31,28,31,30,31,30,31,31,30,31,30,31};
// mask out EPOC seconds
int dateValue = ((int) (value & 0xffffffff)) +
                (SECONDS_IN_DAY -
                (GMT_OFFSET * SECONDS_IN_HOUR));
int offset = (int) (value >> 32);
// break out the seconds
int hms = dateValue % SECONDS_IN_DAY;
int days = dateValue / SECONDS_IN_DAY;
int years = (days - 1) / DAYS_IN_YEAR;
int leaps = years / YEARS_IN_LEAP;
int daysUsed = (years * DAYS_IN_YEAR) + leaps;
// adjust the number of days to account for calculated years
days = days - daysUsed;
// set the starting year, month, and day
timestamp->day = 1;
timestamp->month = 1;
timestamp->year = START_YEAR + years;
// is the current year a leap year
int leap = !(timestamp->year % YEARS_IN_LEAP);
// apply remaining days based on days in months
int daysInCurrentMonth;
while(days)
{
    // get days in current month
    daysInCurrentMonth =
daysInMonth[timestamp->month - 1];
    if(timestamp->month == 2 && leap)
        daysInCurrentMonth =
daysInCurrentMonth + 1;

    // days > days in current month
    if(days > daysInCurrentMonth)
    {
        // increment month
        timestamp->month += 1;
        days = days -
daysInCurrentMonth;

        // month exceeds months in year
        if(timestamp->month >
MONTHS_IN_YEAR)
        {
            // increment year and
            timestamp->year += 1;

            // are we now on a leap
            leap =
!(timestamp->year % YEARS_IN_LEAP);
        }
    }
    else
    {
        // set day of month to remaioning
        timestamp->day = days; days = 0;
    }
}
// set time values to remaining seconds
timestamp->hour = hms / SECONDS_IN_HOUR;
hms = hms % SECONDS_IN_HOUR;
timestamp->minute = hms / SECONDS_IN_MINUTE;
timestamp->second = hms % SECONDS_IN_MINUTE;
return;
}

```

```

//
// copyOutZip
//
// Title      : Copy zip data out of class array
// Parameters  : char * - buffer to copy zip string into
//
// Return Value : int - Length of copy
// Comments    :
//
inline int copyOutZip(char *buffer,char *value,int len = DEFAULT_ZIP_LEN)
{
    int index      = 0;
    int bufferPos  = 0;
    // add each digit of zip number to buffer inserting delimiter at 5
    while(value[index] && bufferPos < len)
    {
        if(index == 5)
            buffer[bufferPos++] = ZIP_DELIMITER;
        buffer[bufferPos++] = value[index++];
    }
    // space fill to the required length
    while(bufferPos < len)
        buffer[bufferPos++] = CHAR_FILL;
    buffer[bufferPos] = NULL;
    return len;
}
//
// copyOutPhone
//
// Title      : Copy phone data out of class array
// Parameters  : char * - buffer to copy phone string into
//
// Return Value : int - Length of copy
// Comments    :
//
inline int copyOutPhone(char *buffer,char *value,int len =
DEFAULT_PHONE_LEN)
{
    int index      = 0;
    int bufferPos  = 0;
    // add each digit of phone number to buffer inserting delimiter before
6, 9, and 12
    while(value[index] && index < len)
    {
        switch(index)
        {
            case 6:
            case 9:
            case 12:
                // insert delimiter
                buffer[bufferPos++] = PHONE_DELIMITER;
            default:
                // add phone digit to buffer
                buffer[bufferPos++] = value[index++];
        }
    }

    // space fill to the required length
    while(bufferPos < len)
        buffer[bufferPos++] = CHAR_FILL;
    buffer[bufferPos] = '\0';
    return len;
}
//
// copyInMoney64
//
// Title      : Copy money data into class array

```

```

// Parameters      : const char * - value string
// Return Value    : INT64b integer value
// Comments        :
//
inline bool copyInMoney64(const char * value,INT64b *number)
{
    //INT64b number      = 0;
    int index           = 0;
    int decimal         = 0;
    int decimals        = 0;
    int digitsAfterDec  = 0;
    bool negativeFlag   = false;
    // convert each digit to a numeric portion
    while(value[index])
    {
        // handle $ . - All the rest assumed numeric
        switch(value[index])
        {
            case MONEY_SYMBOL:
                // ignore $ sign
                break;
            case NEGATIVE_SYMBOL:
                // set negative flag
                negativeFlag = true;
                break;
            case DECIMAL_SYMBOL:
                // set decimal
                decimal=1;
                decimals++;
                if(decimals > 1)
                    //more than 1 decimal point found
                    return false;
                break;
            default:
                // adjust decimal places
                decimal = decimal * 10;
                // add digit to running total
                if(value[index] >= '0' && value[index] <= '9')
                {
                    if(decimal)
                        if(++digitsAfterDec >
2)
                            return false;
                    *number = (*number * 10) +
(value[index] - '0');
                }
                else
                {
                    //non-numeric field inserted
                    return false;
                }
                index++;
        }
    }

    // apply decimal where decimal not found
    if(decimal < 100)
    {
        if(decimal)
        {
            *number *= (100 / decimal);
        }
        else
        {
            *number *= 100;
        }
    }
    // make negative

```

```

        if(negativeFlag)
            *number = *number * (-1);
        return true;
    }
    //
    // copyInMoney
    //
    // Title          : Convert char string money field to double
    // Parameters     : const char * - value string
    // Return Value   : double integer value
    // Comments      :
    //
    inline int copyInMoney(const char *value)
    {
        char buf[20];
        int i,j,decimalFound,digitsAfterDecimal=0;
        int decimal=0;
        //walk past $ if present in char string
        if(*value == '$')
            *value++;
        int len=(int)strlen(value);
        for (i=0;i<len;i++)
        {
            if(value[i] == '.')
            {
                decimalFound++;
                if(decimalFound > 1)
                    return -1;
            }
            if(value[i] == '-')
            {
                if (value[i] != '-')
                {
                    if(decimal)
                    {
                        if(digitsAfterDecimal<2)
                            digitsAfterDecimal++;
                        else
                            return -1;
                    }
                    buf[j++] = value[i];
                }
            }
        }
        int amount = atoi(buf);
        return amount;
    }
    //
    // copyOutMoney64
    //
    // Title          : Copy money data out of class array
    // Parameters     : char * - buffer to copy string 64 bit money into
    //                : INT64b - value
    //                : unsigned len - max number of
    //                : bytes to copy
    // Return Value   : int - Length of copy
    // Comments      :
    //
    inline void copyOutMoney64(char *buffer,INT64b value,unsigned int len =
    DEFAULT_MONEY64_LEN)
    {
        unsigned int    index          = len;
        int              places        = 0;
        bool             negativeFlag   = false;
        bool             moneyFlag      = true;
        // NULL terminate string
        buffer[index] = NULL;
    }
    // check length > 0
    if(!index) return len;
    // handle negative value
    if(value < 0)
    {
        negativeFlag = true;
        value = value * (-1);
    }
    // break off each digit from value, fill if needed
    do
    {
        if(value)
        {
            // get next digit and add to buffer
            buffer[--index] = (char) (value % 10 + '0');
            value /= 10; places++;
            if(places == 2 && index)
            {
                places++;
                buffer[--index] =
                DECIMAL_SYMBOL;
            }
        }
        else
        {
            // add zeros to first place before decimal point
            on (i.e. 0.00)
            if(places < 2 || places == 3)
            {
                buffer[--index] =
                ZERO_SYMBOL;
            }
            else
            {
                // add the decimal point
                if(places == 2)
                {
                    buffer[--index] =
                    DECIMAL_SYMBOL;
                }
                else
                {
                    // add the negative
                    if(negativeFlag)
                    {
                        negativeFlag
                    }
                }
            }
        }
    }
    while (value || places);
    // add the
    if(moneyFlag)
    {
        moneyFlag = false;
        buffer[--index] = MONEY_SYMBOL;
    }
    else
    {
        buffer[--index] = NUMERIC_FILL;
    }
}

```

```

// need to trace place for decimal point and
zero fill
places++;
} while(index);
//return len;
}
//
// copyOutDateTime
//
// Title : Copy date & time data out of class array
// Parameters : char * - buffer to copy date & time string into
// INT64b - value
// Return Value : int - Length of copy
// Comments : Fixed length
//
inline int copyOutDateTime(char *buffer,INT64b value)
{
    datetime timestamp;
    // break value into time/date components
    calcOutDateTime(value,&timestamp);
    // put month into buffer
    *buffer++ = (char)((timestamp.month / 10) + '0');
    *buffer++ = (char)((timestamp.month % 10) + '0');
    *buffer++ = DATE_DELIMITER;
    // put day into
    buffer
    *buffer++ = (char)((timestamp.day / 10) + '0');
    *buffer++ = (char)((timestamp.day % 10) + '0');
    *buffer++ = DATE_DELIMITER;
    // put year into buffer
    int year = timestamp.year;
    *buffer++ = (char)((year / 1000) + '0');
    year = year % 1000;
    *buffer++ = (char)((year / 100) + '0');
    year = year % 100;
    *buffer++ = (char)((year / 10) + '0');
    *buffer++ = (char)((year % 10) + '0');
    *buffer++ = CHAR_FILL;
    // put hour into buffer
    *buffer++ = (char)((timestamp.hour / 10) +
'0');
    *buffer++ = (char)((timestamp.hour % 10) +
'0');
    *buffer++ = TIME_DELIMITER;
    // put minute into buffer
    *buffer++ = (char)((timestamp.minute / 10) +
'0');
    *buffer++ = (char)((timestamp.minute % 10) +
'0');
    *buffer++ = TIME_DELIMITER;
    // put second into buffer
    *buffer++ = (char)((timestamp.second / 10) +
'0');
    *buffer++ = (char)((timestamp.second % 10) +
'0');
    *buffer = NULL; return DEFAULT_DATETIME_LEN;
}
//
// copyOutTime
//
// Title : Copy date data out of class array
// Parameters : char * - buffer to copy date string into
// INT64b - value
// Return Value : int - Length of copy
// Comments : Fixed length
//
inline int copyOutDate(char *buffer,INT64b value)
{
    datetime timestamp;
    // break value into time/date components
    calcOutDateTime(value,&timestamp);
    // put month into buffer
    *buffer++ = (char)((timestamp.month / 10) + '0');
    *buffer++ = (char)((timestamp.month % 10) + '0');
    *buffer++ = DATE_DELIMITER;
    // put day into buffer
    *buffer++ = (char)((timestamp.day / 10) + '0');
    *buffer++ = (char)((timestamp.day % 10) + '0');
    *buffer++ = DATE_DELIMITER;
    // put year into buffer
    int year = timestamp.year;
    *buffer++ = (char)((year / 1000) + '0');
    year = year % 1000;
    *buffer++ = (char)((year / 100) + '0');
    year = year % 100;
    *buffer++ = (char)((year / 10) + '0');
    *buffer++ = (char)((year % 10) + '0');
    *buffer++ = CHAR_FILL;
    *buffer = NULL;

    return DEFAULT_DATE_LEN;
}
//
// copyOutTime
//
// Title : Copy time data out of class array
// Parameters : char * - buffer to copy time string into
// INT64b - value
// Return Value : int - Length of copy
// Comments : Fixed length TBD
//
inline int copyOutTime(char *buffer,INT64b value)
{
    datetime timestamp;
    // break value into time/date components
    calcOutDateTime(value,&timestamp);
    // put hour into buffer
    *buffer++ = (char)((timestamp.hour / 10) + '0');
    *buffer++ = (char)((timestamp.hour % 10) + '0');
    *buffer++ = TIME_DELIMITER;
    // put minute into buffer
    *buffer++ = (char)((timestamp.minute / 10) + '0');
    *buffer++ = (char)((timestamp.minute % 10) + '0');
    *buffer++ = TIME_DELIMITER;
    // put second into buffer
    *buffer++ = (char)((timestamp.second / 10) + '0');
    *buffer++ = (char)((timestamp.second % 10) + '0');
    *buffer = NULL; return DEFAULT_TIME_LEN;
}
//
// copyOutDecimal64
//
// Title : Copy decimal data out of class array
// Parameters : char * - buffer to copy string 64 bit money into
// INT64b - value
// unsigned len - max number of
bytes to copy
// Return Value : int - Length of copy
// Comments :
//
inline int copyOutDecimal64(char *buffer,INT64b value,unsigned int len =
DEFAULT_DECIMAL64_LEN)
{
    unsigned int index = len;
    int places
= 0;
    bool negativeFlag = false;
    // NULL terminate string

```

```
buffer[index] = NULL;
// check length > 0
if(!index) return len;
// handle negative value
if(value < 0)
{
    negativeFlag = true;
    value = value * (-1);
}
// break off each digit from value, fill if needed
do
{
    if(value)
    {
        // get next digit and add to buffer
        buffer[--index] = (char) (value % 10 + '0');
        value /= 10; places++;
        if(places == 2 && index)
        {
            places++;
            buffer[--index] =
DECIMAL_SYMBOL;
        }
    }
    else
    {
        // add zeros to first place before decimal point
        on (i.e. 0.00)
        if(places < 2 || places == 3)
        {
            buffer[--index] =
ZERO_SYMBOL;
        }
        else
        {
            // add the decimal point
            if(places == 2)
            {
                buffer[--index] =
DECIMAL_SYMBOL;
            }
            else
            {
                // add the negative
                indicator
                if(negativeFlag)
                {
                    negativeFlag
= false;
buffer[--index] = NEGATIVE_SYMBOL;
                }
                else buffer[--index] =
NUMERIC_FILL;
            }
        }
        // need to trace place for decimal point and
zero fill
        places++;
    }
} while(index);
return len;
}
////////////////////////////////////
// Macros
////////////////////////////////////
using namespace std;
#ifdef _DEBUG
```

```
int debugFlag = 1;
#else
int debugFlag = 0;
#endif
inline BYTE8b *debugFileName(BYTE8b *filePath)
{
    BYTE8b *fileName = filePath + strlen(filePath);
    while(fileName != filePath)
    {
        if(*fileName == '/' || *fileName == '\\ && *(fileName +
1))
            return (fileName + 1);
        fileName--;
    }
    return filePath;
}
#define DEBUGADDRESS(POINTER) hex << (void *) POINTER << dec
#define ERRORMSG(TEXT)
\
EnterCriticalSection(&errorMutex);
\
\
\
errorStream
<< debugFileName(__FILE__)
\
<< "|" <<
__TIMESTAMP__ << "|" << __LINE__ << "|"
\
<< _getpid()
<< " " << GetCurrentThreadId() << "|"
\
<< TEXT;
\
errorStream.flush();
\
LeaveCriticalSection(&errorMutex);
#ifdef _DEBUG
#define DEBUGMSG(TEXT)
\
EnterCriticalSection(&debugMutex);
\
\
\
debugStream << debugFileName(__FILE__)
\
<< "|" <<
__TIMESTAMP__ << "|" << __LINE__ << "|"
\
<< _getpid()
<< " " << GetCurrentThreadId() << "|"
\
<< TEXT;
\
debugStream.flush();
\
LeaveCriticalSection(&debugMutex);
```

```

#define DEBUGSTRING(TEXT,LENGTH)
        debugVarString(TEXT,LENGTH)

#else
#define DEBUGMSG(TEXT);
#define DEBUGSTRING(TEXT,LENGTH);

#endif
#endif /* _COMMON_TPCC */

```

tpccsapi/tpccsapi.def

```

; tpccsapi.def : declares the module parameters for the DLL.
LIBRARY "tpccsapi"
EXPORTS
    HttpExtensionProc
    GetExtensionVersion
    TerminateExtension

```

tpccsapi/tpccsapi.hpp

```

/*
*****
** Project      : AIX
** Component    : Performance/TPC-W Benchmark
** Name         : tpccsapi.hpp
** Title        : ISAPI interface for tpcc
*****
** Copyright (c) 2001,2002 IBM Corporation
** All rights reserved
*****
** History      :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments    :
**
*****
*/
#ifndef __tpccSAPI_hpp__
#define __tpccSAPI_hpp__
#include <windows.h>
#include <httpext.h>
#include <tpcc.h>
#include <htmlPhraser.h>
#include <iomanip>
#include <db2tpcc.h>
#include <comsvcs.h>
// Terminal struct
// Terminal struct
struct TERM_ENTRY
{
    int          terminalID;
    bool         terminalInUse;
    int          w_id;
    short       d_id;
};
// COM interface
// COM interface
struct COM_HANDLE
{

```

```

    Itpcc_com *comHandle;
    char        *txnBuffer;
    int          size;
};
// TXN handle
// TXN handle
struct TXN_HANDLE
{
    char        htmlPage[MAX_HTML_PAGE_LEN];
    char        htmlHeader[MAX_HTML_HEADER_LEN];
    char        *urlString;
    //user data
    int          w_id;
    int          d_id;
    int          sync_id;
    int          term_id;
    int          conn_id;
    COM_HANDLE  comInterface;
};
struct DLVYQUEUEDATA
{
    int          warehouse;
    short       in_s_0_CARRIER_ID;
    struct _timeb enqueueTime;
};
// Definitions
// Definitions
#define INVALID_ITEM 100
#define HEADER
"Content-Type:text/html\r\nContent-Length: %d\r\nConnection:
Keep-Alive\r\n\r\n"
#define TLS_NULL
0xFFFFFFFF
#define ACCESS_TIMEOUT 3600000
//One hour in milli
seconds
#define DELIVERY_LOG_SUCCESS_STR "--Tran %d
Queue %d.%03d Start %d.%03d\r\nW_ID: %d CARRIER_ID: %d
%s\r\nend-time: %d.%03d\r\n"
// Function Prototypes
// Function Prototypes
int initDlvy();
int initTxnHandle(TXN_HANDLE *txnHandle);
int closeTxnHandle(TXN_HANDLE *txnHandle);
int readRegistryValues();
int getTerminal(int terminal,TXN_HANDLE *txnHandle);
int assignTerminal(TXN_HANDLE *txnHandle);
int getDBInstance();
void doHtml(TXN_HANDLE *txnHandle);
int doLoginForm(htmlPhraser *commandBlock,TXN_HANDLE *txnHandle);
int doLoginResults(htmlPhraser *commandBlock,TXN_HANDLE *txnHandle);
int doNewOrderForm(htmlPhraser *commandBlock,TXN_HANDLE *txnHandle);
int doNewOrderResults(htmlPhraser *commandBlock,TXN_HANDLE *txnHandle);
int doPaymentForm(htmlPhraser *commandBlock,TXN_HANDLE *txnHandle);
int doPaymentResults(htmlPhraser *commandBlock,TXN_HANDLE *txnHandle);
int doOrderStatusForm(htmlPhraser *commandBlock,TXN_HANDLE *txnHandle);
int doOrderStatusResults(htmlPhraser *commandBlock,TXN_HANDLE *txnHandle);

```

```

int doDeliveryForm(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle);
int doDeliveryResults(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle);
int doStockForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doStockResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doExit(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doLoginErrorPage(char *htmlPage, char *message);
int doNewOrderErrorPage(char *htmlPage, char *message, htmlPhraser
*commandBlock, TXN_HANDLE *txnHandle);
int doPaymentErrorPage(char *htmlPage, char *message, htmlPhraser
*commandBlock, TXN_HANDLE *txnHandle);
int doOrderStatusErrorPage(char *htmlPage, char *message, htmlPhraser
*commandBlock, TXN_HANDLE *txnHandle);
int doDeliveryErrorPage(char *htmlPage, char *message, htmlPhraser
*commandBlock, TXN_HANDLE *txnHandle);
int doStockErrorPage(char *htmlPage, char *message, htmlPhraser
*commandBlock, TXN_HANDLE *txnHandle);
void dlvyThreadEntry(void *);
int queueDlvyTxn(int warehouse, short carrier_id);
int appendButtons(char *htmlPage);
int appendItems(char *htmlPage, short itemCount, short cmdIDStart);
int appendHiddenFields(char *htmlPage, TXN_HANDLE *txnHandle);
int displayStatus(char *htmlPage, int rc);
#endif

```

tpccsapi/htmlPhraser.cpp

```

/////////////////////////////////////////////////////////////////
// htmlPhraser.cpp
/////////////////////////////////////////////////////////////////
// Class implementation of htmlPhraser.
// This class will take a query string and break it into a series
// of constituent parts
/////////////////////////////////////////////////////////////////
#include "htmlPhraser.h"
/////////////////////////////////////////////////////////////////
// htmlPhraser::htmlPhraser
/////////////////////////////////////////////////////////////////
// Title : Constructor
// Parameters : char * query string
// Return Value : None
// Comments :
/////////////////////////////////////////////////////////////////
htmlPhraser::htmlPhraser(char *queryString)
{
    // initialize query values
    iCustomerIdFlag = iCarrierNumFlag = iStockThresholdFlag = false;
    // this initializes the query list to NULL's. This means that
    // characters being added are overwriting null characters and
    // therefore the string will be null terminated implicitly.

    memset(iQueryValues, NULL, (MAX_FIELD_NUM *
MAX_FIELD_LEN));
    // controls
    char queryChar = NULL;
    int queryIndex = -1;
    int valueIndex = -1;
    // process each character of query string
    while(*queryString)
    {
        // check for special case characters
        if(queryChar)
        {
            // a percentage sign would indicate a token
            if(*queryString != '%')
            {
                // a plus sign represents a space

```

```

if(*queryString == '+')
{
    queryChar = '+';
    *queryString++;
}
else queryChar = *queryString++;
}
else queryChar =
convertQueryToken(&queryString);
}
else queryChar = '&';
// handle query reference (&)
if(queryChar == '&')
{
    // reset value index
    valueIndex = -1;
    // do we have a numeric query reference
    if(*queryString >= '0' && *queryString <=
'9')
    {
        // numeric query id
        queryIndex =
(( *queryString - '0') *
10) + (*queryString + 1) - '0';

        // walk past the two command
        // characters
        queryString += 2;

        // validate query value
        if(queryIndex >
MAX_QUERY_ID)
            queryIndex = -1;
    }
    else queryIndex = -1;
    // finished processing for query reference
    continue;
}
// we have a query reference but need to wait until we see
'='

// before accepting value
if(valueIndex == -1)
{
    // we are waiting for '='
    if(queryChar == '=')
    {
        valueIndex = 0;
        // set query string flags
        switch(queryIndex)
        {
            case C_ID:
                iCustomerIdFlag = true;
                break;
            case CARRIER_NUM:
                iCarrierNumFlag =
true; break;
            case STK_THRESHOLD:
                iStockThresholdFlag =
true; break;
            default: break;
        }
    }
    // finishes looking for '='
    continue;
}
// add each character to the query value
if(queryIndex > -1 && valueIndex > -1)

```



```

        {
            // we are processing a query value
            if(valueIndex < MAX_FIELD_LEN)
            {
                // we have not exceeded max line len

iQueryValues[queryIndex][valueIndex++] = queryChar;
                continue;
            }
        }
        return;
    }
}
// htmlPhraser::getCommandId
// Title : Returns the page command
// Parameters : None
// Return Value : int - page command
// Comments :
// =====
int htmlPhraser::getCommandId()
{
    // return command numeric code
    switch(*iQueryValues[COMMAND_ID])
    {
        case NEW_ORDER_CODE:
            if(iCustomerIdFlag)
                return
COMMAND_NEW_ORDER_RESULTS;
            else return COMMAND_NEW_ORDER;
        case PAYMENT_CODE:
            if(iCustomerIdFlag)
                return COMMAND_PAYMENT_RESULTS;
            else return COMMAND_PAYMENT;
        case ORDER_STATUS_CODE:
            if(iCustomerIdFlag)
                return
COMMAND_ORDER_STATUS_RESULTS;
            else return COMMAND_ORDER_STATUS;
        case DELIVERY_CODE:
            if(iCarrierNumFlag)
                return COMMAND_DELIVERY_RESULTS;
            else return COMMAND_DELIVERY;
        case STOCK_CODE:
            if(iStockThresholdFlag)
                return COMMAND_STOCK_RESULTS;
            else return COMMAND_STOCK;
        case MENU_CODE:
            return COMMAND_LOGIN_RESULTS;
        case EXIT_CODE:
            return COMMAND_EXIT;
        default:
            return COMMAND_LOGIN;
    };

    // should not get here
    return COMMAND_LOGIN;
}
// =====
// htmlPhraser::validate
// Title : validate url parameter list for all txn types
// Parameters : int - txn type
// Return Value : int - error code
// Comments :
// =====

```

```

int validate(int txnType)
{
    return 0;
}
// =====
// htmlPhraser::convertQueryToken
// Title : Returns the page command
// Parameters : None
// Return Value : int - page command
// Comments :
// =====
char htmlPhraser::convertQueryToken(char **queryString)
{
    char queryChar = NULL;
    // skip over %
    (*queryString)++;
    // look at first character
    switch(**queryString)
    {
        case '2':
            {
                // what follows?
                (*queryString)++;
                switch(**queryString)
                {
                    case '1':
                        queryChar = '!';
                        break;
                    case '3':
                        queryChar = '#';
                        break;
                    case '4':
                        queryChar = '$';
                        break;
                    case '5':
                        queryChar = '%';
                        break;
                    case '6':
                        queryChar = '&';
                        break;
                    case '8':
                        queryChar = '(';
                        break;
                    case '9':
                        queryChar = ')';
                        break;
                    case 'B':
                        queryChar = '+';
                        break;
                    case 'C':
                        queryChar = ',';
                        break;
                    case 'F':
                        queryChar = '/';
                        break;
                    case ' ':
                        queryChar = ' ';
                        break;
                }
            }
        case '3':
            {
                // what follows?
                (*queryString)++;
                switch(**queryString)

```

```

        {
        case 'A':
            queryChar = '!';
            break;
        case 'B':
            queryChar = '!';
            break;
        case 'D':
            queryChar = '=';
            break;
        case 'F':
            queryChar = '?';
            break;
        case ' ':
            queryChar = ' ';
            break;
        }
    }
    break;
case '4':
    {
        // what follows?
        (*queryString)++;
        switch(**queryString)
        {
        case '0':
            queryChar = '@';
            break;
        case ' ':
            queryChar = ' ';
            break;
        }
    }
    break;
case '5':
    {
        // what follows?
        (*queryString)++;
        switch(**queryString)
        {
        case 'B':
            queryChar = '[';
            break;
        case 'D':
            queryChar = ']';
            break;
        case 'E':
            queryChar = '^';
            break;
        case ' ':
            queryChar = ' ';
            break;
        }
    }
    break;
case '7':
    {
        // what follows?
        (*queryString)++;
        switch(**queryString)
        {
        case 'B':
            queryChar = '{';
            break;
        case 'C':
            queryChar = '|';
            break;
        }
    }

```

```

        case 'D':
            queryChar = '}';
            break;
        case 'E':
            queryChar = '~';
            break;
        case ' ':
            queryChar = ' ';
            break;
        }
    }
    break;
    case '+':
        queryChar = '+';
        break;
    }
    // advance pointer and return
    (*queryString)++; return queryChar;
}
/////////////////////////////////////////////////////////////////

```

tpccsapi/StdAfx.cpp

```

// stdafx.cpp : source file that includes just the standard includes
// tpccsapi.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information
#include "stdafx.h"
// TODO: reference any additional headers you need in STDAFX.H
// and not in this file

```

tpccsapi/tpccsapi.cpp

```

/*
*****
** Project      : AIX
** Component    : Performance/TPC-C Benchmark
** Name         : tpccsapi.cpp
** Title        : TPC-C html processing
*****
** Copyright (c) 2003 IBM Corporation
** All rights reserved
*****
** History      :
**      Developed at IBM Austin by the AIX RS/6000
**      performance group.
**
** Comments     :
*****
*/
#include "stdafx.h"
#include "..\tpccCom\tpccCom.h"
#include "..\tpccCom\tpccCom_i.c"
#include <tpccsapi.hpp>
// For custom assert and trace handling with WebDbg.exe
[ module(name="tpccsapi", type="dll") ];
[ emitidl(restricted) ];
#define _WIN32_DCOM
/////////////////////////////////////////////////////////////////
// Globals
/////////////////////////////////////////////////////////////////
int          maxDataSize;
//max struct size of all txn(s)
int          numUsers;
//number of users that client will service.
int          dlvyQueueLen;
//static length of dlvy queue

```

```

int          dlvyThreads;
//number of dlvy threads to create
int          dlvyBufferFreeSlots;          //length of dlvy txn
queue
int          dlvyBufferSlotIndex;          //index into next
available slot in dlvy txn queue
int          dlvyBufferThreadIndex;        //thread
index into dlvy txn queue
int          nullDB;
//null db on client(bypass com call).
int          trace;
static DWORD threadLSIndex;
//isapi thread local storage index
CRITICAL_SECTION isapiLock;
//isapi lock
CRITICAL_SECTION errorLock;
//error log file lock.
CRITICAL_SECTION termLock;
//terminal array lock.
CRITICAL_SECTION dlvyQueueLock;
//dlvy queue critical section lock
HANDLE          dlvyThreadDone =
INVALID_HANDLE_VALUE;          //dlvy thread exit event
HANDLE          dlvyThreadSemaphore
= INVALID_HANDLE_VALUE;        //dlvy thread wrk to do semaphore
int
dlvyThreadID = 0;
struct DLVYQUEUEDATA *dlvyQueue;
//dlvy queue
HANDLE          *dlvyThreadHandles;
//ptr to array of thread handles
TERM_ENTRY     *termArray;
//array of terminal entries to store each users info.
int            termNextFree;
//next available slot in terminal array
FILE           *htmlDebug          = NULL;
//html debug file
FILE           *errorLog           = NULL;
//error file
FILE           *htmlTrace          = NULL;

ofstream debugStream;
ofstream errorStream;
CRITICAL_SECTION debugMutex;
CRITICAL_SECTION errorMutex;
char           dlvyLogPath[128]    = {NULL};
char           errorLogFile[128]   = {NULL};
char           htmlTraceLogFile[128] = {NULL};
char           dbName[64]          = {NULL};
char           dbType[16]          = {NULL};
typedef INT (*CONNECT_PTR)(char *dbName,void **connectHandle);
typedef INT (*DISCONNECT_PTR)(void *connectHandle);
typedef INT (*DLVY_FUNC_PTR)(dlvy_wrapper *dlvy,void
*connectHandle);
typedef INT (*NORD_FUNC_PTR)(nord_wrapper *nord,void
*connectHandle);
typedef INT (*PYMT_FUNC_PTR)(paym_wrapper *pymt,void
*connectHandle);
typedef INT (*ORDS_FUNC_PTR)(ords_wrapper *ords,void *connectHandle);
typedef INT (*STOK_FUNC_PTR)(stok_wrapper *stok,void *connectHandle);
HINSTANCE      dbInstance;
CONNECT_PTR    db_connect;
DISCONNECT_PTR db_disconnect;
DLVY_FUNC_PTR dlvyCall;

//////////
// Page functions arrays
//////////

```

```

typedef int (*pageFuncPtr) (htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle);
pageFuncPtr htmlPageFunctions[MAX_TRANSACTIONS] =
{
    {doLoginForm},
    {doNewOrderForm},
    {doPaymentForm},
    {doOrderStatusForm},
    {doDeliveryForm},
    {doStockForm},
    {doExit},
    {doLoginResults},
    {doNewOrderResults},
    {doPaymentResults},
    {doOrderStatusResults},
    {doDeliveryResults},
    {doStockResults}
};
extern "C" DWORD WINAPI
HttpExtensionProc(LPEXTENSION_CONTROL_BLOCK lpECB)
{
    struct TXN_HANDLE *txnHandle = NULL;
    txnHandle = (TXN_HANDLE *) TlsGetValue(threadLSIndex);
    if(txnHandle == NULL)
    {
        int rc = initTxnHandle(&txnHandle);
        if (rc != OK)
        {
            char response[256]; char htmlHeader[256];
            sprintf(response,"ERROR: Init txnHandle
function failed.\n");
            size_t htmlPageLen = strlen(response);
            //add content length and keep alive header
            sprintf(htmlHeader,HEADER,htmlPageLen);
            lpECB->ServerSupportFunction(lpECB->ConnID,HSE_REQ_SEND_RESPO
NSE_HEADER,"200 OK",NULL,(DWORD*)htmlHeader);
            lpECB->WriteClient(lpECB->ConnID,response,(LPDWORD)&htmlPageLen,0
);
            return
HSE_STATUS_SUCCESS_AND_KEEP_CONN;
        }
        txnHandle = (TXN_HANDLE *)
TlsGetValue(threadLSIndex);
        if (txnHandle == NULL)
        {
            char response[256]; char htmlHeader[256];
            sprintf(response,"ERROR: Unable to retrieve
txnHandle from TLS.\n");
            size_t htmlPageLen = strlen(response);
            //add content length and keep alive header
            sprintf(htmlHeader,HEADER,htmlPageLen);
            lpECB->ServerSupportFunction(lpECB->ConnID,HSE_REQ_SEND_RESPO
NSE_HEADER,"200 OK",NULL,(DWORD*)htmlHeader);
            lpECB->WriteClient(lpECB->ConnID,response,(LPDWORD)&htmlPageLen,0
);
            return
HSE_STATUS_SUCCESS_AND_KEEP_CONN;
        }
        try
        {
            txnHandle->urlString =
(char*)lpECB->lpszQueryString;

```

```

        DEBUGMSG("calling doHtml() w/ query string:" <<
txnHandle->urlString << endl);
        doHtml(txnHandle);

        size_t htmlPageLen;
        htmlPageLen = strlen(txnHandle->htmlPage);
        if(htmlPageLen >= 4096)
        {
            ERRORMSG("WARNING: HTML PAGE IS
>= 4096!, page size:"<<htmlPageLen<<endl);
        }
        //add content length and keep alive header
        sprintf(txnHandle->htmlHeader,HEADER,htmlPageLen);
        size_t headerLen = strlen(txnHandle->htmlHeader);
        if(headerLen >= 256)
        {
            ERRORMSG("WARNING: HTML
HEADER IS >= 256!, header size:"<<headerLen<<endl);
        }
        //write response to user

lpECB->ServerSupportFunction(lpECB->ConnID,HSE_REQ_SEND_RESPO
NSE_HEADER,"200 OK",NULL,(DWORD*)txnHandle->htmlHeader);

lpECB->WriteClient(lpECB->ConnID,txnHandle->htmlPage,(LPDWORD)&ht
mlPageLen,0);
        DEBUGMSG("HTML
PAGE-->"<<endl<<txnHandle->htmlHeader<<txnHandle->htmlPage<<endl);
    }
    catch (...)
    {
        char response[256];
        ZeroMemory(response,256);
        char *ptr = response;
        appendText(&ptr,"<HTML><BODY> Error : Unhandled
Exception </BODY></HTML>");
        DWORD cbResponse = sizeof(response)-1 ;
        //write response to user

lpECB->ServerSupportFunction(lpECB->ConnID,HSE_REQ_SEND_RESPO
NSE_HEADER,"200 OK",NULL,(DWORD*)response);

lpECB->WriteClient(lpECB->ConnID,response,&cbResponse,0);
    }
    return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}
extern "C" BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO*
pVer)
{
    // Create the extension version string, and copy string to
HSE_VERSION_INFO structure.
    pVer->dwExtensionVersion =
MAKELONG(HSE_VERSION_MINOR, HSE_VERSION_MAJOR);

    // Copy description string into HSE_VERSION_INFO structure.
strcpy(pVer->lpszExtensionDesc, "TPCC ISAPI Extension");
    // Initialize isapi critical section
InitializeCriticalSection(&isapiLock);
    // Initialize error log critical section
InitializeCriticalSection(&errorLock);
    // Initialize terminal critical section
InitializeCriticalSection(&termLock);
    // Initialize debug/error critical sections
if(debugFlag)
        InitializeCriticalSection(&debugMutex);
InitializeCriticalSection(&errorMutex);
    // Read registry values

        if(readRegistryValues() != OK)
            return(FALSE);
        // Initialize terminal array
termArray = (TERM_ENTRY*)
calloc(numUsers,sizeof(TERM_ENTRY));
termNextFree = 1;
        //open up error/debug streams
errorStream.rdbuf( )->open(errorLogFile,ios::out);
if(debugFlag)
        debugStream.rdbuf( )->open(htmlTraceLogFile,ios::out);
ERRORMSG("Error log file open."<<endl);
DEBUGMSG("Loading library for dlvy txn."<<endl);
int rc = getDBInstance();
if (rc != OK)
    {
        ERRORMSG("Error, unable to load database dll,
rc:"<<rc);
        DEBUGMSG("Error, unable to load database dll,
rc:"<<rc);
        return FALSE;
    }
DEBUGMSG("Library loaded for dlvy txn."<<endl);
DEBUGMSG("Calling initDlvy." <<endl);

if(initDlvy() != OK)
    return (FALSE);

DEBUGMSG("Initializing TLS." << endl);

// Initialize thread local storage index
threadLSIndex = TlsAlloc();
if (threadLSIndex == TLS_NULL)
    {
        ERRORMSG("Isapi error: unable to initialize thread
local storage(TLS), rc:" << GetLastError())<<endl);
        return(FALSE);
    }
    DEBUGMSG("sizeof out_neword_struct: "<<sizeof(struct
out_neword_struct)<<endl);
    DEBUGMSG("sizeof in_neword_struct: "<<sizeof(struct
in_neword_struct)<<endl);
    DEBUGMSG("sizeof out_payment_struct: "<<sizeof(struct
out_payment_struct)<<endl);
    DEBUGMSG("sizeof in_payment_struct: "<<sizeof(struct
in_payment_struct)<<endl);
    DEBUGMSG("sizeof out_ordstat_struct: "<<sizeof(struct
out_ordstat_struct)<<endl);
    DEBUGMSG("sizeof in_ordstat_struct: "<<sizeof(struct
in_ordstat_struct)<<endl);
    DEBUGMSG("sizeof out_delivery_struct: "<<sizeof(struct
out_delivery_struct)<<endl);
    DEBUGMSG("sizeof in_delivery_struct: "<<sizeof(struct
in_delivery_struct)<<endl);
    DEBUGMSG("sizeof out_stocklev_struct: "<<sizeof(struct
out_stocklev_struct)<<endl);
    DEBUGMSG("sizeof in_stocklev_struct: "<<sizeof(struct
in_stocklev_struct)<<endl);
    //compute the max struct size for com data construct
maxDataSize = max(maxDataSize,sizeof(nord_wrapper));
maxDataSize = max(maxDataSize,sizeof(paym_wrapper));
maxDataSize = max(maxDataSize,sizeof(ords_wrapper));
maxDataSize = max(maxDataSize,sizeof(dlvy_wrapper));
maxDataSize = max(maxDataSize,sizeof(stok_wrapper));
maxDataSize += 10;
    DEBUGMSG("max data struct size:"<<maxDataSize <<endl);

    return true;
}

```

```

extern "C" BOOL WINAPI TerminateExtension(DWORD dwFlags)
{
    return true;
}

/*
*****
** Name          :          initTxnHandle
** Description   :
**              :          Isapi thread initializes
its own com interface
**              :          structure.
** Parameters    :
**              :          TXN_HANDLE**
isapi txn handle
** Returns      :
**              :          int - return code
** Comments     :
**
*****
*/
int initTxnHandle(TXN_HANDLE **txnHandle)
{
    DEBUGMSG("Inside init txn handle, getting isapiLock." << endl);
    EnterCriticalSection(&isapiLock);

    HRESULT hres = NULL;
    try
    {
        DEBUGMSG("Got ispaiLock, initializing txnHandle:
        "<<DEBUGADDRESS(*txnHandle)<< endl);
        *txnHandle = (TXN_HANDLE *)
        calloc(1,sizeof(TXN_HANDLE));
        if (*txnHandle == NULL)
        {
            ERRORMSG("Unable to allocated
            TXN_HANDLE, rc:"<<GetLastError()<<endl);
            return ERR;
        };
        (*txnHandle)->comInterface.comHandle = NULL;
        DEBUGMSG("Initializing txnHandle com data buffer to
        "<<maxDataSize<<"bytes"<<endl);
        (*txnHandle)->comInterface.txnBuffer = (char *)
        CoTaskMemAlloc(maxDataSize);
        if (!((*txnHandle)->comInterface.txnBuffer))
        {
            ERRORMSG("CoTaskMemAlloc() failed of
            size "<<maxDataSize<<"; rc: "<<hres<<endl);
            return(ERR);
        };
        DEBUGMSG("txnHandle com data buffer initialized to "
        << maxDataSize << "bytes" <<endl);
        DEBUGMSG("Calling CoInitialize with txnHandle:
        "<<DEBUGADDRESS(*txnHandle)<<endl);
        hres =
        CoInitializeEx(NULL,COINIT_MULTITHREADED);
        if (FAILED(hres))
        {
            ERRORMSG("CoInitializeEx() failed, rc :
            "<<hres<<endl);
            return(ERR);
        };

        struct _timeb
        startTime;
        struct _timeb
        endTime;
    }
}

```

```

        DEBUGMSG("Calling CoCreateInstance with
        txnHandle:"<<DEBUGADDRESS(*txnHandle)<< endl);
        _ftime(&startTime);
        hres =
        CoCreateInstance(CLSID_tpcc_com,NULL,CLSCTX_SERVER,IID_Itpcc_co
        m,(void **)&(*txnHandle)->comInterface.comHandle);
        if (FAILED(hres))
        {
            _ftime(&endTime);
            //store error code in txnHandle
            ERRORMSG("CoCreateInstance() failed,
            code:"<<HRESULT_CODE(hres)<<"
            facility:"<<HRESULT_FACILITY(hres)<<
            " hres:"<<hres<< " time
            waiting:"<<
            (((endTime.time -
            startTime.time)*1000)+
            (endTime.millitm -
            startTime.millitm))/1000.0)<<endl);

            DEBUGMSG("CoCreateInstance() failed,
            code:"<<HRESULT_CODE(hres)<<"
            facility:"<<HRESULT_FACILITY(hres)<<
            " hres:"<<hres<< " time
            waiting:"<<
            (((endTime.time -
            startTime.time)*1000)+
            (endTime.millitm -
            startTime.millitm))/1000.0)<<endl);

            return(ERR);
        };
        _ftime(&endTime);
        DEBUGMSG("CoCreateInstance successful.txnHandle
        com initialized, time waiting for object to be activated:" <<
        (((endTime.time - startTime.time)*1000)+
        (endTime.millitm -
        startTime.millitm))/1000.0)<<endl);

        //call set complete to return object to pool.
        (*txnHandle)->comInterface.comHandle->doSetComplete();
        //set the com buffers size
        DEBUGMSG("Setting txnHandle: " <<
        DEBUGADDRESS(*txnHandle) << "com buffer size to " << maxDataSize<<
        endl)
        (*txnHandle)->comInterface.size = maxDataSize;
        DEBUGMSG("txnHandle:
        "<<DEBUGADDRESS(*txnHandle) <<"set to " << maxDataSize << endl);
        TlsSetValue(threadLSIndex,*txnHandle);
        DEBUGMSG("txnHandle:
        "<<DEBUGADDRESS(*txnHandle) << "stored in TLS" << endl);

        ZeroMemory((*txnHandle)->htmlPage,MAX_HTML_PAGE_LEN);
        ZeroMemory((*txnHandle)->htmlHeader,MAX_HTML_HEADER_LEN);

        LeaveCriticalSection(&isapiLock);
        return(OK);
    }
    catch(...)
    {
        DEBUGMSG("Unhandled exeception in initTxnHandle,
        unlocking isapi lock" <<endl);
        ERRORMSG("Unhandled exeception in initTxnHandle,
        unlocking isapi lock" <<endl);
        LeaveCriticalSection(&isapiLock);
    }
}

```

```

    };
    return ERR;
}
/*
*****
** Name          :          getDBInstance
** Description   :
**              load db specific lib
based on dbType registry
**              value.
** Parameters    :
**
** Returns      :
**              int - return code
** Comments     :
**              This function only
exists for the dlvy threads
**              Dlvy threads hold
direct connections to the database
**              and therefore need to
know what db interface to talk to.
*****
*/
int getDBInstance()
{
    if(nullDB)
    {
        dbInstance =
LoadLibrary("c:\\inetpub\\wwwroot\\tpcc\\nullDB.dll");
        if(dbInstance == NULL)
        {
            return ERR_NULL_DLL_NOT_LOADED;
        }
    }
    else if( (strcmp(dbType,"DB2") == 0) )
    {
        dbInstance =
LoadLibrary("c:\\inetpub\\wwwroot\\tpcc\\tpcDB2glue.dll");
        if(dbInstance == NULL)
        {
            return ERR_DB2_DLL_NOT_LOADED;
        }
    }
    else if( (strcmp(dbType,"ORACLE") == 0) )
    {
        return ERR_ORACLE_DLL_NOT_LOADED;
    }
    else
    {
        return ERR_UNKNOWN_DB;
    }
    db_connect =
(CONNECT_PTR)GetProcAddress(dbInstance,"connect_db");
    if(db_connect == NULL)
    {
        return ERR_CONNECT_ADDRESS_NOT_FOUND;
    }
    dlvyCall =
(DLVY_FUNC_PTR)GetProcAddress(dbInstance,"do_dlvy");
    if(dlvyCall == NULL)
    {
        return ERR_DLVY_ADDRESS_NOT_FOUND;
    }
    return OK;
}
/*

```

```

*****
** Name          :          initDlvy
** Description   :
**              initialize dlvy
threads/dlvy queueu
** Parameters    :
**
** Returns      :
**              int - return code
** Comments     :
*****
*/
int initDlvy()
{
    // Initialize critical section
    InitializeCriticalSection(&dlvyQueueLock);
    //create dlvy queue
    dlvyQueue = (DLVYQUEUEDATA *)
calloc(dlvyQueueLen,sizeof(DLVYQUEUEDATA));
    dlvyThreadDone = CreateEvent(NULL,
TRUE,          //manual reset
FALSE,        //initially not signalled.
NULL);
    if(dlvyThreadDone == NULL)
    {
        DEBUGMSG("Error: dlvyThreadDone handled init
failed, GetLastError:<<GetLastError()<<endl);
        ERRORMSG("Error : dlvyThreadDone handled init
failed, GetLastError:<<GetLastError()<<endl);
        return ERR_DLVY_EVENT_INIT_FAILED;
    }
    //create dlvy semaphore
    dlvyThreadSemaphore =
CreateSemaphore(NULL,0,dlvyQueueLen,NULL);
    if(dlvyThreadSemaphore == NULL)
    {
        DEBUGMSG("Error: dlvyThreadSemaphore semaphore
init failed, GetLastError:<<GetLastError()<<endl);
        ERRORMSG("Error: dlvyThreadSemaphore semaphore
init failed, GetLastError:<<GetLastError()<<endl);
        return ERR_DLVY_SEMAPHORE_INIT_FAILED;
    }
    //set number of free slots available in queue
    dlvyBufferFreeSlots = dlvyQueueLen;

    //index into next available slot in dlvy txn queue
    dlvyBufferSlotIndex = 0;

    //thread index into dlvy txn queue
    dlvyBufferThreadIndex = 0;
    dlvyThreadHandles = new HANDLE[dlvyThreads];
    //create threads
    for(int threadCount = 0;threadCount < dlvyThreads;threadCount++)
    {
        dlvyThreadHandles[threadCount] =
(HANDLE)_beginthread(dlvyThreadEntry,0,NULL);
        if(dlvyThreadHandles[threadCount] ==
INVALID_HANDLE_VALUE)
            return ERR_DLVY_THREAD_FAILED;
    }
    return OK;
}

```

```

/*
*****
** Name           :           readRegistryValues
** Description    :
**               :           initialize isapi global
variables from registry
** Parameters    :
**               :
** Returns       :
**               :           int - return code
** Comments      :
*****
*/
int readRegistryValues()
{
    HKEY    registryKey;
    char    value[MAX_STRING_LEN];
    DWORD   regType;
    DWORD   regValue;
    DWORD   regValueSize = MAX_STRING_LEN;

    //open up registry key

    if(RegOpenKeyEx(HKEY_LOCAL_MACHINE,REGISTRY_SUB_KEY,0,K
EY_READ,&registryKey) != ERROR_SUCCESS)
        return ERR_UNABLE_TO_OPEN_REG;

    //get null db flag
    regValueSize = sizeof(regValue);
    if(RegQueryValueEx(registryKey,NULL_DB,0,&regType,(BYTE
*)&regValue,&regValueSize) == ERROR_SUCCESS)
        nullDB = regValue;
    else
        nullDB = 0;

    //get num dlvy threads
    regValueSize = sizeof(regValue);

    if(RegQueryValueEx(registryKey,DELIVERY_THREADS,0,&regType,(BYT
E *)&regValue,&regValueSize) == ERROR_SUCCESS)
        dlvyThreads = regValue;
    else
        dlvyThreads =
DEFAULT_DLVY_THREADS;
    //get dlvy queue len
    regValueSize = sizeof(regValue);

    if(RegQueryValueEx(registryKey,DELIVERY_QUEUE_LEN,0,&regType,(B
YTE *)&regValue,&regValueSize) == ERROR_SUCCESS)
        dlvyQueueLen = regValue;
    else
        dlvyQueueLen =
DEFAULT_DLVY_QUEUE_LEN;
    //get the htmlTrace flag
    regValueSize = sizeof(regValue);

    if(RegQueryValueEx(registryKey,HTML_TRACE,0,&regType,(BYTE
*)&regValue,&regValueSize) == ERROR_SUCCESS)
        trace = regValue;
    else
        trace = 0;

    //get the client null db flag
    regValueSize = sizeof(regValue);
    if(RegQueryValueEx(registryKey,NULL_DB,0,&regType,(BYTE
*)&regValue,&regValueSize) == ERROR_SUCCESS)
        nullDB = regValue;
    else
        nullDB = 0;
}

```

```

//get the num of users
regValueSize = sizeof(regValue);

if(RegQueryValueEx(registryKey,NUM_USERS,0,&regType,(BYTE
*)&regValue,&regValueSize) == ERROR_SUCCESS)
    numUsers = regValue;
else
    numUsers = DEFAULT_NUM_USERS;

//get dlvy log file path
regValueSize = sizeof(value);
if
(RegQueryValueEx(registryKey,DELIVERY_LOG_PATH,0,&regType,(BYT
E *)&value,&regValueSize)== ERROR_SUCCESS )
    strcpy(dlvyLogPath,value);
else
    strcpy(dlvyLogPath,DEFAULT_DLVY_LOG_PATH);

//get global error log file path/name
regValueSize = sizeof(value);
if
(RegQueryValueEx(registryKey,ERROR_LOG_FILE,0,&regType,(BYTE *)
&value,&regValueSize)== ERROR_SUCCESS )
    strcpy(errorLogFile,value);
else
    strcpy(errorLogFile,DEFAULT_ERROR_LOG_FILE);

//get global error log file path/name
regValueSize = sizeof(value);
if
(RegQueryValueEx(registryKey,HTML_TRACE_LOG_FILE,0,&regType,(B
YTE *)&value,&regValueSize)== ERROR_SUCCESS )
    strcpy(htmlTraceLogFile,value);
else
    strcpy(htmlTraceLogFile,DEFAULT_HTML_TRACE_LOG_FILE);

//get db name
regValueSize = sizeof(value);
if (RegQueryValueEx(registryKey,DB_NAME,0,&regType,(BYTE
*)&value,&regValueSize)== ERROR_SUCCESS )
    strcpy(dbName,value);
else
    strcpy(dbName,DEFAULT_DB_NAME);

//get db type
regValueSize = sizeof(value);
if (RegQueryValueEx(registryKey,DB_TYPE,0,&regType,(BYTE
*)&value,&regValueSize)== ERROR_SUCCESS )
    strcpy(dbType,value);
RegCloseKey(registryKey);

return OK;
}
/*
*****
** Name           :           doLoginForm
** Description    :
**               :           HTML Login page entry point
** Parameters    :
**               :           htmlPhraser*      command
block
**               :           TXN_HANDLE*      txn handle
struct
** Returns       :
**               :           int - return code
** Comments      :
*****
*/
int doLoginForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    DEBUGMSG("Entering doLoginForm()."<<endl);
}

```

```

char *html=txnHandle->htmlPage;
DEBUGMSG("Creating html login page"<<endl);
//begin html page
appendText(&html,"<HTML><HEAD><TITLE>TPC-C Client
Home Page</TITLE></HEAD>"
ACTION="\ ""
METHOD="\GET">"
Login.</H2>"
TYPE="\hidden" NAME="\ ""
CMD_TXN_ID
"\ VALUE="\ ""
CMD_MENU
"\ ">"
"<H3>Warehouse
<INPUT NAME="\ ""
NAME="\ ""
CMD_W_ID
"\ SIZE=6>"
" District <INPUT
CMD_D_ID
"\ SIZE=2></H3>"
"<INPUT
TYPE="\submit" VALUE="\Submit">"
"</FORM>");

html+=sprintf(html,"dlvy Queue Length:%d <BR> num dlvy threads:%d <BR>
dlvy queue free slots:%d <BR> isapi queue index:%d <BR> thread queue
index:%d <BR> </BODY></HTML>\n",
dlvyQueueLen,
dlvyThreads,
dlvyBufferFreeSlots,
dlvyBufferSlotIndex,
dlvyBufferThreadIndex);
DEBUGMSG("Html login page done"<<endl);
return OK;
}
/*
*****
** Name : doLoginResults
** Description : HTML Login results page entry
point
** Parameters : htmlPhraser* command
block
** TXN_HANDLE* txn handle
struct
** Returns : int - return code
** Comments :
**
*****
*/
int doLoginResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
char *html=txnHandle->htmlPage;

//validate parameters
if( (txnHandle->w_id = atoi(commandBlock->get_W_ID())) == 0 )
{
doLoginErrorPage(html,ERR_INVALID_W_ID);
return OK;
}
}

```

```

if( (txnHandle->d_id = atoi(commandBlock->get_D_ID())) == 0 )
{
doLoginErrorPage(html,ERR_INVALID_D_ID);
return OK;
}
//store user into terminal array,
//function will ERR if the terminal array is full
if( assignTerminal(txnHandle) != OK)
{
doLoginErrorPage(html,ERR_TERMINAL_FULL);
return OK;
};
appendText(&html,"<HTML><HEAD><TITLE>TPC-C Main
Menu</TITLE></HEAD>\r\n"
ACTION="\ ""
APP_NAME
"\ ""
METHOD="\GET">"\r\n"
"<H3>Please Select
Transaction.</H3>\r\n");
html+=appendButtons(html);
html+=appendHiddenFields(html,txnHandle);
appendText(&html,"</FORM></BODY></HTML>");
return OK;
}
/*
*****
** Name : doLoginErrorPage
** Description : HTML Login page entry point
** Parameters : char * html page
buffer
** char * error
message
** Returns : int - return code
** Comments :
**
*****
*/
int doLoginErrorPage(char *htmlPage,char *errorMessage)
{
char *html=htmlPage;
//begin html page
appendText(&html,"<HTML><HEAD><TITLE>TPC-C Client
Home Page</TITLE></HEAD>"
ACTION="\ ""
APP_NAME
"\ ""
METHOD="\GET">");
appendText(&html,"<H2>Please Login.</H2>"
"<INPUT
TYPE="\hidden" NAME="\ ""
CMD_TXN_ID
"\ VALUE="\ ""
CMD_MENU
"\ ">"
"<H3>Warehouse
<INPUT NAME="\ ""
NAME="\ ""
CMD_W_ID
"\ SIZE=6>"
" District <INPUT
CMD_D_ID
"\ SIZE=2></H3>"

```



```

" <INPUT
TYPE="submit" VALUE="Submit">"
" </FORM>";
appendText(&html_errorMessage);
appendText(&html,"<BODY></HTML>");
return OK;
}
/*
*****
** Name : doNewOrderForm
** Description :
** HTML neworder page entry point
** Parameters :
** htmlPhraser* command
block
** TXN_HANDLE* txn handle
struct
** Returns :
** int - return code
** Comments :
**
*****
*/
int doNewOrderForm(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
char *html=txnHandle->htmlPage;
appendText(&html,"<HTML><HEAD><TITLE>TPC-C New
Order</TITLE></HEAD>\r\n"
" <BODY><FORM
ACTION=""
APP_NAME
""
METHOD="GET">\r\n"
" <CENTER><H3>Please Fill In New Order Form.</H3></CENTER>\r\n"
//check if not needed
"Submit Transaction
CMD_TXN_ID
"" VALUE=""
CMD_NORD
"">");
//append the hidden
html+=appendHiddenFields(html,txnHandle);
//int buffer for warehouse
char buffer[15];
appendText(&html," <PRE>"
" 1 2 3
4 5 6 7 8 9\r\n"
"123456789012345678901234567890123456789012345678901234567890123
456789012345678901234567890\r\n"
"Warehouse: ");
appendText(&html,itoa(txnHandle->w_id,buffer,10),7,1);
appendText(&html,"District: <INPUT NAME=""
CMD_D_ID
"" SIZE=1>
Date:<BR>"
"Customer <INPUT NAME=""
CMD_C_ID
"" SIZE=6> Name:
Credit: %Disc.:<BR>"
"Order Number:
Number of Lines: W_tax: D_tax:<BR><BR>"

```

```

//
4 5 6 7 8 9\r\n"
//
"123456789012345678901234567890123456789012345678901234567890123
456789012345678901234567890\r\n"
" Supp_W Item_Num
Item_Name Qty Stock B/G Price Amount <BR> ");
//append the 15 items commands
html+=appendItems(html,NORD_ITEMS,ITEM_START);
//seal up html page
appendText(&html,"</PRE></BODY></HTML>");
return OK;
}
/*
*****
** Name : doNewOrderResults
** Description :
** HTML neworder page entry point
** Parameters :
** htmlPhraser* command
block
** TXN_HANDLE* txn handle
struct
** Returns :
** int - return code
** Comments :
**
*****
*/
int doNewOrderResults(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
DEBUGMSG("Entered doNewOrderResults" << endl);
char *html=txnHandle->htmlPage;
struct nord_wrapper *nord = NULL;
DEBUGMSG("Casting COM txnBuffer to nord struct" <<endl);
nord = (nord_wrapper*)txnHandle->comInterface.txnBuffer;
ZeroMemory(nord,maxDataSize);
DEBUGMSG("COM txnBuffer initialized, validating input
parameters" << endl);
//set warehouse,district and customer id from command block
nord->in_nord.s_W_ID = txnHandle->w_id;
DEBUGMSG("nord w_id:" << nord->in_nord.s_W_ID << endl);
if( (nord->in_nord.s_D_ID = atoi(commandBlock->get_D_ID())
== 0)
{
doNewOrderErrorPage(html,ERR_INVALID_D_ID,commandBlock,txnHandle
);
return OK;
}
DEBUGMSG("nord d_id:" << nord->in_nord.s_D_ID << endl);
if((nord->in_nord.s_C_ID = atoi(commandBlock->get_C_ID())) ==
0)
{
doNewOrderErrorPage(html,ERR_INVALID_C_ID,commandBlock,txnHandle
);
return OK;
}
DEBUGMSG("nord c_id:" << nord->in_nord.s_C_ID << endl);
int itemCmd = ITEM_START;
short itemComplete = 0;
char field[256] = {NULL};
for (int itemIndex=0;itemIndex<NORD_ITEMS;itemIndex++)

```

```

{
    //supply warehouse
    if( *(commandBlock->get_ITEM_SUPP_W(itemIndex))
)
        if(
(nord->in_nord.in_item[nord->in_nord.s_O_OL_CNT].s_OL_SUPPLY_W_ID
= atoi(commandBlock->get_ITEM_SUPP_W(itemIndex))) == 0)
        {
doNewOrderErrorPage(html,ERR_INVALID_SUPPLY_W_ID,commandBlock
,txnHandle);
            return OK;
        }
        else
            itemComplete++;
        //item number
        if(
*(commandBlock->get_ITEM_ITEM_NUM(itemIndex))
        {
            if(itemComplete==1)
            {
                if(
(nord->in_nord.in_item[nord->in_nord.s_O_OL_CNT].s_OL_I_ID =
atoi(commandBlock->get_ITEM_ITEM_NUM(itemIndex))) == 0)
                {
doNewOrderErrorPage(html,ERR_INVALID_ITEM_NUM,commandBlock,txn
Handle);
                    return OK;
                }
                else
                    itemComplete++;
            }
            //missing previous value of item supp
warehouse, flag error
        }
        else
        {
doNewOrderErrorPage(html,ERR_INVALID_SUPPLY_W_ID,commandBlock
,txnHandle);
            return OK;
        }
        }
        else if( (itemComplete==1) //nothing in the command
block, check to see if the previous item value is present
        {
doNewOrderErrorPage(html,ERR_INVALID_ITEM_NUM,commandBlock,txn
Handle);
            return OK;
        }
        //item qty
        if( *(commandBlock->get_ITEM_QTY(itemIndex))
        {
            if(itemComplete==2)
            {
                if(
(nord->in_nord.in_item[nord->in_nord.s_O_OL_CNT].s_OL_QUANTITY =
atoi(commandBlock->get_ITEM_QTY(itemIndex))) == 0)
                {
doNewOrderErrorPage(html,ERR_INVALID_ITEM_OTY,commandBlock,txn
Handle);
                    return OK;
                }
                else
                    itemComplete++;
            }
        }
    }
}
//missing previous value of item number
else if (itemComplete ==1)
{
doNewOrderErrorPage(html,ERR_INVALID_ITEM_NUM,commandBlock,txn
Handle);
    return OK;
}
//missing 1st value of supp warehouse
else
{
doNewOrderErrorPage(html,ERR_INVALID_SUPPLY_W_ID,commandBlock
,txnHandle);
    return OK;
}
else if(itemComplete==2) //nothing in
the command block, check to see if the previous item values are present
{
doNewOrderErrorPage(html,ERR_INVALID_ITEM_NUM,commandBlock,txn
Handle);
    return OK;
}
DEBUGMSG("nord item:" <<
nord->in_nord.s_O_OL_CNT << "SUPPLY_W_ID:" <<
nord->in_nord.in_item[nord->in_nord.s_O_OL_CNT].s_OL_SUPPLY_W_ID
<<
" OL_I_ID:" <<
nord->in_nord.in_item[nord->in_nord.s_O_OL_CNT].s_OL_I_ID << "
OL_QUANTITY:" <<
nord->in_nord.in_item[nord->in_nord.s_O_OL_CNT].s_OL_QUANTITY
<<endl);
    if(itemComplete == 3)
        nord->in_nord.s_O_OL_CNT++;
    itemComplete=0;
}
DEBUGMSG("complete nord
items:"<<nord->in_nord.s_O_OL_CNT<<" initializing remaina unused items "
<< NORD_ITEMS - nord->in_nord.s_O_OL_CNT << " to 0" <<endl);
    for(int
itemIndex=nord->in_nord.s_O_OL_CNT;itemIndex<NORD_ITEMS;itemIndex
++)
    {
nord->in_nord.in_item[itemIndex].s_OL_SUPPLY_W_ID=0;
nord->in_nord.in_item[itemIndex].s_OL_I_ID = 0;
nord->in_nord.in_item[itemIndex].s_OL_QUANTITY
=0;
    }
    DEBUGMSG("nord creating new order results html title page"
<<endl);
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C New Order
Results</TITLE></HEAD>\r\n"
" <BODY><FORM
ACTION=""
APP_NAME
""
METHOD=""GET"">\r\n");
    //append menu buttons
html+=appendButtons(html);
html+=appendHiddenFields(html,txnHandle);
appendText(&html,"</FORM><CENTER><H3>New Order</H3>
<BR></CENTER>"
" <PRE>"

```

```

//      "      1      2      3
4      5      6      7      8      9\r\n"
//
"1234567890123456789012345678901234567890123456789012345678901234567890\r\n
456789012345678901234567890\r\n
";

//assume failure
nord->out_nord.s_transtatus = -1;
DEBUGMSG("nord executing COM interface function" << endl);
HRESULThres;
try
{
    hres =
txnHandle->comInterface.comHandle->doNewOrder(&txnHandle->comInterfa
ce.size,(UCHAR*)&txnHandle->comInterface.txnBuffer);
}
catch(...)
{
    html+=sprintf(html,"ERROR: nord com call caused
exeception to occur.</PRE></BODY></HTML>");
    ERRORMSG("ERROR : nord com call cause exeception
to occur,"<<endl);
    return OK;
}
if(FAILED(hres))
{
    ERRORMSG("ERROR : nord com call failed, rc:" <<
hex << hres);
    DEBUGMSG("ERROR : nord com call failed, rc:" <<
hex << hres);
    return OK;
}

//com call successful, return object back to pool.
hres = txnHandle->comInterface.comHandle->doSetComplete();
if(FAILED(hres))
{
    ERRORMSG("ERROR : nord setcomplete call failed,
rc:" << hex << hres);
    DEBUGMSG("ERROR : nord setcomplete call failed,
rc:" << hex << hres);
}
nord = (nord_wrapper *)txnHandle->comInterface.txnBuffer;
if(FAILED(hres))
{
    html+=sprintf(html,"ERROR: nord com doSetComplete
failed, rc:%ld</PRE></BODY></HTML>",hres);
    ERRORMSG("ERROR : nord com doSetComplete
failed, rc:"<<DEBUGADDRESS(hres)<<endl);
    return OK;
}
DEBUGMSG("nord COM interface function successful,
s_transtatus:" << nord->out_nord.s_transtatus << endl);
int rc = nord->out_nord.s_transtatus;
char buffer[10];
appendText(&html,"Warehouse: ");
appendText(&html,itoa(nord->in_nord.s_W_ID,buffer,10),6,1);
appendText(&html,"District: ");
appendText(&html,itoa(nord->in_nord.s_D_ID,buffer,10),26,1);
appendText(&html,"Date: ");
if(rc == OK)
{
    char dateTmeBuffer[50];

copyOutDateTime(dateTmeBuffer,nord->out_nord.s_O_ENTRY_D_time);
    appendText(&html,dateTmeBuffer);
}
appendText(&html," <BR>"
"Customer: ");
appendText(&html,itoa(nord->in_nord.s_C_ID,buffer,10),8,1);
appendText(&html,"Name: ");

appendText(&html,nord->out_nord.s_C_LAST,LAST_NAME_LEN+3,1);
appendText(&html,"Credit: ");
appendText(&html,nord->out_nord.s_C_CREDIT,5,1);

appendText(&html,"%Disc.: ");
if(rc == OK)
{
    html+=sprintf(html,"%2.2lf",nord->out_nord.s_C_DISCOUNT/100.0);
}
appendText(&html," <BR>"
"Order Number: ");
if(rc != INVALID_STATUS)

appendText(&html,itoa(nord->out_nord.s_O_ID,buffer,10),10,1);

appendText(&html,"Number of Lines: ");

if(rc != INVALID_STATUS)

appendText(&html,itoa(nord->out_nord.s_O_OL_CNT,buffer,10),10,1);
appendText(&html,"W_Tax: ");
if(rc == OK)
{
    html+=sprintf(html,"%5.2lf",nord->out_nord.s_W_TAX/100.0);
}
appendText(&html," D_Tax: ");
if(rc == OK)
{
    html+=sprintf(html,"%5.2lf",nord->out_nord.s_D_TAX/100.0);
}
appendText(&html," <BR> <BR>"
//      "      1      2      3      4      5      6
7      8      9\r\n"
//
"12345678901234567890123456789012345678901234567890123456789012345678901234567890\r\n
456789012345678901234567890\r\n"
" Supp_W Item_Id Item_Name
Qty Stock B/G Price Amount <BR>");

//display items
if (rc == OK)
{
    //display valid items
    for(int itemCount=0;itemCount <
nord->out_nord.s_O_OL_CNT;itemCount++)
    {
        appendText(&html,itoa(nord->in_nord.in_item[itemCount].s_OL_SUPPLY_W
_ID,buffer,10),8,1);

appendText(&html,itoa(nord->in_nord.in_item[itemCount].s_OL_I_ID,buffer,1
0),10,1);

appendText(&html,nord->out_nord.item[itemCount].s_I_NAME,DEFAULT_S
TRING_LEN+1,1);

appendText(&html,itoa(nord->in_nord.in_item[itemCount].s_OL_QUANTITY,
buffer,10),5,1);

```

```
appendText(&html,itoa(nord->out_nord.item[itemCount].s_S_QUANTITY,buff
fer,10),7,1);
        html+=sprintf(html,"%c  $%-7.2lf  $%-7.2lf
<BR> ",nord->out_nord.item[itemCount].s_brand_generic,
nord->out_nord.item[itemCount].s_I_PRICE/100.0,
nord->out_nord.item[itemCount].s_OL_AMOUNT/100.0);
    }
    //display blank line for remaining empty items in the
order
        for(int lineBreaks=0;lineBreaks <
(NORD_ITEMS-nord->out_nord.s_O_OL_CNT);lineBreaks++)
            appendText(&html,"<BR>");
    }
    else
        appendText(&html,"<BR><BR><BR><BR><BR><BR><BR><BR>");
<BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR>");
        appendText(&html,"r\n <BR> ");
        html+=displayStatus(html,rc);
        if(rc == OK)
            html+=sprintf(html," Total:
$%.2lf",nord->out_nord.s_total_amount/100.0);
        else
            appendText(&html," Total: <BR>");

        appendText(&html,"</PRE></BODY></HTML>");
        DEBUGMSG("nord html page complete. returning to calling
function" << endl);
        return OK;
    }
}
/*
*****
** Name : doNewOrderErrorPage
** Description :
** HTML neworder page entry point
** Parameters :
** char * html result
page
** char * error
message
** htmlPhraser* command block
** TXN_HANDLE* txn handle
struct
** Returns :
** int - return code
** Comments :
*****
*/
int doNewOrderErrorPage(char *htmlPage,char *message,htmlPhraser
*commandBlock,TXN_HANDLE *txnHandle)
{
    char *html=htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C New
Order</TITLE></HEAD>r\n"
        "<BODY><FORM
ACTION=\\""
        APP_NAME
        "\"
METHOD=\\"GET\\"">r\n"
"<CENTER><H3>Please Fill In New Order Form.</H3></CENTER>r\n"
"Submit Transaction
<INPUT TYPE=\\"submit\\" NAME=\\""
        CMD_TXN_ID
        "\" VALUE=\\""
        CMD_NORD
        "\">");
    //append the hidden warehouse and district fields
    html+=appendHiddenFields(html,txnHandle);
    //int buffer for warehouse
    char buffer[15];
    /*appendText(&html,"<PRE>
1 2 3 4 5
6 7 8 9r\n"
"1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890r\n"
        "Warehouse: ");*/
    appendText(&html,"<PRE>Warehouse: ");
    appendText(&html,itoa(txnHandle->w_id,buffer,10),7,1);
    appendText(&html,"District: <INPUT NAME=\\""
        CMD_D_ID
        "\" SIZE=1>
Date:<BR>"
        "Customer <INPUT NAME=\\""
        CMD_C_ID
        "\" SIZE=6> Name:
Credit: %Disc.<BR>"
        "Order Number:
Number of Lines: W_tax: D_tax:<BR><BR>"
        "//" 1 2 3
        4 5 6 7 8 9r\n"
        //"1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890r\n"
        "Supp_W Item_Num
Item_Name Qty Stock B/G Price Amount <BR> ");
    //append the 15 items commands
    html+=appendItems(html,NORD_ITEMS,ITEM_START);
    appendText(&html,message);
    //seal up html page
    appendText(&html,"</PRE></BODY></HTML>");
    return OK;
}
/*
*****
** Name : doPaymentForm
** Description :
** HTML payment page entry point
** Parameters :
** htmlPhraser* command
block
** TXN_HANDLE* txn handle
struct
** Returns :
** int - return code
** Comments :
*****
*/
int doPaymentForm(htmlPhraser *commandBlock,TXN_HANDLE
*txnHandle)
{
    char *html=txnHandle->htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD>r\n"
        "<BODY><FORM
ACTION=\\""
        APP_NAME
        "\"
METHOD=\\"GET\\"">r\n"

```

```

"<CENTER><H3>Please Fill In Payment Form.</H3></CENTER><BR>\r\n"
"Submit Transaction
<INPUT TYPE=\"submit\" NAME=\"\"
        CMD_TXN_ID
        \"\" VALUE=\"\"
        CMD_PYMT
        \"\">);
html+=appendHiddenFields(html,txnHandle);
appendText(&html,\"<BR><PRE>\r\n"
        "Date:<BR>"
        "Warehouse: ");
char buffer[15];
appendText(&html,ittoa(txnHandle->w_id,buffer,10));

appendSpaces(&html,10);
appendText(&html,\"District: <INPUT NAME=\"\"
        CMD_D_ID
        \"\" SIZE=1>\r\n<BR>"
        "<BR><BR><BR>"
        "Customer: "
        "<INPUT NAME=\"\"
        CMD_C_ID
        \"\" SIZE=5>"
        " "
        "Cust-Warehouse: "
        "<INPUT NAME=\"\"
        CMD_C_W_ID
        \"\" SIZE=5>"
        " "
        "Cust-District: "
        "<INPUT NAME=\"\"
        CMD_C_D_ID
        \"\" SIZE=1><BR>"
        "Name: <INPUT
NAME=\"\"
        CMD_C_NAME
        \"\" SIZE=20>");
appendText(&html,"
        Since: <BR>"
        "
        "
        "
        "Amount Paid: "
        "<INPUT NAME=\"\"
        CMD_AMT_PAID
        \"\" SIZE=10>"
        " "
        "New
Cust-Balance:<BR>"
        "Credit Limit:<BR>"
<BR>Cust-Data:<BR> <BR> <BR> <BR> </PRE>");
return OK;
}
/*
*****
** Name          : doPaymentResults
** Description    :
**               : HTML neworder page entry point
** Parameters    :
**               : htmlPhraser*   command
block            : TXN_HANDLE*   txn handle
** Returns      :

```

```

**               : int - return code
** Comments      :
**               :
*****
*/
int doPaymentResults(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
    char *html=txnHandle->htmlPage;
    char buffer[50];
    struct paym_wrapper *pymt = NULL;
    pymt = (paym_wrapper*)txnHandle->comInterface.txnBuffer;
    ZeroMemory(pymt,maxDataSize);

    //set login warehouse id from command block
    pymt->in_paym.s_W_ID = txnHandle->w_id;
    //set district from command block
    if( (pymt->in_paym.s_D_ID = atoi(commandBlock->get_D_ID()))
== 0)
    {
doPaymentErrorPage(html,ERR_INVALID_D_ID,commandBlock,txnHandle);
        return OK;
    }

    //set customer id from command block
    if( (pymt->in_paym.s_C_ID = atoi(commandBlock->get_C_ID()))
== 0)
    {
        if(*(commandBlock->get_C_NAME()) == NULL)
        {
            //no customer id nor customer last name
            specified.
doPaymentErrorPage(html,ERR_MISSING_C_ID_OR_CLAST,commandBlock,
txnHandle);
                return OK;
        }
        else
        {
            strcpy(pymt->in_paym.s_C_LAST,commandBlock->get_C_NAME());
        }
        else
        {
            //make sure that the user only inserted just c_id
            if(*(commandBlock->get_C_NAME()) != NULL)
            {
doPaymentErrorPage(html,ERR_C_ID_OR_CLAST_ONLY,commandBlock,tx
nHandle);
                return OK;
            }
        }
        //get customer warehouse id field
        if( (pymt->in_paym.s_C_W_ID =
atoi(commandBlock->get_C_W_ID())) == 0)
        {
doPaymentErrorPage(html,ERR_INVALID_C_W_ID,commandBlock,txnHand
le);
                return OK;
        }
        //get customer district id field
        if( (pymt->in_paym.s_C_D_ID =
atoi(commandBlock->get_C_D_ID())) == 0)
        {

```

```

doPaymentErrorPage(html,ERR_INVALID_C_D_ID,commandBlock,txnHandle);
    return OK;
}

if(!copyInMoney64(commandBlock->get_AMT_PAID(),&pymt->in_paym.s_H_AMOUNT))
{

doPaymentErrorPage(html,ERR_INVALID_PAYMENT_AMOUNT,commandBlock,txnHandle);
    return OK;
}
appendText(&html,"<HTML><HEAD><TITLE>TPC-C Payment Results</TITLE></HEAD>\r\n"
"        "<BODY><FORM
ACTION=""
APP_NAME
METHOD="">
    html+=appendButtons(html);
    html+=appendHiddenFields(html,txnHandle);

appendText(&html,"</FORM><CENTER><H3>Payment</H3></CENTER>");
;

    DEBUGMSG("Calling com entry api payment,
w_id:"<<pymt->in_paym.s_W_ID<<"
d_id:"<<pymt->in_paym.s_D_ID<<endl);
    //assume failure
    pymt->out_paym.s_transtatus = -1;
    HRESULThres;
    try
    {
        hres =
txnHandle->comInterface.comHandle->doPayment(&txnHandle->comInterface.size,(UCHAR**)&txnHandle->comInterface.txnBuffer);
    }
    catch(...)
    {
        html+=sprintf(html,"ERROR: Com Payment call caused exception to occur.</PRE></BODY></HTML>");
        ERRORMSG("ERROR : Com Payment call caused exception to occur."<<endl);
        return OK;
    }
    if(FAILED(hres))
    {
        html+=sprintf(html,"ERROR: pymt com call failed, rc:%x</PRE></BODY></HTML>",hres);
        ERRORMSG("ERROR : pymt com call failed, rc:"<<hres<<endl);
        return OK;
    }
    hres = txnHandle->comInterface.comHandle->doSetComplete();
    if(FAILED(hres))
    {
        html+=sprintf(html,"ERROR: pymt com doSetComplete failed, rc:%ld</PRE></BODY></HTML>",hres);
        ERRORMSG("ERROR : pymt com doSetComplete failed, rc:"<<DEBUGADDRESS(hres)<<endl);
        return OK;
    }
    pymt = (pymt_wrapper *)txnHandle->comInterface.txnBuffer;
    //get return code
    int rc = pymt->out_paym.s_transtatus;
    if( rc != OK)

```

```

{
    html+=displayStatus(html,rc);
    appendText(&html,"</PRE></BODY></HTML>");
    ERRORMSG("Payment TXN ERROR"<<endl

<<"pymt->in_paym.s_C_D_ID:"<<pymt->in_paym.s_C_D_ID<<endl
<<"pymt->in_paym.s_C_ID:"<<pymt->in_paym.s_C_ID<<endl
<<"pymt->in_paym.s_C_LAST:"<<pymt->in_paym.s_C_LAST<<endl
<<"pymt->in_paym.s_C_W_ID:"<<pymt->in_paym.s_C_W_ID<<endl
<<"pymt->in_paym.s_D_ID:"<<pymt->in_paym.s_D_ID<<endl
<<"pymt->in_paym.s_H_AMOUNT:"<<pymt->in_paym.s_H_AMOUNT<<endl
<<"pymt->in_paym.s_H_DATE_time:"<<pymt->in_paym.s_H_DATE_time<<endl
<<"pymt->in_paym.s_W_ID:"<<pymt->in_paym.s_W_ID<<endl
<<"pymt->out_paym.deadlocks:"<<pymt->out_paym.deadlocks<<endl
<<"pymt->out_paym.s_C_BALANCE:"<<pymt->out_paym.s_C_BALANCE<<endl
<<"pymt->out_paym.s_C_CITY:"<<pymt->out_paym.s_C_CITY<<endl
<<"pymt->out_paym.s_C_CREDIT:"<<pymt->out_paym.s_C_CREDIT<<endl
<<"pymt->out_paym.s_C_CREDIT_LIM:"<<pymt->out_paym.s_C_CREDIT_LIM<<endl
<<"pymt->out_paym.s_C_DATA:"<<pymt->out_paym.s_C_DATA<<endl
<<"pymt->out_paym.s_C_DISCOUNT:"<<pymt->out_paym.s_C_DISCOUNT<<endl
<<"pymt->out_paym.s_C_FIRST:"<<pymt->out_paym.s_C_FIRST<<endl
<<"pymt->out_paym.s_C_ID:"<<pymt->out_paym.s_C_ID<<endl
<<"pymt->out_paym.s_C_LAST:"<<pymt->out_paym.s_C_LAST<<endl
<<"pymt->out_paym.s_C_MIDDLE:"<<pymt->out_paym.s_C_MIDDLE<<endl
<<"pymt->out_paym.s_C_PHONE:"<<pymt->out_paym.s_C_PHONE<<endl
<<"pymt->out_paym.s_C_SINCE_time:"<<pymt->out_paym.s_C_SINCE_time<<endl
<<"pymt->out_paym.s_C_STATE:"<<pymt->out_paym.s_C_STATE<<endl
<<"pymt->out_paym.s_C_STREET_1:"<<pymt->out_paym.s_C_STREET_1<<endl
<<"pymt->out_paym.s_C_STREET_2:"<<pymt->out_paym.s_C_STREET_2<<endl
<<"pymt->out_paym.s_C_ZIP:"<<pymt->out_paym.s_C_ZIP<<endl
<<"pymt->out_paym.s_D_CITY:"<<pymt->out_paym.s_D_CITY<<endl
<<"pymt->out_paym.s_D_STATE:"<<pymt->out_paym.s_D_STATE<<endl

```

```

<<"pymt->out_paym.s_D_STREET_1:"<<pymt->out_paym.s_D_STREET_1<
<endl

<<"pymt->out_paym.s_D_STREET_2:"<<pymt->out_paym.s_D_STREET_2<
<endl

<<"pymt->out_paym.s_D_ZIP:"<<pymt->out_paym.s_D_ZIP<<endl

<<"pymt->out_paym.s_H_DATE_time:"<<pymt->out_paym.s_H_DATE_time
<<endl

<<"pymt->out_paym.s_transtatus:"<<pymt->out_paym.s_transtatus<<endl

<<"pymt->out_paym.s_W_CITY:"<<pymt->out_paym.s_W_CITY<<endl

<<"pymt->out_paym.s_W_STATE:"<<pymt->out_paym.s_W_STATE<<endl

<<"pymt->out_paym.s_W_STREET_1:"<<pymt->out_paym.s_W_STREET_1
<<endl

<<"pymt->out_paym.s_W_STREET_2:"<<pymt->out_paym.s_W_STREET_2
<<endl

<<"pymt->out_paym.s_W_ZIP:"<<pymt->out_paym.s_W_ZIP<<endl);
    return OK;
}
//      appendText(&html, "<BR><PRE>\r\n");
//      appendText(&html, "    1    2    3    4    5    6    7
//      8<BR>");
//
appendText(&html, "12345678901234567890123456789012345678901234567
890123456789012345678901234567890<BR>");
//start creating result body
appendText(&html, "<BR><PRE>\r\n"
           "Date: ");
copyOutDateTime(buffer, pymt->out_paym.s_H_DATE_time);
appendText(&html, buffer);
appendText(&html, "<BR>"
           "Warehouse: ");

appendText(&html, itoa(pymt->in_paym.s_W_ID, buffer, 10), 6+24, 1);
appendText(&html, "District: ");
appendText(&html, itoa(pymt->in_paym.s_D_ID, buffer, 10), 2, 1);
appendText(&html, "<BR>");
//print out warehouse and district information

appendText(&html, pymt->out_paym.s_W_STREET_1, STREET_LEN+21, 1);

appendText(&html, pymt->out_paym.s_D_STREET_1, STREET_LEN, 1);
appendText(&html, "<BR>");

appendText(&html, pymt->out_paym.s_W_STREET_2, STREET_LEN+21, 1);

appendText(&html, pymt->out_paym.s_D_STREET_2, STREET_LEN, 1);
appendText(&html, "<BR>");
appendText(&html, pymt->out_paym.s_W_CITY, CITY_LEN+1, 1);

appendText(&html, pymt->out_paym.s_W_STATE, STATE_LEN+1, 1);
copyOutZip(buffer, pymt->out_paym.s_W_ZIP);
appendText(&html, buffer);
appendText(&html, pymt->out_paym.s_D_CITY, CITY_LEN+1, 1);

appendText(&html, pymt->out_paym.s_D_STATE, STATE_LEN+1, 1);
copyOutZip(buffer, pymt->out_paym.s_D_ZIP);
appendText(&html, buffer);
//print out customer information

```

```

appendText(&html, "<BR> <BR>Customer: ");
appendText(&html, itoa(pymt->out_paym.s_C_ID, buffer, 10), 5+1, 1);
appendText(&html, "Cust-Warehouse: ");

appendText(&html, itoa(pymt->in_paym.s_C_W_ID, buffer, 10), 6+1, 1);
appendText(&html, "Cust-District: ");
appendText(&html, itoa(pymt->in_paym.s_C_D_ID, buffer, 10));
//add customer information
appendText(&html, "<BR>Name: ");

appendText(&html, pymt->out_paym.s_C_FIRST, FIRST_NAME_LEN+1, 1);

appendText(&html, pymt->out_paym.s_C_MIDDLE, INITIALS_LEN+1, 1);
DEBUGMSG("Last name:"<<pymt->out_paym.s_C_LAST<<endl);

appendText(&html, pymt->out_paym.s_C_LAST, LAST_NAME_LEN+5, 1);
appendText(&html, "Since: ");
copyOutDateTime(buffer, pymt->out_paym.s_C_SINCE_time);
appendText(&html, buffer);
appendText(&html, "<BR>");
appendSpaces(&html, 8);

appendText(&html, pymt->out_paym.s_C_STREET_1, STREET_LEN+20, 1);
appendText(&html, "Credit: ");
appendText(&html, pymt->out_paym.s_C_CREDIT);
appendText(&html, "<BR>");
appendSpaces(&html, 8);

appendText(&html, pymt->out_paym.s_C_STREET_2, STREET_LEN+21, 1);
appendText(&html, "%Disc: ");

html+=sprintf(html, "%2.2lf", pymt->out_paym.s_C_DISCOUNT/100.0);

appendText(&html, "<BR>");
appendSpaces(&html, 8);
appendText(&html, pymt->out_paym.s_C_CITY, CITY_LEN+1, 1);

appendText(&html, pymt->out_paym.s_C_STATE, STATE_LEN+1, 1);
copyOutZip(buffer, pymt->out_paym.s_C_ZIP);
appendText(&html, buffer, 15, 1);

appendText(&html, "Phone: ");
copyOutPhone(buffer, pymt->out_paym.s_C_PHONE);
appendText(&html, buffer);
appendText(&html, "<BR> <BR>Amount Paid: $");

html+=sprintf(html, "%-9.2lf", pymt->in_paym.s_H_AMOUNT/100.0);

appendText(&html, "New Cust-Balance: $");

html+=sprintf(html, "%-9.2lf", pymt->out_paym.s_C_BALANCE/100.0);

appendText(&html, "<BR>Credit Limit: $");

html+=sprintf(html, "%-9.2lf", pymt->out_paym.s_C_CREDIT_LIM/100.0);

appendText(&html, "<BR> <BR>Cust-Data: ");
if(pymt->out_paym.s_C_CREDIT[0] == 'B' &&
pymt->out_paym.s_C_CREDIT[1] == 'C')
{
appendCustData(&html, pymt->out_paym.s_C_DATA);
appendText(&html, "<BR>");
}
else
appendText(&html, "<BR> <BR> <BR>");
html+=displayStatus(html, rc);
appendText(&html, "</PRE></BODY></HTML>");

```

```

return OK;
}
/*
*****
** Name          : doPaymentErrorPage
** Description    :
**               append payment error body
** Parameters    :
**               char *          html page
result
**               char *          error
message
**               htmlPhraser * command block
**               TXN_HANDLE*   txn handle
struct
** Returns      :
**               int - return code
** Comments     :
**
*****
*/
int doPaymentErrorPage(char *htmlPage,char *message,htmlPhraser
*commandBlock,TXN_HANDLE *txnHandle)
{
    char *html=htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD>\r\n"
                "<BODY><FORM
ACTION=\"\"
                APP_NAME
                \"\"
METHOD=\"GET\">\r\n"
                "<CENTER><H3>Please Fill In Payment Form.</H3></CENTER> <BR>\r\n"
                "Submit Transaction
                CMD_TXN_ID
                \"\" VALUE=\"\"
                CMD_PYMT
                \"\">);
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"<BR><PRE>\r\n"
                "Date:<BR>"
                "Warehouse: ");
    char buffer[15];
    appendText(&html,itoa(txnHandle->w_id,buffer,10));
    appendSpaces(&html,10);
    appendText(&html,"District: <INPUT NAME=\"\"
                CMD_D_ID
                \"\" SIZE=1>\r\n<BR>"
                "<BR> <BR> <BR>
                \"Customer: "
                "<INPUT NAME=\"\"
                CMD_C_ID
                \"\" SIZE=5>"
                " "
                "Cust-Warehouse: "
                "<INPUT NAME=\"\"
                CMD_C_W_ID
                \"\" SIZE=6>"
                " "
                "Cust-District: "
                "<INPUT NAME=\"\"
                CMD_C_D_ID
                \"\" SIZE=1><BR>"
                "<BR>"
                "Name: <INPUT
                CMD_C_NAME
                \"\" SIZE=20>");
    appendText(&html,"
                Since: <BR>"
                "
                "
                "
                "Amount Paid: "
                "<INPUT NAME=\"\"
                CMD_AMT_PAID
                \"\" SIZE=10>"
                "
                "New
                Cust-Balance:<BR>"
                "Credit Limit:<BR>
                <BR> <BR> Cust-Data:<BR> <BR> <BR> <BR> ");
    appendText(&html,message);
    appendText(&html,"</PRE>");
    return OK;
}
/*
*****
** Name          : doOrderStatusForm
** Description    :
**               HTML orderStatus page entry
point
** Parameters    :
**               htmlPhraser*   command
block
**               TXN_HANDLE*   txn handle
struct
** Returns      :
**               int - return code
** Comments     :
**
*****
*/
int doOrderStatusForm(htmlPhraser *commandBlock,TXN_HANDLE
*txnHandle)
{
    char *html=txnHandle->htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Order
Status</TITLE></HEAD>\r\n"
                "<BODY><FORM
ACTION=\"\"
                APP_NAME
                \"\"
METHOD=\"GET\">\r\n"
                "<CENTER><H3>Please Fill In Order Status Form.</H3></CENTER>
                <BR>\r\n"
                "Submit Transaction
                <INPUT TYPE=\"submit\" NAME=\"\"
                CMD_TXN_ID
                \"\" VALUE=\"\"
                CMD_ORDS
                \"\">"
                "<BR> ");
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"<PRE>\r\n"
                "Warehouse: ");
    char buffer[15];
    appendText(&html,itoa(txnHandle->w_id,buffer,10));
}

```

```

return OK;
}
/*
*****
** Name          : doPaymentErrorPage
** Description    :
**               append payment error body
** Parameters    :
**               char *          html page
result
**               char *          error
message
**               htmlPhraser * command block
**               TXN_HANDLE*   txn handle
struct
** Returns      :
**               int - return code
** Comments     :
**
*****
*/
int doPaymentErrorPage(char *htmlPage,char *message,htmlPhraser
*commandBlock,TXN_HANDLE *txnHandle)
{
    char *html=htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD>\r\n"
                "<BODY><FORM
ACTION=\"\"
                APP_NAME
                \"\"
METHOD=\"GET\">\r\n"
                "<CENTER><H3>Please Fill In Payment Form.</H3></CENTER> <BR>\r\n"
                "Submit Transaction
                CMD_TXN_ID
                \"\" VALUE=\"\"
                CMD_PYMT
                \"\">);
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"<BR><PRE>\r\n"
                "Date:<BR>"
                "Warehouse: ");
    char buffer[15];
    appendText(&html,itoa(txnHandle->w_id,buffer,10));
    appendSpaces(&html,10);
    appendText(&html,"District: <INPUT NAME=\"\"
                CMD_D_ID
                \"\" SIZE=1>\r\n<BR>"
                "<BR> <BR> <BR>
                \"Customer: "
                "<INPUT NAME=\"\"
                CMD_C_ID
                \"\" SIZE=5>"
                " "
                "Cust-Warehouse: "
                "<INPUT NAME=\"\"
                CMD_C_W_ID
                \"\" SIZE=6>"
                " "
                "Cust-District: "
                "<INPUT NAME=\"\"
                CMD_C_D_ID
                \"\" SIZE=1><BR>"
                "<BR>"
                "Name: <INPUT
                CMD_C_NAME
                \"\" SIZE=20>");
    appendText(&html,"
                Since: <BR>"
                "
                "
                "
                "Amount Paid: "
                "<INPUT NAME=\"\"
                CMD_AMT_PAID
                \"\" SIZE=10>"
                "
                "New
                Cust-Balance:<BR>"
                "Credit Limit:<BR>
                <BR> <BR> Cust-Data:<BR> <BR> <BR> <BR> ");
    appendText(&html,message);
    appendText(&html,"</PRE>");
    return OK;
}
/*
*****
** Name          : doOrderStatusForm
** Description    :
**               HTML orderStatus page entry
point
** Parameters    :
**               htmlPhraser*   command
block
**               TXN_HANDLE*   txn handle
struct
** Returns      :
**               int - return code
** Comments     :
**
*****
*/
int doOrderStatusForm(htmlPhraser *commandBlock,TXN_HANDLE
*txnHandle)
{
    char *html=txnHandle->htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Order
Status</TITLE></HEAD>\r\n"
                "<BODY><FORM
ACTION=\"\"
                APP_NAME
                \"\"
METHOD=\"GET\">\r\n"
                "<CENTER><H3>Please Fill In Order Status Form.</H3></CENTER>
                <BR>\r\n"
                "Submit Transaction
                <INPUT TYPE=\"submit\" NAME=\"\"
                CMD_TXN_ID
                \"\" VALUE=\"\"
                CMD_ORDS
                \"\">"
                "<BR> ");
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"<PRE>\r\n"
                "Warehouse: ");
    char buffer[15];
    appendText(&html,itoa(txnHandle->w_id,buffer,10));
}

```



```

appendText(&html," District: <INPUT NAME=""
                                CMD_D_ID
                                "" SIZE=1>\r\n<BR>"
                                "Customer: "
                                "<INPUT NAME=""
                                CMD_C_ID
                                "" SIZE=5>"
                                " "
                                "Name: "
                                "<INPUT NAME=""
                                CMD_C_NAME
                                "" SIZE=20><BR>"
                                "Cust-Balance: <BR>"
                                "Order-Number:
Entry-Date: Carrier-Number<BR>"
                                "Supply-W
Item-Num Qty Amount Delivery<BR></PRE>");
appendText(&html,"</BODY></HTML>");
return OK;
}
/*
*****
** Name : doOrderStatusResults
** Description :
** HTML orderStatus page entry
point
** Parameters :
** htmlPhraser* command
block
** char * html result
page
** Returns :
** int - return code
** Comments :
**
*****
*/
int doOrderStatusResults(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
    char *html=txnHandle->htmlPage;
    struct ords_wrapper *ords = NULL;
    ords = (ords_wrapper *) txnHandle->comInterface.txnBuffer;
    ZeroMemory(ords,maxDataSize);
    //set warehouse login id from command blk
    ords->in_ords.s_W_ID = txnHandle->w_id;
    //set district login id from command blk
    if( ords->in_ords.s_D_ID = atoi(commandBlock->get_D_ID())) ==
0)
    {
doOrderStatusErrorPage(html,ERR_INVALID_D_ID,commandBlock,txnHandl
e);
return OK;
    }
    if( ords->in_ords.s_C_ID = atoi(commandBlock->get_C_ID())) ==
0)
    {
        if(*(commandBlock->get_C_NAME()) == NULL)
        {
            //no customer id nor customer last name
            specified.
doOrderStatusErrorPage(html,ERR_MISSING_C_ID_OR_CLAST,commandBl
ock,txnHandle);
return OK;
        }
    }
}
else
strcpy(ords->in_ords.s_C_LAST,commandBlock->get_C_NAME());
}
else
{
//make sure that the user only inserted just c_id
if(*(commandBlock->get_C_NAME()) != NULL)
{
doOrderStatusErrorPage(html,ERR_C_ID_OR_CLAST_ONLY,commandBloc
k,txnHandle);
return OK;
}
}
appendText(&html,"<HTML><HEAD><TITLE>TPC-C Order
Status Results</TITLE></HEAD>\r\n"
ACTION=""
"APP_NAME
"
METHOD=""GET"">\r\n");
html+=appendButtons(html);
html+=appendHiddenFields(html,txnHandle);
appendText(&html,"</FORM>");
ords->out_ords.s_transtatus = -1;
HRESULThres;
try
{
hres =
txnHandle->comInterface.comHandle->doOrderStatus(&txnHandle->comInterf
ace.size,(UCHAR**)&txnHandle->comInterface.txnBuffer);
}
catch(...)
{
html+=sprintf(html,"ERROR: ords com call caused
exeception.</PRE></BODY></HTML>");
return OK;
}
if(FAILED(hres))
{
html+=sprintf(html,"ERROR: ords com call failed,
rc:%x</PRE></BODY></HTML>",hres);
ERRORMSG("ERROR : ords com call failed,
rc:"<<DEBUGADDRESS(hres)<<endl);
return OK;
}
hres = txnHandle->comInterface.comHandle->doSetComplete();
if(FAILED(hres))
{
html+=sprintf(html,"ERROR: ords com doSetComplete
failed, rc:%ld</PRE></BODY></HTML>",hres);
ERRORMSG("ERROR : ords com doSetComplete failed,
rc:"<<DEBUGADDRESS(hres)<<endl);
return OK;
}
ords = (ords_wrapper *)txnHandle->comInterface.txnBuffer;
int rc = ords->out_ords.s_transtatus;
if( rc != OK)
{
html+=displayStatus(html,rc);
appendText(&html,"</PRE></BODY></HTML>");
ERRORMSG("ERROR order status"<<endl
<<"ords->in_ords.s_C_ID:"<<ords->in_ords.s_C_ID<<endl
<<"ords->in_ords.s_C_LAST:"<<ords->in_ords.s_C_LAST<<endl

```

```

<<"ords->in_ords.s_D_ID:"<<ords->in_ords.s_D_ID<<endl
<<"ords->in_ords.s_W_ID:"<<ords->in_ords.s_W_ID<<endl
<<"ords->out_ords.deadlocks:"<<ords->out_ords.deadlocks<<endl
<<"ords->out_ords.s_C_BALANCE:"<<ords->out_ords.s_C_BALANCE<<en
dl
<<"ords->out_ords.s_C_FIRST:"<<ords->out_ords.s_C_FIRST<<endl
<<"ords->out_ords.s_C_ID:"<<ords->out_ords.s_C_ID<<endl
<<"ords->out_ords.s_C_ID:"<<ords->out_ords.s_C_ID<<endl
<<"ords->out_ords.s_C_MIDDLE:"<<ords->out_ords.s_C_MIDDLE<<endl
<<"ords->out_ords.s_O_CARRIER_ID:"<<ords->out_ords.s_O_CARRIER_ID
<<endl
<<"ords->out_ords.s_O_ENTRY_D_time:"<<ords->out_ords.s_O_ENTRY_D_
time<<endl
<<"ords->out_ords.s_O_ID:"<<ords->out_ords.s_O_ID<<endl
<<"ords->out_ords.s_ol_cnt:"<<ords->out_ords.s_ol_cnt<<endl);
        return OK;
    }
    //start creating result body
appendText(&html,"</FORM><CENTER><H3>Order-Status</H3></CENTE
R>");
    appendText(&html,"<BR><PRE>\r\nWarehouse: ");
    char buffer[50];

    appendText(&html,ittoa(ords->in_ords.s_W_ID,buffer,10),6+1,1);
    appendText(&html,"District: ");
    appendText(&html,ittoa(ords->in_ords.s_D_ID,buffer,10));
    appendText(&html,"<BR>"

        "Customer: ");

    //get customer id
    appendText(&html,ittoa(ords->in_ords.s_C_ID,buffer,10),6+1,1);
    appendText(&html,"Name: ");
    //get first, middle, and last from wrapper
appendText(&html,ords->out_ords.s_C_FIRST,FIRST_NAME_LEN+1,1);
appendText(&html,ords->out_ords.s_C_MIDDLE,INITIALS_LEN+1,1);
appendText(&html,ords->out_ords.s_C_LAST,LAST_NAME_LEN+5,1);
    //get customer balance from wrapper
    appendText(&html,"&#10\r\nCust-Balance: $");
    html+=sprintf(html,"% .2lf",ords->out_ords.s_C_BALANCE/100.0);
    //display order number, entry date, and carrier number
    appendText(&html,"&#10<BR> &#10<BR>"

        "Order-Number ");
    appendText(&html,ittoa(ords->out_ords.s_O_ID,buffer,10),12,1);
    appendText(&html,"Entry-Date: ");
    copyOutDateTime(buffer,ords->out_ords.s_O_ENTRY_D_time);
    appendText(&html,buffer,22,1);
    appendText(&html,"Carrier-Number: ");

appendText(&html,ittoa(ords->out_ords.s_O_CARRIER_ID,buffer,10));
    //add item title columns
    appendText(&html,"&#10<BR>"

        "Supply-W "
        "Item-Id "

```

```

"Qty "
"Amount "
"Delivery-Date<BR>
");
    //display items
    for (int
itemCount=0;itemCount<ords->out_ords.s_ol_cnt;itemCount++)
    {
    //appendSpaces(&html,2);

        //get supp w
appendText(&html,ittoa(ords->out_ords.item[itemCount].s_OL_SUPPLY_W_I
D,buffer,10),11,1);
        //get item num
appendText(&html,ittoa(ords->out_ords.item[itemCount].s_OL_I_ID,buffer,10),
11,1);
        //get item qty
appendText(&html,ittoa(ords->out_ords.item[itemCount].s_OL_QUANTITY,bu
ffer,10),6,1);
        //get item dollor amount
html+=sprintf(html,"%-14.2lf",ords->out_ords.item[itemCount].s_OL_AMOU
NT/100.0);
        //get delivery date
copyOutDate(buffer,ords->out_ords.item[itemCount].s_OL_DELIVERY_D_ti
me);
        appendText(&html,buffer);
        appendText(&html," &#10<BR> ");
    }
    //append line breaks if item count is less than 15
    for (int itemCount=0;itemCount <
(15-ords->out_ords.s_ol_cnt);itemCount++)
        appendText(&html,"&#10<BR> ");

    html+=displayStatus(html,rc);

    appendText(&html,"&#10</PRE>&#10</BODY>&#10</HTML>");
    return OK;
}
/*
*****
** Name : doOrderStatusErrorPage
** Description :
** HTML orderStatus error page
** Parameters :
** char * html page
result
** char * error
message
** htmlPhraser* command
block
** TXN_HANDLE* txn handle
** Returns :
** int - return code
** Comments :
**
*****
*/
int doOrderStatusErrorPage(char *htmlPage,char *message,htmlPhraser
*commandBlock, TXN_HANDLE *txnHandle)
{
    char *html=htmlPage;
    appendText(&html,"&#10<HTML>&#10<HEAD>&#10<TITLE>TPC-C Order
Status</TITLE>&#10</HEAD>&#10\r\n"

```

```

ACTION=\\\"
METHOD=\\\"GET\\\">\\r\\n\"
\"<CENTER><H3>Please Fill In Order Status Form.</H3></CENTER>
<BR>\\r\\n\"
<INPUT TYPE=\\\"submit\\\" NAME=\\\"
\"Submit Transaction
CMD_TXN_ID
\\\" VALUE=\\\"
CMD_ORDS
\\\">\"
\"<BR> \";
html+=appendHiddenFields(html,txnHandle);
appendText(&html,\\\"<PRE>\\r\\n\"
\"Warehouse: \");
char buffer[15];
appendText(&html,ittoa(txnHandle->w_id,buffer,10));
appendText(&html,\" District: <INPUT NAME=\\\"
CMD_D_ID
\\\" SIZE=1>\\r\\n<BR>\"
\"Customer: \"
\"<INPUT NAME=\\\"
CMD_C_ID
\\\" SIZE=5>\"
\" \"
\"Name: \"
\"<INPUT NAME=\\\"
CMD_C_NAME
\\\" SIZE=20><BR>\"
\"Cust-Balance: <BR>\"
\"Order-Number:
Entry-Date:           Carrier-Number<BR>\"
Item-Num  Qty    Amount    Delivery <BR>\";
appendText(&html,message);
appendText(&html,\\\"</PRE></BODY></HTML>\\\");
return OK;
}
/*
*****
** Name             : doDeliveryForm
** Description      :
**                  HTML payment page entry point
** Parameters       :
**                  htmlPhraser*    command
block
**                  TXN_HANDLE*    txn handle
struct
** Returns          :
**                  int - return code
** Comments         :
**
*****
*/
int doDeliveryForm(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
    char *html=txnHandle->htmlPage;
    appendText(&html,\\\"<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD>\\r\\n\"
\"<BODY><FORM
APP_NAME
\\\"
METHOD=\\\"GET\\\">\\r\\n\"
html+=appendButtons(html);
APP_NAME
\\\"
METHOD=\\\"GET\\\">\\r\\n\"
\"<CENTER><H3>Delivery.</H3></CENTER>\\r\\n\"
\"Submit Transaction
\"<INPUT TYPE=\\\"submit\\\" NAME=\\\"
\"Submit Transaction
CMD_TXN_ID
\\\" VALUE=\\\"
CMD_DLTV
\\\">\";
html+=appendHiddenFields(html,txnHandle);
appendText(&html,\\\"<BR> <PRE>\"
\"Warehouse: \");
char buffer[10];
appendText(&html,ittoa(txnHandle->w_id,buffer,10));
appendText(&html,\\\" <BR> <BR>\"
\"Carrier Number: \"
\"<INPUT NAME=\\\"
CMD_CARRIER_NUM
\\\" SIZE=1>\"
\"</FORM></PRE>\\\");
appendText(&html,\\\"</BODY></HTML>\\\");
return OK;
}
/*
*****
** Name             : doDeliveryResults
** Description      :
**                  HTML payment page entry point
** Parameters       :
**                  htmlPhraser*    command
block
**                  TXN_HANDLE*    txn handle
** Returns          :
**                  int - return code
** Comments         :
**
*****
*/
int doDeliveryResults(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
    char *html = txnHandle->htmlPage;
    //declare delivery structure
    struct dlvy_wrapper dlvy;
    //set warehouse login id from command blk
    dlvy.in_dlvy.s_W_ID = txnHandle->w_id;
    //set the carrier id from command blk
    if( dlvy.in_dlvy.s_O_CARRIER_ID =
atoi(commandBlock->get_CARRIER_NUM()) == 0)
    {
doDeliveryErrorPage(html,ERR_INVALID_CARRIER,commandBlock,txnHan
dle);
return OK;
}
    //print title, add hidden fields , txn buttons
    appendText(&html,\\\"<HTML><HEAD><TITLE>TPC-C Delivery
Results</TITLE></HEAD>\\r\\n<BODY><FORM ACTION=\\\"
APP_NAME
\\\"
METHOD=\\\"GET\\\">\\r\\n\";
html+=appendButtons(html);

```

```

        html+=appendHiddenFields(html,txnHandle);
        appendText(&html,
"<FORM><CENTER><H3>Delivery</H3></CENTER>");
        int rc =
queueDlvyTxn(dlvy.in_dlv.s_W_ID,dlvy.in_dlv.s_O_CARRIER_ID);
        if( rc != OK)
        {
                html+=displayStatus(html,rc);
                appendText(&html,"</PRE></BODY></HTML>\r\n");
                ERRORMSG("ERROR: Unable to queue dlvy txn,
rc:"<<rc<<endl);
                return OK;
        }
//start creating result body
appendText(&html,"Warehouse: ");

//get w_id from wrapper
char buffer[15];
appendText(&html,itoa(dlvy.in_dlv.s_W_ID,buffer,10));
appendText(&html,"<BR><BR>Carrier Number: ");

//get carrier_id from wrapper

appendText(&html,itoa(dlvy.in_dlv.s_O_CARRIER_ID,buffer,10));
appendText(&html,"<BR><BR>Execution Status: Delivery has
been queued </PRE></BODY></HTML>");
return OK;
}
/*
*****
** Name          : doDeliveryErrorPage
** Description   :
**              HTML payment error page entry
point
** Parameters   :
**              char *      html result
page
**              char *      error
message
**              htmlPhraser  command
block
**              TXN_HANDLE*  txn handle
**
** Returns      :
**              int - return code
** Comments     :
**
*****
*/
int doDeliveryErrorPage(char *htmlPage,char *message,htmlPhraser
*commandBlock,TXN_HANDLE *txnHandle)
{
        char *html=htmlPage;
        appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD>\r\n"
                "<BODY><FORM
ACTION=\"\"
APP_NAME
\"\"
METHOD=\"GET\">\r\n"
"<CENTER><H3>Delivery.</H3></CENTER>\r\n"
"<INPUT TYPE=\"submit\" NAME=\"\"
CMD_TXN_ID
\"\" VALUE=\"\"
CMD_DLVS
\">");
}

```

```

        html+=appendHiddenFields(html,txnHandle);
        appendText(&html,"<BR><PRE>
                "Warehouse: ");
        char buffer[15];
        appendText(&html,itoa(txnHandle->w_id,buffer,10));

        appendText(&html," <BR> <BR>"
                "Carrier Number: "
                "<INPUT NAME=\"\"
CMD_CARRIER_NUM
                \"\" SIZE=1> <BR>");
        appendText(&html,message);
        appendText(&html,"</PRE></BODY></HTML>");
        return OK;
}
/*
*****
** Name          : doStockForm
** Description   :
**              HTML stock page entry point
** Parameters   :
**              htmlPhraser  command
block
**              TXN_HANDLE*  txn handle
** Returns      :
**              int - return code
** Comments     :
**
*****
*/
int doStockForm(htmlPhraser *commandBlock,TXN_HANDLE *txnHandle)
{
        char *html=txnHandle->htmlPage;
        appendText(&html,"<HTML><HEAD><TITLE>TPC-C Stock
Level</TITLE></HEAD>\r\n"
                "<BODY><FORM
ACTION=\"\"
APP_NAME
\"\"
METHOD=\"GET\">\r\n"
"<CENTER><H3>Please Fill In Stock Form.</H3></CENTER> <BR>\r\n"
"<INPUT TYPE=\"submit\" NAME=\"\"
CMD_TXN_ID
\"\" VALUE=\"\"
CMD_STOK
\">");
        html+=appendHiddenFields(html,txnHandle);
        appendText(&html,"<PRE>
                "Warehouse: ");
        char buffer[15];
        appendText(&html,itoa(txnHandle->w_id,buffer,10),6+1,1);
        appendText(&html,"District: ");
        appendText(&html,itoa(txnHandle->d_id,buffer,10));
        appendText(&html," <BR> <BR>"
                "Stock Level
Threshold: "
                "<INPUT NAME=\"\"
CMD_STK_THRESHOLD
                \"\" SIZE=1> <BR>
<BR>"
                "Low Stock: <BR>"
                "</PRE>");
}

```

```

        appendText(&html,"</FORM></BODY></HTML>");
        return OK;
    }
}
/*
*****
** Name          : doStockResults
** Description    :
**               HTML stock page entry point
** Parameters    :
**               htmlPhraser*    command
block
**               TXN_HANDLE*    txn handle
struct
** Returns      :
**               int - return code
** Comments     :
**
*****
*/
int doStockResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    char *html = txnHandle->htmlPage;
    struct stok_wrapper *stok;
    stok = (stok_wrapper*)txnHandle->comInterface.txnBuffer;
    ZeroMemory(stok,maxDataSize);
    //set warehouse login id from command blk
    stok->in_stok.s_W_ID = txnHandle->w_id;
    //set district login id from command blk
    stok->in_stok.s_D_ID = txnHandle->d_id;
    //set stock level threshold id from command blk
    if( (stok->in_stok.s_threshold =
atoi(commandBlock->get_STK_THRESHOLD()) == 0)
    {
        doStockErrorPage(html,ERR_INVALID_THRESHOLD,commandBlock,txnHandle);
        return OK;
    }
    //assume failure, set s_transtatus to err
    stok->out_stok.s_transtatus = INVALID_STATUS;
    //print title, add hidden fields , txn buttons
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Stock
Level Results</TITLE></HEAD>\r\n"
                "<BODY><FORM
ACTION=""
                APP_NAME
                ""
METHOD=""GET"">\r\n");
    html+=appendButtons(html);
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"</FORM>");
    stok->out_stok.s_transtatus = -1;

    DEBUGMSG("Calling com entry api for stock call,
w_id:<<stok->in_stok.s_W_ID<< d_id:<<stok->in_stok.s_D_ID<<
threshold:<<stok->in_stok.s_threshold<<endl);
    HRESULThres;
    try
    {
        hres =
txnHandle->comInterface.comHandle->doStockLevel(&txnHandle->comInterface.size,(UCHAR**)&txnHandle->comInterface.txnBuffer);
    }
    catch(...)
    {
        html+=sprintf(html,"ERROR: Com Stock call caused
exception to occur.</PRE></BODY></HTML>");

```

```

        ERRORMSG("ERROR : Com Stock call caused
exception to occur."<<endl);
        return OK;
    }
    if(FAILED(hres))
    {
        html+=sprintf(html,"ERROR: stok com call failed,
rc:%ld</PRE></BODY></HTML>",hres);
        ERRORMSG("ERROR : stok com call failed,
rc:<<DEBUGADDRESS(hres)<<endl);
        return OK;
    }
    hres = txnHandle->comInterface.comHandle->doSetComplete();
    if(FAILED(hres))
    {
        html+=sprintf(html,"ERROR: stok com doSetComplete
failed, rc:%ld</PRE></BODY></HTML>",hres);
        ERRORMSG("ERROR : stok com doSetComplete failed,
rc:<<DEBUGADDRESS(hres)<<endl);
        return OK;
    }
    stok = (stok_wrapper *)txnHandle->comInterface.txnBuffer;
    int rc = stok->out_stok.s_transtatus;
    if(rc != OK)
    {
        html+=displayStatus(html,rc);
        appendText(&html,"</PRE></BODY></HTML>");
        ERRORMSG("ERROR stok txn failed"<<endl
<<"stok->in_stok.s_D_ID:"<<stok->in_stok.s_D_ID<<endl
<<"stok->in_stok.s_threshold:"<<stok->in_stok.s_threshold<<endl
<<"stok->in_stok.s_W_ID:"<<stok->in_stok.s_W_ID<<endl
<<"stok->out_stok.deadlocks:"<<stok->out_stok.deadlocks<<endl
<<"stok->out_stok.s_low_stock:"<<stok->out_stok.s_low_stock<<endl
<<"stok->out_stok.s_transtatus:"<<stok->out_stok.s_transtatus<<endl);
        return OK;
    }
    //start creating result body
    appendText(&html,"<FORM><CENTER><H3>Stock-Level</H3></CENTER>");
    appendText(&html,"<BR><PRE>\r\n"
                "Warehouse: ");
    //get w_id from wrapper
    char buffer[10];
    appendText(&html,ittoa(stok->in_stok.s_W_ID,buffer,10),6+1,1);
    appendText(&html,"District: ");
    appendText(&html,ittoa(stok->in_stok.s_D_ID,buffer,10));
    appendText(&html,"<BR> <BR>"
                "Stock Level
Threshold: ");
    appendText(&html,ittoa(stok->in_stok.s_threshold,buffer,10));
    appendText(&html,"<BR> <BR>"
                "Low Stock: ");
    appendText(&html,ittoa(stok->out_stok.s_low_stock,buffer,10));
    appendText(&html,"<BR> <BR>");
    html+=displayStatus(html,rc);
    appendText(&html,"</PRE></BODY></HTML>");
    return OK;
}
/*

```

```

*****
** Name : doStockErrorPage
** Description :
** HTML stock page entry point
** Parameters :
** char * html result
page
** char * query string
** htmlPhraser command
block
** TXN_HANDLE * handle for
this transaction
** Returns :
** int - return code
** Comments :
*****
*/
int doStockErrorPage(char *htmlPage,char *message,htmlPhraser
*commandBlock,TXN_HANDLE *txnHandle)
{
    char *html=htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Stock
Level</TITLE></HEAD>\r\n"
ACTION="\
APP_NAME
"
METHOD="\GET">\r\n"
"<CENTER><H3>Please Fill In Stock Form.</H3></CENTER> <BR>\r\n"
"Submit Transaction
<INPUT TYPE="submit" NAME="\
CMD_TXN_ID
" VALUE="\
CMD_STOK
">");
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"<PRE>
"Warehouse: ");
    char buffer[15];
    appendText(&html,itoa(txnHandle->w_id,buffer,10));
    appendSpaces(&html,2);
    appendText(&html,"District: ");
    appendText(&html,commandBlock->get_D_ID());
    appendText(&html," <BR> <BR>"
"Stock Level
Threshold: "
"<INPUT NAME="\
CMD_STK_THRESHOLD
" SIZE=1> <BR>
<BR>"
"Low Stock: <BR>");
    appendText(&html,message);
    appendText(&html,"</PRE></FORM></BODY></HTML>");
    return OK;
}
/*
*****
** Name : doExit
** Description :
** HTML exit page entry point
** Parameters :
*****
htmlPhraser* command
block
TXN_HANDLE* txn handle
struct
** Returns :
** int - return code
** Comments :
*****
*/
int doExit(htmlPhraser *commandBlock,TXN_HANDLE *txnHandle)
{
    return (doLoginForm(commandBlock,txnHandle));
}
/*
*****
** Name : displayStatus
** Description :
** appends status string to the html
page
** Parameters :
** char* html page
** int rc
** Returns :
** amount of characters the function
appened
** to the html page
** Comments :
*****
*/
int displayStatus(char *htmlPage,int rc)
{
    char *html = htmlPage;
    appendText(&html,"");
    switch (rc)
    {
        case OK:
            appendText(&html,"Execution Status: Transaction
Committed",50,1);
            break;
        case INVALID_ITEM:
            appendText(&html,"Execution Status: Item number is not
valid",50,1);
            break;
        case INVALID_STATUS:
            appendText(&html,"Execution Status: ERROR: Rollback
INVALID_STATUS",50,1);
            break;
        case INVALID_COM_STATUS:
            appendText(&html,"Execution Status: ERROR: Rollback
COM FAILURE",50,1);
            break;
        case ERR_DLVY_QUEUE_FULL:
            appendText(&html,"Execution Status: ERROR: Rollback
DLVY QUEUE FULL",50,1);
            break;
        default:
            appendText(&html,"Execution Status: ERROR:
Rollback",50,1);
    };
    appendText(&html," ");
    return (int)(html - htmlPage);
}
/*
*****
** Name : appendButtons

```

```

** Description      :
**                  append hidden field to recognize
user after login
**
** Parameters       :
**                  *htmlPage
html result page
**
**                  *TXN_HANDLE
txn handle
** Returns         :
**                  int
amount of characters the function appened
**
**                  to the html page
** Comments        :
**
*****
*/
int appendHiddenFields(char *htmlPage, TXN_HANDLE *txnHandle)
{
    char *html = htmlPage;
    char buffer[15];
    appendText(&html, "<INPUT TYPE=\"hidden\" NAME=\"\"
                CMD_TERM_ID
                \"\" VALUE=\"\"");
    appendText(&html, itoa(txnHandle->term_id, buffer, 10));
    appendText(&html, ">\r\n");
    return (int)(html-htmlPage);
}
/*
*****
** Name            : appendButtons
** Description     :
**                  appends buttons transaction
buttons to result page
** Parameters      :
**                  *htmlPage
**
** Returns        :
**                  amount of characters the function
appened
**
**                  to the html page
** Comments       :
**
*****
*/
int appendButtons(char *htmlPage)
{
    char *html = htmlPage;
    appendText(&html, "<INPUT TYPE=\"submit\" NAME=\"\"
                CMD_TXN_ID
                \"\" VALUE=\"\"
                CMD_NORD
                \"\">\r\n"
                "<INPUT
                CMD_TXN_ID
                \"\" VALUE=\"\"
                CMD_PYMT
                \"\">\r\n"
                "<INPUT
                CMD_TXN_ID
                \"\" VALUE=\"\"
                CMD_ORDS
                \"\">\r\n");
}
/*
*****
** Name            : appendItems
** Description     :
**                  appends items to new order and
order status page
** Parameters      :
**                  *htmlPage
**
**                  html result page
**
**                  items to append
**
**                  item CMD id start
**
** Returns        :
**                  amount of characters the function
appened
**
**                  to the html page
** Comments       :
**
*****
*/
int appendItems(char *htmlPage, short itemCount, short cmdIDStart)
{
    char *html = htmlPage;
    char numBuffer[MAX_INT_BUFFER];
    for(int item=0; item < itemCount; item++)
    {
        appendText(&html, "<BR> <INPUT NAME=\"\"");
        appendText(&html, itoa(cmdIDStart++, numBuffer, 10));
        appendText(&html, "\" SIZE=6> <INPUT NAME=\"\"");
        appendText(&html, itoa(cmdIDStart++, numBuffer, 10));
        appendText(&html, "\" SIZE=6>
        <INPUT NAME=\"\"");
        appendText(&html, itoa(cmdIDStart++, numBuffer, 10));
        appendText(&html, "\" SIZE=2>\r\n");
    }
    return (int)(html - htmlPage);
}
/*
*****
** Name            : dlvyThreadEntry
** Description     :
**                  dlvy thread worker entry point
** Parameters      :
**
** Returns        :

```

```

**
** Comments
**
initDly enter at
**
**
connection
**
**
**
worker semaphore
**
**
**
*****
*/
void dlvThreadEntry(void *)
{
    int rc = 0;
    DEBUGMSG("dlvyThread " << GetCurrentThreadId() << " entered
dlvyThreadEntry, calling db_connect to db:" << dbName << endl);

    void *connectHandle;
    //connect to database.
    DEBUGMSG("ptr created. calling db_connect to db:" << dbName
<< endl);
    rc = db_connect(dbName,&connectHandle);

    if(rc != OK)
    {
        ERRORMSG("dlvyThread " << GetCurrentThreadId()
<<" unable to connect to database, rc:" << rc << endl);
        DEBUGMSG("dlvyThread " << GetCurrentThreadId()
<<" unable to connect to database, rc:" << rc << endl);
        return;
    }

    DEBUGMSG("dlvyThread " << GetCurrentThreadId() << " connect
to db:" << dbName << " successful" << endl);

    FILE *dlvyLog = NULL;
    char logFileName[MAX_STRING_LEN] = {NULL};

    EnterCriticalSection(&isapiLock);
    //open dlvy log file for this thread
    sprintf(logFileName,"%s\\del_%d.txt",dlvyLogPath,dlvyThreadID);
    dlvLog = fopen(logFileName,"w");
    if(!dlvyLog)
    {
        ERRORMSG("dlvyThread " << GetCurrentThreadId()
<<" unable to open dlvy log "
<< dlvyLogPath << "\\del_" <<
dlvyThreadID << endl);
        DEBUGMSG("dlvyThread " << GetCurrentThreadId()
<<" unable to open dlvy log "
<< dlvyLogPath << "\\del_" <<
dlvyThreadID << endl);
        return;
    }
    //increment the global dlvy thread id
    dlvThreadID++;
    LeaveCriticalSection(&isapiLock);

    DEBUGMSG("dlvyThread " << GetCurrentThreadId() <<" dlvy log
file name: " << logFileName << " open." << endl);
    HANDLE workerHandles[2];
    //handle array to store event to wait on
    struct DLVYQUEUEDATA dlvQueueData;
    //dlvy queue struct to store queued txn

```

```

    struct dlv_wrapper dlvTxn;
    //dlvy wrapper of db2 structs
    struct _timeb
endQueueTime; //time stamp to queue removal time
    struct _timeb
endProcessTime; //time stamp for end process time
    char orderIDs[MAX_STRING_LEN] = {NULL};
    //string to store oids for each district
    int bytesWritten = 0;
    int dlvCount = 0;
    DEBUGMSG("dlvyThread entering work loop" << endl);
    //successful, while true
    while(true)
    {
        try
        {
            DEBUGMSG("dlvyThread initializing wait
handles" << endl);

            //wait for both program exit AND if there is
work to do
            workerHandles[0] = dlvThreadDone;
            workerHandles[1] = dlvThreadSemaphore;
            DEBUGMSG("dlvyThread going to sleep
waiting for wrk" << endl);
            rc =
WaitForMultipleObjects(2,&workerHandles[0],FALSE,INFINITE);

            DEBUGMSG("dlvyThread awake, checking
wake condition" << endl);
            if(rc == WAIT_OBJECT_0)
                break;
            else if(rc == (WAIT_OBJECT_0+1))
            {
                DEBUGMSG("dlvyThread awake,
wake condition of dlvThreadSemaphore" << endl);
            }
            DEBUGMSG("dlvyThread trying to enter
critical section" << endl);

            EnterCriticalSection(&dlvQueueLock);

            DEBUGMSG("dlvyThread entered critical
section" << endl);

            //remove queued dlvy txn
            dlvQueueData.enqueueTime.time
= dlvQueue[dlvyBufferThreadIndex].enqueueTime.time;
            dlvQueueData.enqueueTime.millitm
= dlvQueue[dlvyBufferThreadIndex].enqueueTime.millitm;
            dlvQueueData.in_s_0_CARRIER_ID
= dlvQueue[dlvyBufferThreadIndex].in_s_0_CARRIER_ID;
            dlvQueueData.warehouse
= dlvQueue[dlvyBufferThreadIndex].warehouse;

            DEBUGMSG("dlvyThread removed dlvy:"
<< dlvCount << ",w_id:" << dlvQueueData.warehouse
<< " carrier_id:" << dlvQueueData.in_s_0_CARRIER_ID << endl);
            DEBUGMSG("dlvyThread removed dlvy in
queue index: " <<dlvyBufferThreadIndex<< " w_id: " <<
dlvQueueData.warehouse
<< " carrier_id: " << dlvQueueData.in_s_0_CARRIER_ID << endl);
            //increment the number of free slots
            dlvBufferFreeSlots++;
            //increment the thread index to next slot in
dlvy queue

```



```

        dlvyBufferThreadIndex++;
        DEBUGMSG("dlvyThread incremented
amount of free slots:" << dlvyBufferFreeSlots << " and thread index:" <<
        dlvyBufferThreadIndex << endl);
        //check if we reached the end of dlvy queue, if
so, reset back index back to 0
        if(dlvyBufferThreadIndex == dlvyQueueLen)
        {
            DEBUGMSG("dlvyThread reset
dlvyBufferThreadIndex to 0, current dlvyBufferThreadIndex:" <<
dlvyBufferThreadIndex
            << " free
slots:"<<dlvyBufferFreeSlots<<endl);
            dlvyBufferThreadIndex=0;
        }
        DEBUGMSG("dlvyThread releasing critical
section" << endl);
        LeaveCriticalSection(&dlvyQueueLock);
        //take enqueue time
        _ftime(&endQueueTime);
        DEBUGMSG("dlvyThread executing txn
w_id:" << dlvyQueueData.warehouse
        << " carrier_id:" <<
dlvyQueueData.in_s_0_CARRIER_ID << endl);
        //prepare to call database
        dlvyTxn.in_dlvy.s_O_CARRIER_ID
dlvyQueueData.in_s_0_CARRIER_ID;
        dlvyTxn.in_dlvy.s_W_ID
= dlvyQueueData.warehouse;
        dlvyTxn.out_dlvy.s_transtatus = -1;
        //increment dlvy count
        dlvyCount++;
        DEBUGMSG("dlvyThread %d calling dlvy
txn" << rc << endl);
        //call dlvy txn
        rc = dlvyCall(&dlvyTxn,connectHandle);
        _ftime(&endProcessTime);
        rc = dlvyTxn.out_dlvy.s_transtatus;
        DEBUGMSG("dlvy txn response time:"<<
        (((endProcessTime.time -
endQueueTime.time)*1000)+
(endProcessTime.millitm - endQueueTime.millitm))/1000.0)<<
        "
w_id:"<<dlvyTxn.in_dlvy.s_W_ID<<" carrier:"
<<dlvyTxn.in_dlvy.s_O_CARRIER_ID<<
        "
deadLocks:"<<dlvyTxn.out_dlvy.deadlocks<<" rc: "<< rc <<endl);
        DEBUGMSG("dlvyThread dlvy s_transtatus:"
<< rc << endl);
        if(rc == OK)
        {
            bytesWritten=0;
            char *buffer = orderIDs;
            for(int
            districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
            {
                if(dlvyTxn.out_dlvy.s_O_ID[districtIndex] == 0)

```

```

            bytesWritten
            = sprintf(buffer,"%nD_ID %d had no new orders",districtIndex);
            else
            bytesWritten
            = sprintf(buffer,"%d ",dlvyTxn.out_dlvy.s_O_ID[districtIndex]);
            buffer+=bytesWritten;
        }
        else
        sprintf(orderIDs,"%nDelivery
transaction failed");
        fprintf(dlvyLog,DELIVERY_LOG_SUCCESS_STR,
dlvyCount,
dlvyQueueData.enqueueTime.time,
dlvyQueueData.enqueueTime.millitm,
endQueueTime.time,
endQueueTime.millitm,
dlvyQueueData.warehouse,
dlvyQueueData.in_s_0_CARRIER_ID,
orderIDs,
endProcessTime.time,
endProcessTime.millitm);
        fflush(dlvyLog);
    }
    catch(...)
    {
        ERRORMSG("ERROR: Unhandled
exeception in dlvy thread. Thread exiting"<<endl);
        fprintf(dlvyLog,"ERROR: Unhandled
exeception in dlvy thread %ld. Thread exiting.\n",GetCurrentThreadId());
        fflush(dlvyLog);
        LeaveCriticalSection(&dlvyQueueLock);
    }
} //end while true
}
/*
*****
** Name : queueDlvyTxn
** Description :
** function queues dlvy txn in dlvy
queue
** Parameters :
** int warehouse
** short carrier
** Returns :
** int error code
** Comments :
** Function will queue
dlvy txn if 2 points are true
** 1) We have room in our
dlvy buffer
** 2) We writing over the
end of the queue
**

```

```

*****
*/
int queueDlvyTxn(int warehouse, short carrier_id)
{
    DEBUGMSG("Taking lock to queue dlvy txn.");
    EnterCriticalSection(&dlvyQueueLock);
    DEBUGMSG("Lock aquired to queue dlvy txn");
    if(dlvyBufferFreeSlots)
    {
        DEBUGMSG("Checking if we are inserting at tail of
dlvy queue."<<endl);
        if( dlvyBufferSlotIndex == (dlvyBufferThreadIndex-1))
        {
            ERRORMSG("Error dlvy queue inserting
over unserviced queued dlvy txn."<<endl);
            DEBUGMSG("Error dlvy queue inserting
over unserviced queued dlvy txn."<<endl);
            LeaveCriticalSection(&dlvyQueueLock);
            return
ERR_DLVE_QUEUE_EATING_TAIL;
        }
        DEBUGMSG("free slots dlvy
queue:"<<dlvyBufferFreeSlots<<" inserting txn in slot: "
<<dlvyBufferSlotIndex<<
        "w_id:"<<warehouse<<" carrier:
"<<carrier_id<<endl);
        dlvyQueue[dlvyBufferSlotIndex].warehouse =
warehouse;
        dlvyQueue[dlvyBufferSlotIndex].in_s_0_CARRIER_ID
= carrier_id;

        _ftime(&dlvyQueue[dlvyBufferSlotIndex].enqueueTime);
        //decrement the number of free slots in the buffer
        dlvyBufferFreeSlots--;

        //increment the index to the next dlvy queue slot.
        dlvyBufferSlotIndex++;

        DEBUGMSG("dlvy txn queued, slots available in
queue:"<<dlvyBufferFreeSlots<<" queue slot index:"<<dlvyBufferSlotIndex
        <<"w_id:"<<warehouse<<"
carrier:"<<carrier_id<<endl);
        DEBUGMSG("dlvy txn queued, slots available in queue:
"<<dlvyBufferFreeSlots<<" queue slot index: "<<dlvyBufferSlotIndex
        <<" w_id: "<<warehouse<<"
carrier: "<<carrier_id<<endl);
        if(dlvyBufferSlotIndex == dlvyQueueLen)
        {
            DEBUGMSG("queue slot index hit end of
queue, reset to 0, current index:"<<dlvyBufferSlotIndex<<" free
slots:"<<dlvyBufferFreeSlots<<endl);
            dlvyBufferSlotIndex=0;
        }
        else
        {
            //no slots available in dlvy buffer, release critical section
            and return an nord->in_nord.in_item
            LeaveCriticalSection(&dlvyQueueLock);
            ERRORMSG("dlvy queue buffer full, increase the dlvy
queue length."<<endl);
            return ERR_DLVE_QUEUE_FULL;
        }
        LeaveCriticalSection(&dlvyQueueLock);
        //release semaphore to wake thread that there is work
        ReleaseSemaphore(dlvyThreadSemaphore,1,NULL);
        return OK;
    }
}

```

```

/*
*****
** Name          : doHtml
** Description   :
**              : HTML processing page entry
point
** Parameters    :
**              : txn handle
** Returns      :
**              : int - return code
** Comments     :
**
*****
*/
void doHtml(TXN_HANDLE *txnHandle)
{
    DEBUGMSG("Entered doHtml(), parsing query string:"<<
txnHandle->urlString <<" into command block"<<endl);
    htmlPhraser        commandBlock(txnHandle->urlString);
    DEBUGMSG("Query string parsed. command:"<<
commandBlock.getCommandId() <<" user's terminal id:" <<
commandBlock.get_TERM_ID() <<endl);

    int terminalID = atoi(commandBlock.get_TERM_ID());
    int commandID = commandBlock.getCommandId();
    DEBUGMSG("User sent in a terimal id:"<<terminalID<<", checking
to see if user has logged in before"<<endl);
    if(terminalID > 0)
    {
        DEBUGMSG("Terminal id > 0, user has logged in
already, terminalID:"<<terminalID<<" retrieving warehouse district
pair"<<endl);

        if(getTerminal(terminalID,txnHandle) != OK)
            return;
        DEBUGMSG("User had valid terminal id, user's login
warehouse:"<<txnHandle->w_id<<" district:"<<txnHandle->d_id<<endl);
    }
    else
    {
        DEBUGMSG("User did not submit a terminal id or valid
terminal id, ensure that the user is trying to log in."<<endl);
        if( (commandID != TXN_LOGIN) && (commandID !=
TXN_LOGIN_RESULTS) )
        {
            DEBUGMSG("ERROR : User has not logged
in."<<endl);
            ERRORMSG("ERROR : User has not logged
in."<<endl);
            sprintf(txnHandle->htmlPage,"ERROR: User
has not logged in or did not submit a valid terminal.");
            return;
        }
        DEBUGMSG("User is in process of logging in,
commandID:"<<commandID<<endl);
    }
    DEBUGMSG("Calling html page
function:"<<commandBlock.getCommandId()<<endl);
    int rc =
htmlPageFunctions[commandBlock.getCommandId()](&commandBlock,txnHa
ndle);
    DEBUGMSG("Return from html page
function:"<<commandBlock.getCommandId()<<endl);
    return;
}
/*
*****

```

```

** Name           : getTerminal
** Description    :
**               retrieves terminal information
based on terminal id
** Parameters    :
**               int
terminal id
**               TERM_HANDLE* txn handle
** Returns      :
**               int - return code
** Comments     :
**
*****
*/
int getTerminal(int terminal, TXN_HANDLE *txnHandle)
{
    //check to see if terminal id is out of range
    if(terminal >= numUsers)
    {
        //terminal id not valid.
        sprintf(txnHandle->htmlPage, "ERROR: Client does not
support more than %d users, terminal id:%d", numUsers, terminal);
        ERRORMSG("ERROR : Client does not support more
than "<<numUsers<<" users, terminal id:"<<terminal<<endl);
        return ERR;
    }
    //check if terminal id is points to a not in use terminal
    if(!(termArray+terminal)->terminalInUse)
    {
        sprintf(txnHandle->htmlPage, "ERROR: Terminal id
given points to a not in use terminal.");
        ERRORMSG("ERROR : Terminal id given points to a
not in use terminal."<<endl);
        return ERR;
    }
    DEBUGMSG("Storing terminal warehouse, district , and initial term
id for user:"<<terminal<<endl);
    //assign terminal values to txn_handle
    txnHandle->d_id = termArray[terminal].d_id;
    txnHandle->w_id = termArray[terminal].w_id;
    txnHandle->term_id = termArray[terminal].terminalID;
    DEBUGMSG("Users terminal:"<<terminal<<" , stored
warehouse:"<<txnHandle->w_id<<
" district:"<<txnHandle->d_id<<" terminalID
stored:"<<txnHandle->term_id<<endl);
    return OK;
}
/*
*****
** Name           : assignTerminal
** Description    :
**               assigns terminal index to user
** Parameters    :
**               TERM_HANDLE* txn handle
** Returns      :
**               int - return code
** Comments     :
**
*****
*/
int assignTerminal(TXN_HANDLE *txnHandle)
{
    EnterCriticalSection(&termLock);

    //check if terminal array is full.
    if(termNextFree == numUsers)
    {
        LeaveCriticalSection(&termLock);

```

```

        return ERR;
    }
    DEBUGMSG("Storing user warehouse:"<<txnHandle->w_id<<"
district:"<< txnHandle->d_id<<
" in terminal slot:"<<termNextFree<<endl);
    //store users w_id and d_id
    termArray[termNextFree].d_id = txnHandle->d_id;
    termArray[termNextFree].w_id = txnHandle->w_id;

    //set terminal slot to be in use
    termArray[termNextFree].terminalInUse = true;
    termArray[termNextFree].terminalID = termNextFree;
    //in txn handle, set the terminal id
    txnHandle->term_id = termNextFree;

    //increment to next free terminal.
    termNextFree++;
    DEBUGMSG("User warehouse:"<<txnHandle->w_id<<"
district:"<< txnHandle->d_id <<
" stored in terminal slot:"<<txnHandle->term_id<<" next
terminal free:"<<termNextFree<<endl);
    LeaveCriticalSection(&termLock);
    return OK;
}

```

A.2 Client Transaction Code

Makefile.config

```

#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#####
#####
#
# Makefile.config - NT/Win2000 Makefile Configuration
#
# Make Configuration (MSVC)
MAKE=nmake.exe

# Compiler Configuration (MSVC).
# CFLAGS_DEBUG may be set to "-Zi -Od", "-DDEBUGIT" "-Zi -Od
-DDEBUGIT" or left blank
CC=cl.exe
CFLAGS_OS=-DSQLWINT -MT -GS- -DWIN64 -J -Zp8
-DREG_KIT_METHOD
CFLAGS_OUT=/Fo
CFLAGS_DEBUG=

# Linker Configuration (MSVC)
LD_EXEC=link.exe
LD_STORP=link.exe
LDFLAGS_EXEC=
LDFLAGS_SHLIB=/DLL
LDFLAGS_STORP=$(LDFLAGS_SHLIB) /DEF:rpctpc.def

```

```
LDFLAGS_LIB=/LIBPATH:$(TPCC_SQLLIB)\lib
/LIBPATH:"C:\MsSDKx64\Lib\AMD64" db2api.lib WinMM.lib
LDFLAGS_OUT=/OUT:
```

```
# Library Configuration
AR=lib.exe
ARFLAGS=
ARFLAGS_LIB=
ARFLAGS_OUT=/OUT:
```

```
# OS Commands
ERASE=del /F
ERASEDIR=rmdir /S
MOVE=MOVE
COPY=COPY
```

```
# OS File Extensions & Path Separator
OBJEXT=.obj
LIBEXT=.lib
SHLIBEXT=.dll
BINEXT=.exe
SLASH=\\
CMDSEP=&
```

tpccenv.bat

```
@REM
*****
@REM Licensed Materials - Property of IBM
@REM
@REM Governed under the terms of the International
@REM License Agreement for Non-Warranted Sample Code.
@REM
@REM (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
@REM All Rights Reserved.
@REM
@REM US Government Users Restricted Rights - Use, duplication or
@REM disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
@REM
*****
@REM
@REM tpccenv.bat - Windows Environment Setup
@REM

@REM The Kit Version
set TPCC_VERSION=CK041012

@REM The DB2 Instance Name (for DB2)
set DB2INSTANCE=DB2

@REM The OS being used (i.e. "UNIX", "WINDOWS")
set PLATFORM=WINDOWS

@REM The type of make command and slash used by the OS
@REM (i.e. UNIX - "/", WINDOWS - "\\")
@REM These are referenced all over the kit.
set SLASH=\\
set MAKE=nmake

set TPCC_SPTYPE=SPGENERAL

set DB2VERSION=v8

@REM The schema name is typically the SQL authorization ID (or username).
```

```
@REM This is required for runstats and EEE.
set TPCC_SCHEMA=%USERNAME%
```

```
@REM DB2 EE/EEE Configuration
set DB2EDITION=EE
set DB2NODE=0
set DB2NODES=1
```

```
@REM TPCC General Configuration
set HOME=C
set TPCC_DBNAME=TPCC
set TPCC_ROOT=c:\tpcekit\tpc-c.ibm
set TPCC_SQLLIB=c:\SQLLIB
set TPCC_RUNDATA=c:\tpcekit\tpccdata
```

```
@REM TPCC Debug Configuration
set TPCC_DEBUGDIR=c:\temp
```

```
@REM Specifies where stored procedures should be placed and if they should
@REM be fenced.
set TPCC_SPDIR=%TPCC_SQLLIB%\function
set TPCC_FENCED=NO
```

include/db2tpcc.h

```
/*
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****/
/*
 * db2tpcc.h - Macros and Miscellany
 */
#ifndef __DB2TPCC_H
#define __DB2TPCC_H
#include <sys/types.h>
#include "lval.h"
/*
***** */
/* Transaction Return Codes (s_transtatus) */
/*
***** */
#define INVALID_ITEM 100
#define TRAN_OK 0
#define FATAL_SQLERROR -1
/*
***** */
/* Definition of Unused and Bad Items */
/*
***** */
/* Define unused item ID to be 0. This allows the SUT to determine the
/* number of items in the order as required by 2.4.1.3 and 2.4.2.2 since
/* the assumption that any item with OL_I_ID = 0 is unused will be true.
/* This in turn requires that the value used for an invalid item is */
```

```

/* equal to ITEMS + 1. */
/*
*****
***** */
#define INVALID_ITEM_ID (2 * ITEMS) + 1
#define UNUSED_ITEM_ID 0
#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES
/*****
*****/
/* NURand Constants */
/* C_C_LAST_RUN and C_C_LAST_LOAD must adhere to clause 2.1.6.
*/
/* Analysis indicates that a C_LAST delta of 85 is optimal. */
/*****
*****/
#define C_C_LAST_RUN 88
#define C_C_LAST_LOAD 173
#define C_C_ID 319
#define C_OL_I_ID 3849
#define A_C_LAST 255
#define A_C_ID 1023
#define A_OL_I_ID 8191
/*****
*****/
/* Transaction Type Identifiers */
/*****
*****/
#define CLIENT_SQL 0
#define NEWORD_SQL 1
#define PAYMENT_SQL 2
#define ORDSTAT_SQL 3
#define DELIVERY_SQL 4
#define STOCKLEV_SQL 5
#define SPGENERAL_PAD 3
#define SPGENERAL_ADJUST sizeof(int16_t)
struct in_neword_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    struct in_items_struct {
        int32_t s_OL_I_ID;
        int32_t s_OL_SUPPLY_W_ID;
        int16_t s_OL_QUANTITY;
        int16_t pad1[3];
    } in_item[15];
    int64_t s_O_ENTRY_D_time; /* init by SUT */
    int32_t s_C_ID;
    int32_t s_W_ID;
    int16_t s_D_ID;
    int16_t s_O_OL_CNT; /* init by SUT */
    int16_t s_all_local;
    int16_t duplicate_items;
};
struct out_neword_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    struct items_struct {
        int32_t s_I_PRICE;
        int32_t s_OL_AMOUNT;
        int16_t s_S_QUANTITY;
        int16_t pad2;
        char s_I_NAME[25];
        char s_brand_generic;
    } item[15];
    int64_t s_O_ENTRY_D_time;
    int32_t s_W_TAX;
    int32_t s_D_TAX;
    int32_t s_C_DISCOUNT;
    int32_t s_total_amount;
    int32_t s_O_ID;
    int16_t s_O_OL_CNT;
    int16_t s_transtatus;
    int16_t deadlocks;
    char s_C_LAST[17];
    char s_C_CREDIT[3];
};
struct in_payment_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_H_DATE_time; /* init by SUT */
    int64_t s_H_AMOUNT;
    int32_t s_W_ID;
    int32_t s_C_W_ID;
    int32_t s_C_ID;
    int16_t s_C_D_ID;
    int16_t s_D_ID;
    char s_C_LAST[17];
};
struct out_payment_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_H_DATE_time;
    int64_t s_C_SINCE_time;
    int64_t s_C_CREDIT_LIM;
    int64_t s_C_BALANCE;
    int32_t s_C_DISCOUNT;
    int32_t s_C_ID;
    int16_t s_transtatus;
    int16_t deadlocks;
    char s_W_STREET_1[21];
    char s_W_STREET_2[21];
    char s_W_CITY[21];
    char s_W_STATE[3];
    char s_W_ZIP[10];
    char s_D_STREET_1[21];
    char s_D_STREET_2[21];
    char s_D_CITY[21];
    char s_D_STATE[3];
    char s_D_ZIP[10];
    char s_C_FIRST[17];
    char s_C_MIDDLE[3];
    char s_C_LAST[17];
    char s_C_STREET_1[21];
    char s_C_STREET_2[21];
    char s_C_CITY[21];
    char s_C_STATE[3];
    char s_C_ZIP[10];
    char s_C_PHONE[17];
    char s_C_CREDIT[3];
    char s_C_DATA[201];
};
struct in_ordstat_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_C_ID;
    int32_t s_W_ID;
    int16_t s_D_ID;
    int16_t pad1[3];
    char s_C_LAST[17];
};
struct out_ordstat_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_C_BALANCE;
    int64_t s_O_ENTRY_D_time;
    int32_t s_C_ID;

```

```

int32_t s_O_ID;
int16_t s_O_CARRIER_ID;
int16_t s_ol_cnt;
int16_t pad1[2];
struct oitems_struct {
    int64_t s_OL_DELIVERY_D_time;
    int32_t s_OL_AMOUNT;
    int32_t s_OL_I_ID;
    int32_t s_OL_SUPPLY_W_ID;
    int16_t s_OL_QUANTITY;
    int16_t pad2;
} item[15];
int16_t s_transtatus;
int16_t deadlocks;
char s_C_FIRST[17];
char s_C_MIDDLE[3];
char s_C_LAST[17];
};
struct in_delivery_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_O_DELIVERY_D_time; /* init by SUT */
    int32_t s_W_ID;
    int16_t s_O_CARRIER_ID;
};
struct out_delivery_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_O_ID[10];
    int16_t s_transtatus;
    int16_t deadlocks;
};
struct in_stocklev_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_threshold;
    int32_t s_W_ID;
    int16_t s_D_ID;
};
struct out_stocklev_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_low_stock;
    int16_t s_transtatus;
    int16_t deadlocks;
};
/*
*****
***** */
/* Transaction Prototypes */
/*
*****
***** */
#ifdef __cplusplus
extern "C" {
#endif
extern int neword_sql(struct in_neword_struct*, struct out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct out_stocklev_struct*);
#ifdef __cplusplus
}
#endif
/*
*****
***** */

```

```

/* DB2 Connect/Disconnect & Thread Context Wrappers */
/*
***** */
***** */
#ifdef __cplusplus
extern "C" {
#endif
extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*, char*, char*);
extern int disconnect_from_TM(void);

#ifdef __cplusplus
}
#endif
#endif // __DB2TPCC_H

include/lval.h

#ifdef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 12600
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900
#endif // __LVAL_H

include/tpccapp.h

/*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****
***** */
/*
* tpccapp.h - Application Macros
*/
#ifdef __TPCCAPP_H
#define __TPCCAPP_H
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <time.h>
#define daricall
#include "sqlca.h"
#include "sqlcodes.h"
#ifdef SWAP_ENDIAN
#define SWAP_BYTE(Var) SwapEndian((void*)&Var, sizeof(Var))
/*****
*****
FUNCTION: SwapEndian
PURPOSE: Swap the byte order of a structure
EXAMPLE: int I=0x12345678; SWAP_BYTE(I); I => 0x78563412;
IMPLEMENTATION: Fold Addr in half, swap header & tail by XOR op

```

```

e.g.: *a = 0x12 [ Addr + 0];
      *b = 0x78 [ Addr + 4 - 0 - 1 = Addr+3];
      *a ^= *b; // sets *a to 0x6A
      *b ^= *a; // sets *b to 0x12
      *a ^= *b; // sets *a to 0x78
      Now *a => 0x78 && *b => 0x12
*****
*****/
void SwapEndian(void *Addr, int nb)
{
    int i;
    for (i=0; i<nb/2; i++)
    {
        char *a = (char*)Addr+i;
        char *b = (char*)Addr+(nb-i-1);
        *a ^= *b;
        *b ^= *a;
        *a ^= *b;
    }
}
#endif //SWAP_ENDIAN

/*****
*****/
/* SQLCODE Macros */
/*****
*****/
#define DLCHK(a) \
    if (sqlca.sqlcode == SQL_RC_E911) { goto a; }
#define NACOMPCHK(last) \
    if (sqlca.sqlcode != SQL_RC_E1339) { last = -1; } \
    else { int a = ((sqlca.sqlerrmc[4] == 0x20) ? 0 : sqlca.sqlerrmc[4]-0x30); \
           int b = ((sqlca.sqlerrmc[5] == 0x20) ? 0 : sqlca.sqlerrmc[5]-0x30); \
           if (b == 0) { last = a; } else { last = a * 10 + b; } \
    }
#endif // __TPCCAPP_H

include/tpccdbg.h

/*****
*****/
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****
*****/
/*
* tpccdbg.h - Debugging Macros
*/
#ifndef __TPCCDBG_H
#define __TPCCDBG_H
#ifdef __cplusplus
extern "C" {
#endif
extern void new_debug (struct out_neword_struct *neword_ptr,
                     struct in_neword_struct *in_neword_ptr,
                     char *msg);
extern void pay_debug (struct out_payment_struct *payment_ptr,
                     struct in_payment_struct *in_payment_ptr,
                     char *msg);
extern void ord_debug (struct out_ordstat_struct *ordstat_ptr,

```

```

                     struct in_ordstat_struct *in_ordstat_ptr,
                     char *msg);
extern void del_debug (struct out_delivery_struct *delivery_ptr,
                     struct in_delivery_struct *in_delivery_ptr,
                     char *msg);
extern void stk_debug (struct out_stocklev_struct *stocklev_ptr,
                     struct in_stocklev_struct *in_stocklev_ptr,
                     char *msg);
extern void new_print (struct out_neword_struct *neword_ptr,
                     struct in_neword_struct *in_neword_ptr,
                     char *filename,
                     char *msg);
extern void pay_print (struct out_payment_struct *payment_ptr,
                     struct in_payment_struct *in_payment_ptr,
                     char *filename,
                     char *msg);
extern void ord_print (struct out_ordstat_struct *ordstat_ptr,
                     struct in_ordstat_struct *in_ordstat_ptr,
                     char *filename,
                     char *msg);
extern void del_print (struct out_delivery_struct *delivery_ptr,
                     struct in_delivery_struct *in_delivery_ptr,
                     char *filename,
                     char *msg);
extern void stk_print (struct out_stocklev_struct *stocklev_ptr,
                     struct in_stocklev_struct *in_stocklev_ptr,
                     char *filename,
                     char *msg);
#endif __cplusplus
}
#endif
#endif // __TPCCDBG_H

```

Src.Common/Makefile

```

#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#####
#####
#
# Makefile - Makefile for Src.Common
#
!include $(TPCC_ROOT)/Makefile.config

#
#####
#####
# Preprocessor, Compiler and LInker Flags
#
#####
#####
BND_OPTS = GRANT PUBLIC \
           MESSAGES $*.bnd.msg
PRP_OPTS = BINDFILE \
           OPTLEVEL 1 \

```

```

ISOLATION RR \
MESSAGES $*.prep.msg \
LEVEL $(TPCC_VERSION)\
NOLINEMACRO

INCLUDES =      -I$(TPCC_SQLLIB)$(SLASH)include
                -I$(TPCC_ROOT)$(SLASH)include

CFLAGS =        $(CFLAGS_OS) $(CFLAGS_DEBUG) $(INCLUDES) \
                -DSQLA_NOLINES -D$(DB2EDITION)
                -D$(DB2VERSION) \
                -D$(TPCC_SPTYPE)

UTIL_OBJ =      tpcmisc$(OBJEXT) tpcdbg$(OBJEXT)
tpccctx$(OBJEXT)

#
#####
# User Targets
#
#####

all:            connect $(UTIL_OBJ) disconnect

clean:
                - $(ERASE) *$(OBJEXT) *.bnd *.msg tpcctx.c

#
#####
# Helper Targets
#
#####

connect:
                - db2 connect to $(TPCC_DBNAME)

disconnect:
                - db2 connect reset
                - db2 terminate

rebind:
                db2 bind tpcctx.bnd $(BND_OPTS)

#
#####
# Build Rules
#
#####

.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc

.sqc.c:
                @echo "Prepping $*.sqc"
                -db2 prep $*.sqc $(PRP_OPTS)
                @echo "Binding $*.bnd"
                db2 bind $*.bnd $(BND_OPTS)

#
#####
# Dependencies

```

```

#
#####
# Source
tpccdbg$(OBJEXT): tpcdbg.c
tpccctx$(OBJEXT): tpcctx.c
tpcmisc$(OBJEXT): tpcmisc.c

# Headers
tpccdbg.c: $(TPCC_ROOT)/include/db2tpcc.h

Src.Common/tpccctx.sqc

/*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****
*****/
/*
 * tpcctx.sqc - TPCC context code
 */
#include <stdlib.h>
#include <stdio.h>
#include <sqlutil.h>
#include "db2tpcc.h"
int connect_to_TM(char *in_dbname);
int connect_to_TM_auth(char *in_dbname, char *in_username, char
*in_password);
int disconnect_from_TM(void);
int connect_to_TM(char *in_dbname)
{
    return connect_to_TM_auth(in_dbname, "", "");
}
int connect_to_TM_auth(char *in_dbname, char *in_username, char
*in_password)
{
    SQL_STRUCTURE sqlca sqlca;
    int ConnectSQLCODE = 0;
    EXEC SQL BEGIN DECLARE SECTION;
    char dbname[9];
    char username[129];
    char password[15];
    EXEC SQL END DECLARE SECTION;
    strncpy(dbname,in_dbname,8);
    if (strcmp(in_username,"") == 0)
    {
        EXEC SQL CONNECT TO :dbname IN SHARE MODE;
    } else {
        strncpy(username,in_username,128);
        strncpy(password,in_password,14);
        EXEC SQL CONNECT TO :dbname IN SHARE MODE USER :username
        USING :password;
    }
    ConnectSQLCODE = SQLCODE;
    if (ConnectSQLCODE != 0)
    {
        sqlerror( CLIENT_SQL, "CONNECT", __FILE__, __LINE__, &sqlca);
        return ConnectSQLCODE;
    }
}

```



```

    }
    return 0;
}
int disconnect_from_TM(void)
{
    SQL_STRUCTURE sqlca sqlca;
    int DisconnectSQLCODE = 0;
    EXEC SQL CONNECT RESET;
    DisconnectSQLCODE = SQLCODE;
    if (DisconnectSQLCODE != 0) {
        sqlerror( CLIENT_SQL, "DISCONNECT", __FILE__, __LINE__, &sqlca);
    }
    if (DisconnectSQLCODE) {
        return DisconnectSQLCODE;
    }
    return 0;
}

```

Src.Common/tpccdbg.c

```

/*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****/
/*
 * tcdbg.c - Debugging Routines
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <time.h>
#include "sqlca.h"
#include "sql.h"
#include "db2tpcc.h"
#include "tpccdbg.h"
#define DEBUG_FILENAME_SZ 128
#define DEBUG_PATH_SIZE 128
void del_print();
void new_print();
void ord_print();
void pay_print();
void stk_print();
void current_tmstamp(char *buf);
static int debugInit = 0;
static char debugPath[DEBUG_PATH_SIZE] = "";
/*-----*/
/* InitializeDebug */
/*-----*/
__inline void InitializeDebug(void) {
    if (debugInit == 0) {
        char *p = getenv("TPCC_DEBUGDIR");
        if (p) {
            strncpy(debugPath, p, DEBUG_PATH_SIZE);
        } else {
            strcpy(debugPath, "/tmp");
        }
        strcat(debugPath, "/");
    }
}

```

```

    debugInit = 1;
}
/*-----*/
/* sqlerror */
/*-----*/
void sqlerror(int tranType, char *msg, char *file, int line, SQL_STRUCTURE
sqlca *psqlca)
{
    FILE *err_fp = NULL;
    char err_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];
    char tranName[16];
    int j,k;
    char timeStamp[27];
    char errStr[512] = "";
    InitializeDebug();
    strncpy(err_fn, debugPath, DEBUG_PATH_SIZE);
    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;
    switch(tranType)
    {
        case NEWORD_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "new.err.out");
            strcpy(tranName, "NEW_ORDER");
            break;
        case DELIVERY_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "del.err.out");
            strcpy(tranName, "DELIVERY");
            break;
        case PAYMENT_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "pay.err.out");
            strcpy(tranName, "PAYMENT");
            break;
        case ORDSTAT_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "ord.err.out");
            strcpy(tranName, "ORDER_STAT");
            break;
        case STOCKLEV_SQL:
            //sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "stk.err.out");
            strcpy(tranName, "STOCK_LVL");
            break;
        case 0:
            strcat(err_fn, "cli.err.out");
            strcpy(tranName, "CLIENT");
            break;
        default:
            return;
    }
    /* Generate Formatted Error Message */
    sqlaintp(errStr, 512, 78, psqlca);
    err_fp = fopen(err_fn, "a+");
    fprintf(err_fp, "-----\n");
    fprintf(err_fp, "Transaction: %s (%s)\n", tranName, msg);
    fprintf(err_fp, "FILE %s (%u)\n", file, line);
    fprintf(err_fp, "SQLCODE %d ", psqlca->sqlcode);
    fprintf(err_fp, "PID %d ", getpid());
    fprintf(err_fp, "TIME %s\n", timeStamp);
    fprintf(err_fp, "-----\n");
    fprintf(err_fp, "%s", errStr);
    fprintf(err_fp, "-----\n");
    if (psqlca->sqlerrmc[0] != ' ' || psqlca->sqlerrmc[1] != ' ')
    {
        fprintf(err_fp, "slerrmc: ");
        for(j = 0; j < 5; j++)

```

```

{
for(k=0; k < 16; k++) {
int pos = j * 16 + k;
if (pos < 70) fprintf(err_fp, "%02x ", psqlca->sqlerrmc[pos]);
else fprintf(err_fp, " ");
}
fprintf(err_fp, " |");
for(k=0; k < 16; k++) {
int pos = j * 16 + k;
char c = ' ';
if (pos < 70) {
c = psqlca->sqlerrmc[pos];
if (!isprint(c)) c = ' ';
}
fprintf(err_fp, "%c", c);
}
fprintf(err_fp, "\n");
if (j < 4) fprintf(err_fp, "      ");
}
}
fprintf(err_fp, "sqlerrp: ");
for(j=0; j < 8; j++)
fprintf(err_fp, "%c", psqlca->sqlerrp[j]);
fprintf(err_fp, "\n");
fprintf(err_fp, "sqlerrd: ");
for(j=0; j < 6; j++)
fprintf(err_fp, " %d", psqlca->sqlerrd[j]);
fprintf(err_fp, "\n");
if (psqlca->sqlwarn[0] != ' ')
{
fprintf(err_fp, "sqlwarn: ");
for(j=0; j < 8; j++)
fprintf(err_fp, "%c ", psqlca->sqlwarn[j]);
fprintf(err_fp, "\n");
}
fprintf(err_fp, "\n");
fclose(err_fp);
}

/*-----*/
/* del_debug */
/*-----*/
void del_debug (struct out_delivery_struct *delivery_ptr,
struct in_delivery_struct *in_delivery,
char *msg)
{
char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

InitializeDebug();
strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
strcat(debug_fn, "del.debug.out");
del_print(delivery_ptr, in_delivery, debug_fn, msg);
}

/*-----*/
/* del_print */
/*-----*/
void del_print (struct out_delivery_struct *delivery_ptr,
struct in_delivery_struct *in_delivery,
char *filename,
char *msg)
{
FILE *debug_fp;
char timeStamp[27];
int j;
current_tmstamp(&timeStamp[0]);
timeStamp[19] = (char)NULL;
if ((debug_fp = fopen(filename, "a+")) == NULL)
{
}

```

```

return;
}
fprintf(debug_fp, "Delivery debug information follows %s (%s)\n",
timeStamp, msg);
fprintf(debug_fp, " PID %d ", getpid());

fprintf(debug_fp, "\n=====
=====");
fprintf(debug_fp, "in_delivery_struct {\n");
fprintf(debug_fp, "\ts_W_ID = %d (%X)\n",
in_delivery->s_W_ID, in_delivery->s_W_ID);
fprintf(debug_fp, "\ts_O_CARRIER_ID = %d (%X)\n",
in_delivery->s_O_CARRIER_ID, in_delivery->s_O_CARRIER_ID);
fprintf(debug_fp, "\ts_O_DELIVERY_D = %lld (%lX)\n",
in_delivery->s_O_DELIVERY_D_time,
in_delivery->s_O_DELIVERY_D_time);
fprintf(debug_fp, "}\n");
fprintf(debug_fp, "out_delivery_struct {\n");
fprintf(debug_fp, "\ts_transtatus = %d (%X)\n",
delivery_ptr->s_transtatus, delivery_ptr->s_transtatus);
fprintf(debug_fp, "\tdeadlocks = %d (%X)\n",
delivery_ptr->deadlocks, delivery_ptr->deadlocks);
for (j = 0; j < 10; j++) {
fprintf(debug_fp, "\ts_O_ID[%d] = %d\n",
j, delivery_ptr->s_O_ID[j]);
}
fprintf(debug_fp, "\t}\n");
fclose(debug_fp);
}

/*-----*/
/* new_debug */
/*-----*/
void new_debug (struct out_neword_struct *neword_ptr,
struct in_neword_struct *in_neword,
char *msg)
{
char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

InitializeDebug();
strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
strcat(debug_fn, "new.debug.out");
new_print(neword_ptr, in_neword, debug_fn, msg);
}

/*-----*/
/* new_print */
/*-----*/
void new_print (struct out_neword_struct *neword_ptr,
struct in_neword_struct *in_neword,
char *filename,
char *msg)
{
FILE *debug_fp;
char timeStamp[27];
int j, items;
current_tmstamp(&timeStamp[0]);
timeStamp[19] = (char)NULL;
if ((debug_fp = fopen(filename, "a+")) == NULL)
{
return;
}
fprintf(debug_fp, "New order debug information follows %s (%s)\n",
timeStamp, msg);
fprintf(debug_fp, " PID %d ", getpid());

fprintf(debug_fp, "\n=====
=====");
fprintf(debug_fp, "in_neword_struct {\n");

```

```

fprintf(debug_fp, "ts_C_ID = %d (%X)\n",
        in_neword->s_C_ID, in_neword->s_C_ID);
fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
        in_neword->s_W_ID, in_neword->s_W_ID);
fprintf(debug_fp, "ts_D_ID = %d (%X)\n",
        in_neword->s_D_ID, in_neword->s_D_ID);
fprintf(debug_fp, "ts_O_OL_CNT = %d (%X)\n",
        in_neword->s_O_OL_CNT, in_neword->s_O_OL_CNT);
fprintf(debug_fp, "ts_all_local = %d (%X)\n",
        in_neword->s_all_local, in_neword->s_all_local);
fprintf(debug_fp, "ts_O_ENTRY_D = %lld (%lX)\n",
        in_neword->s_O_ENTRY_D_time, in_neword->s_O_ENTRY_D_time);
// fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
//         in_neword->s_transtatus, in_neword->s_transtatus);
// fprintf(debug_fp, "tduplicate_items= %d (%X)\n",
//         in_neword->duplicate_items, in_neword->duplicate_items);
fprintf(debug_fp, "titems {\n");
items = in_neword->s_O_OL_CNT;
for (j=0; j<items; j++) {
    if(j != 0)
        fprintf(debug_fp, "\n");
    fprintf(debug_fp, "ts_OL_I_ID[%d] = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_I_ID,
            in_neword->in_item[j].s_OL_I_ID);
    fprintf(debug_fp, "ts_OL_SUPPLY_W_ID[%d] = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_SUPPLY_W_ID,
            in_neword->in_item[j].s_OL_SUPPLY_W_ID);
    fprintf(debug_fp, "ts_OL_QUANTITY[%d] = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_QUANTITY,
            in_neword->in_item[j].s_OL_QUANTITY);
}
fprintf(debug_fp, "\n}\n");
fprintf(debug_fp, "out_neword_struct {\n");
fprintf(debug_fp, "ts_C_LAST = %s\n",
        neword_ptr->s_C_LAST);
fprintf(debug_fp, "ts_C_CREDIT = %s\n",
        neword_ptr->s_C_CREDIT);
fprintf(debug_fp, "ts_W_TAX = %d\n",
        neword_ptr->s_W_TAX);
fprintf(debug_fp, "ts_D_TAX = %d\n",
        neword_ptr->s_D_TAX);
fprintf(debug_fp, "ts_C_DISCOUNT = %d\n",
        neword_ptr->s_C_DISCOUNT);
fprintf(debug_fp, "ts_O_ID = %d (%X)\n",
        neword_ptr->s_O_ID, neword_ptr->s_O_ID);
fprintf(debug_fp, "ts_O_OL_CNT = %d (%X)\n",
        neword_ptr->s_O_OL_CNT, neword_ptr->s_O_OL_CNT);
fprintf(debug_fp, "ts_O_ENTRY_D = %lld (%lX)\n",
        neword_ptr->s_O_ENTRY_D_time,
        neword_ptr->s_O_ENTRY_D_time);
neword_ptr->s_O_ENTRY_D_time);
fprintf(debug_fp, "ts_total_amount = %d\n",
        neword_ptr->s_total_amount);
fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
        neword_ptr->s_transtatus, neword_ptr->s_transtatus);
fprintf(debug_fp, "tdeadlocks = %d (%X)\n",
        neword_ptr->deadlocks, neword_ptr->deadlocks);
// fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
//         neword_ptr->s_W_ID, neword_ptr->s_W_ID);
// fprintf(debug_fp, "ts_D_ID = %d (%X)\n",
//         neword_ptr->s_D_ID, neword_ptr->s_D_ID);
// fprintf(debug_fp, "ts_all_local = %d (%X)\n",
//         neword_ptr->s_all_local, neword_ptr->s_all_local);
// fprintf(debug_fp, "tduplicate_items= %d (%X)\n",
//         neword_ptr->duplicate_items, neword_ptr->duplicate_items);
fprintf(debug_fp, "titems {\n");
items = neword_ptr->s_O_OL_CNT;
for (j=0; j<items; j++) {
    if(j != 0)

```

```

fprintf(debug_fp, "\n");
fprintf(debug_fp, "ts_I_NAME[%d] = %s\n",
        j, neword_ptr->item[j].s_I_NAME);
fprintf(debug_fp, "ts_I_PRICE[%d] = %d\n",
        j, neword_ptr->item[j].s_I_PRICE);
fprintf(debug_fp, "ts_OL_AMOUNT[%d] = %d\n",
        j, neword_ptr->item[j].s_OL_AMOUNT);
fprintf(debug_fp, "ts_S_QUANTITY[%d] = %d (%X)\n",
        j, neword_ptr->item[j].s_S_QUANTITY,
        neword_ptr->item[j].s_S_QUANTITY);
fprintf(debug_fp, "ts_brand_generic[%d] = %c\n",
        j, neword_ptr->item[j].s_brand_generic);
}
fprintf(debug_fp, "\n}\n");
fclose(debug_fp);
}

/*-----*/
/* ord_debug */
/*-----*/
void ord_debug (struct out_ordstat_struct *ordstat_ptr,
               struct in_ordstat_struct *in_ordstat,
               char *msg)
{
    char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
    strcat(debug_fn, "ord.debug.out");
    ord_print(ordstat_ptr, in_ordstat, debug_fn, msg);
}

/*-----*/
/* ord_print */
/*-----*/
void ord_print (struct out_ordstat_struct *ordstat_ptr,
               struct in_ordstat_struct *in_ordstat,
               char *filename,
               char *msg)
{
    FILE *debug_fp;
    char timeStamp[27];
    int j, items;
    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;
    if ((debug_fp = fopen(filename, "a+")) == NULL)
    {
        return;
    }
    fprintf(debug_fp, "Order status debug information follows %s (%s)\n",
            timeStamp, msg);
    fprintf(debug_fp, " PID %d ", getpid());

    fprintf(debug_fp, "\n=====
    fprintf(debug_fp, "in_ordstat_struct {\n");
    fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
            in_ordstat->s_W_ID, in_ordstat->s_W_ID);
    fprintf(debug_fp, "ts_D_ID = %d (%X)\n",
            in_ordstat->s_D_ID, in_ordstat->s_D_ID);
    fprintf(debug_fp, "ts_C_ID = %d (%X)\n",
            in_ordstat->s_C_ID, in_ordstat->s_C_ID);
    fprintf(debug_fp, "ts_C_LAST = %s\n",
            in_ordstat->s_C_LAST);
    fprintf(debug_fp, "}\n");
    fprintf(debug_fp, "out_ordstat_struct {\n");
    fprintf(debug_fp, "ts_C_ID = %d (%X)\n",
            ordstat_ptr->s_C_ID, ordstat_ptr->s_C_ID);

```



```

fprintf(debug_fp,"ts_C_STREET_2 = %s\n",
        payment_ptr->s_C_STREET_2);
fprintf(debug_fp,"ts_C_CITY = %s\n",
        payment_ptr->s_C_CITY);
fprintf(debug_fp,"ts_C_STATE = %s\n",
        payment_ptr->s_C_STATE);
fprintf(debug_fp,"ts_C_ZIP = %s\n",
        payment_ptr->s_C_ZIP);
fprintf(debug_fp,"ts_C_PHONE = %s\n",
        payment_ptr->s_C_PHONE);
fprintf(debug_fp,"ts_C_SINCE = %lld (%lX)\n",
        payment_ptr->s_C_SINCE_time, payment_ptr->s_C_SINCE_time);
fprintf(debug_fp,"ts_C_CREDIT = %s\n",
        payment_ptr->s_C_CREDIT);
fprintf(debug_fp,"ts_C_DATA = %s\n",
        payment_ptr->s_C_DATA);
fprintf(debug_fp,"ts_transtatus = %d (%X)\n",
        payment_ptr->s_transtatus, payment_ptr->s_transtatus);
fprintf(debug_fp,"tdeadlocks = %d (%X)\n",
        payment_ptr->deadlocks, payment_ptr->deadlocks);
fprintf(debug_fp,"}\n\n");
fclose(debug_fp);
}

/*-----*/
/* stk_debug */
/*-----*/
void stk_debug (struct out_stocklev_struct *stocklev,
               struct in_stocklev_struct *in_stocklev,
               char *msg)
{
    char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
    strcat(debug_fn, "stk.debug.out");
    stk_print(stocklev, in_stocklev, debug_fn, msg);
}

/*-----*/
/* stk_print */
/*-----*/
void stk_print (struct out_stocklev_struct *stocklev,
               struct in_stocklev_struct *in_stocklev,
               char *filename,
               char *msg)
{
    FILE *debug_fp;
    char timeStamp[27];
    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;
    if ((debug_fp = fopen(filename, "a+")) == NULL)
    {
        return;
    }
    fprintf(debug_fp, "Stock level debug information follows %s (%s)\n",
            timeStamp, msg);
    fprintf(debug_fp, " PID %d ", getpid());

    fprintf(debug_fp, "\n=====
=====");
    fprintf(debug_fp, "in_stocklev_struct {\n");
    fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
            in_stocklev->s_W_ID, in_stocklev->s_W_ID);
    fprintf(debug_fp, "ts_D_ID = %d (%X)\n",
            in_stocklev->s_D_ID, in_stocklev->s_D_ID);
    fprintf(debug_fp, "ts_threshold = %d (%X)\n",
            in_stocklev->s_threshold, in_stocklev->s_threshold);
    fprintf(debug_fp, "}\n\n");
}

```

```

fprintf(debug_fp, "out_stocklev_struct {\n");
fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
        stocklev->s_transtatus, stocklev->s_transtatus);
fprintf(debug_fp, "tdeadlocks = %d (%X)\n",
        stocklev->deadlocks, stocklev->deadlocks);
fprintf(debug_fp, "ts_low_stock = %d (%X)\n",
        stocklev->s_low_stock, stocklev->s_low_stock);
fprintf(debug_fp, "}\n\n");
fclose(debug_fp);
}
void current_tmstamp(char *buf)
{
    time_t t = time(NULL);
    strncpy(buf, ctime(&t), 19);
}

```

Src.Common/tpccmisc.c

```

/*-----*/
/*-----*/
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
/*-----*/
/*-----*/
/*
 * tpccmisc.c - Miscellaneous routines
 */
#include <stdlib.h>
#include <sys/types.h>
#include <sys/time.h>
double current_time_ms(void);
double current_time(void);

/* Current time in SECONDS, precision SECONDS */
double current_time(void)
{
    /* use time() to get seconds */
    return(time(NULL));
}

/* Current time in SECONDS, precision MILLISECONDS */
double current_time_ms(void)
{
    /* gettimeofday() returns seconds and microseconds */
    /* convert to fractional seconds */
    struct timeval t;
    gettimeofday(&t, NULL);
    return (t.tv_sec + (double)t.tv_usec/(1000*1000));
}

```

Src.Srv/Makefile

```

#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
## All Rights Reserved.
##

```

```

## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#####

#
# Makefile - Makefile for Src.Srv
#

!include $(TPCC_ROOT)/Makefile.config

#
#####
# Preprocessor, Compiler and Linker Flags
#
#####

BND_OPTS =      GRANT PUBLIC \
                MESSAGES $*.bnd.msg
PRP_OPTS =      BINDFILE \
                EXPLAIN ALL \
                MESSAGES $*.prep.msg

INCLUDES =      -I$(TPCC_SQLLIB)$(SLASH)include
                -I$(TPCC_ROOT)$(SLASH)include

CFLAGS =        $(CFLAGS_OS) $(INCLUDES) $(CFLAGS_DEBUG) \
                -D$(DB2EDITION) -D$(DB2VERSION) \
                -DSQLA_NOLINES -DLINT_ARGS

LDFLAGS =       $(LDFLAGS_STORP) $(LDFLAGS_LIB)

#
#####
# File Collections
#
#####

STORED_PROCS = new ord del

UTIL_OBJ =      $(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT) \
                $(TPCC_ROOT)/Src.Common/tpccdbg$(OBJEXT)

DLL =           rpctpc$(SHLIBEXT)

#
#####
# User Targets
#
#####

all:            connect explain catalog $(DLL) install plan disconnect

clean:          connect uncatalog unexplain disconnect
                - $(ERASE) $(TPCC_SPDIR)$(SLASH)rpctpc$(SHLIBEXT)
                - $(ERASE) *.bnd *.msg *.out *$(OBJEXT) $(DLL) tpcc_all_sql.c
                - $(ERASE) TPCC_ALL.*.plan

#
#####
# Helper Targets
#####

#
#####
#
#####
catalog:        uncatalog
                - perl $(TPCC_ROOT)$(SLASH)utils$(SLASH)genproc.pl
                $(STORED_PROCS)
                - db2 -tvf cat-proc.ddl +o -z cat-proc.out
                - db2 -td$$ -vf cat-func.ddl +o -z cat-func.out

uncatalog:
                - perl $(TPCC_ROOT)$(SLASH)utils$(SLASH)genproc.pl
                $(STORED_PROCS)
                - db2 -td$$ -vf uncat-func.ddl +o -z uncat-func.out
                - db2 -tvf uncat-proc.ddl +o -z uncat-proc.out

explain:
                - perl $(TPCC_ROOT)$(SLASH)utils$(SLASH)fixup_explain.pl
                - db2 -tvf
                $(TPCC_ROOT)$(SLASH)utils$(SLASH)EXPLAIN.DDL +o -z EXPLAIN.out

unexplain:
                - db2 -tvf
                $(TPCC_ROOT)$(SLASH)utils$(SLASH)UNEXPLAIN.DDL +o -z
                UNEXPLAIN.out

connect:
                - db2 connect to $(TPCC_DBNAME)

disconnect:
                - db2 connect reset
                - db2 terminate

# This (environment) variable is used by db2expln
DB2EXPLN_BUFFER=3000000

plan:
                - db2exfmt -d $(TPCC_DBNAME) -e $(TPCC_SCHEMA) -s
                $(TPCC_SCHEMA) -w -1 -n TPCC_ALL -g -# 0 -o TPCC_ALL.exfmt.plan
                - db2expln -d $(TPCC_DBNAME) -c $(TPCC_SCHEMA) -p
                TPCC_ALL -s 0 -g -o TPCC_ALL.expln.plan

rebind:         connect catalog
                db2 bind tpcc_all_sql.bnd $(BND_OPTS) QUERYOPT 7

#
#####
#####
# Install Targets
#
#####

install:        $(DLL)
                - mkdir $(TPCC_SPDIR)
                $(COPY) $(DLL) $(TPCC_SPDIR)

#
#####
#####
# Build Rules
#
#####

.SUFFIXES: $(OBJEXT) .c .sqc

# d230437mte: QUERYOPT 7 required for UNION ALL

```

```

# Only stock needs CS , and that can be specified on the SELECT statement
tpcc_all_sql.c:
    @echo "Prepping $*.sql"
    -db2 prep $*.sql $(PRP_OPTS) ISOLATION RR
    @echo "Binding $*.bnd"
    db2 bind $*.bnd $(BND_OPTS) QUERYOPT 7

# Stored procedures are built in a special way

tpcc_all_sql$(OBJEXT):
    $(CC) -c tpcc_all_sql.c $(CFLAGS) -D$(TPCC_SPTYPE)
$(CFLAGS_OUT)$@

$(DLL): $(UTIL_OBJ) tpcc_all_sql$(OBJEXT)
    $(LD_STORP) $(LD_FLAGS) $(UTIL_OBJ)
tpcc_all_sql$(OBJEXT) $(LD_FLAGS_OUT)$@

#
#####
#####
# Dependencies
#
#####
#####

# Executables (Stored Procedures)
$(DLL): $(UTIL_OBJ) tpcc_all_sql$(OBJEXT)

# Source
tpcc_all_sql$(OBJEXT): tpcc_all_sql.c

# Headers
tpcc_all_sql.c:    $(TPCC_ROOT)/include/db2tpcc.h

```

Src.Srv/cat-func.ddl

```

-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
-----
--
-- cat-func.ddl - Create table functions
--
--
-- DELIVERY
--
CREATE FUNCTION DEL( W_ID      INTEGER
                  ,D_ID      SMALLINT
                  ,CARRIER_ID SMALLINT
                  ,DELIVERY_D BIGINT
                  )
RETURNS TABLE ( O_ID INTEGER )
SPECIFIC DELIVERY
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION
LANGUAGE SQL
VAR: BEGIN ATOMIC
  DECLARE O_ID INTEGER ;
  DECLARE C_ID INTEGER ;
  DECLARE AMOUNT INTEGER ;

```

```

/* Delete the order from new order table */
SET VAR.O_ID = ( SELECT NO_O_ID
                FROM OLD TABLE ( DELETE
                                FROM ( SELECT NO_O_ID
                                      FROM NEW_ORDER
                                      WHERE NO_W_ID = DEL.W_ID
                                      AND NO_D_ID = DEL.D_ID
                                      ORDER BY NO_O_ID ASC
                                      FETCH FIRST 1 ROW ONLY
                                ) AS NEW_ORDER
                ) AS D
                )
;
/* Update the order as delivered and retrieve the customer id */
SET VAR.C_ID = ( SELECT O_C_ID
                FROM OLD TABLE ( UPDATE ORDERS
                                SET O_CARRIER_ID = DEL.CARRIER_ID
                                WHERE O_W_ID = DEL.W_ID
                                AND O_D_ID = DEL.D_ID
                                AND O_ID = VAR.O_ID
                                ) AS U
                )
;
SET VAR.AMOUNT = ( SELECT SUM( OL_AMOUNT )
                  FROM OLD TABLE ( UPDATE ORDER_LINE
                                SET OL_DELIVERY_D = DEL.DELIVERY_D
                                WHERE OL_W_ID = DEL.W_ID
                                AND OL_D_ID = DEL.D_ID
                                AND OL_O_ID = VAR.O_ID
                                ) AS U
                  )
;
/* Charge the customer */
UPDATE CUSTOMER
  SET C_BALANCE = C_BALANCE + VAR.AMOUNT
  ,C_DELIVERY_CNT = C_DELIVERY_CNT + SMALLINT( 1 )
WHERE C_W_ID = DEL.W_ID
  AND C_D_ID = DEL.D_ID
  AND C_ID = VAR.C_ID
;
/* Return the order id to the caller (or NULL) */
RETURN VALUES VAR.O_ID ;
END
$

--
-- ORDER STATUS
--
CREATE FUNCTION ORD_C_LAST( W_ID INTEGER
                          ,D_ID SMALLINT
                          ,C_LAST VARCHAR(16)
                          )
RETURNS TABLE( O_ID      INTEGER
               ,O_CARRIER_ID SMALLINT
               ,O_ENTRY_D  BIGINT
               ,C_BALANCE  BIGINT
               ,C_FIRST   VARCHAR(16)
               ,C_MIDDLE  CHAR(2)
               ,C_ID      INTEGER
               )
SPECIFIC ORD_C_LAST
READS SQL DATA NO EXTERNAL ACTION DETERMINISTIC
LANGUAGE SQL
VAR: BEGIN ATOMIC
  DECLARE C_BALANCE BIGINT ;
  DECLARE C_FIRST  VARCHAR(16) ;
  DECLARE C_MIDDLE CHAR(2) ;
  DECLARE C_ID    INTEGER ;

```

```

DECLARE O_ID    INTEGER;
DECLARE O_CARRIER_ID SMALLINT;
DECLARE O_ENTRY_D  BIGINT;
/* Retrieve the Customer information */
SET ( C_BALANCE, C_FIRST, C_MIDDLE, C_ID )
= ( SELECT C_BALANCE, C_FIRST, C_MIDDLE, C_ID
    FROM ( SELECT C_ID
           , C_BALANCE
           , C_FIRST
           , C_MIDDLE
           , COUNT(*) OVER() AS COUNT
           , ROWNUMBER() OVER (ORDER BY C_FIRST) AS NUM
    FROM CUSTOMER
    WHERE C_W_ID = ORD_C_LAST.W_ID
    AND C_D_ID = ORD_C_LAST.D_ID
    AND C_LAST = ORD_C_LAST.C_LAST
    ) AS V1
    WHERE NUM = (COUNT + BIGINT(1)) / BIGINT(2)
    )
;
/* Take advantage of the index to fetch the first row (and hence max(o_id) ) */
SET ( O_ID, O_CARRIER_ID, O_ENTRY_D )
= ( SELECT O_ID
    , O_CARRIER_ID
    , O_ENTRY_D
    FROM ORDERS
    WHERE O_W_ID = ORD_C_LAST.W_ID
    AND O_D_ID = ORD_C_LAST.D_ID
    AND O_C_ID = VAR_C_ID
    ORDER BY O_ID DESC
    FETCH FIRST 1 ROW ONLY
    )
;

RETURN VALUES ( VAR.O_ID
    , VAR.O_CARRIER_ID
    , VAR.O_ENTRY_D
    , VAR.C_BALANCE
    , VAR.C_FIRST
    , VAR.C_MIDDLE
    , VAR.C_ID
    )
;

END $
CREATE FUNCTION ORD_C_ID( W_ID INTEGER
    , D_ID SMALLINT
    , C_ID INTEGER
    )
RETURNS TABLE( O_ID    INTEGER
    , O_CARRIER_ID SMALLINT
    , O_ENTRY_D  BIGINT
    , C_BALANCE  BIGINT
    , C_FIRST   VARCHAR(16)
    , C_MIDDLE  CHAR(2)
    , C_LAST    VARCHAR(16)
    )
SPECIFIC ORD_C_ID
READS SQL DATA NO EXTERNAL ACTION DETERMINISTIC
LANGUAGE SQL
VAR: BEGIN ATOMIC
DECLARE C_BALANCE  BIGINT;
DECLARE C_FIRST   VARCHAR(16);
DECLARE C_MIDDLE  CHAR(2);
DECLARE C_LAST    VARCHAR(16);
DECLARE O_ID      INTEGER;
DECLARE O_CARRIER_ID SMALLINT;
DECLARE O_ENTRY_D  BIGINT;
/* Retrieve the Customer information */
SET ( C_BALANCE, C_FIRST, C_MIDDLE, C_ID )
= ( SELECT C_BALANCE, C_FIRST, C_MIDDLE, C_ID
    FROM CUSTOMER
    WHERE C_ID = ORD_C_ID.C_ID
    AND C_W_ID = ORD_C_ID.W_ID
    AND C_D_ID = ORD_C_ID.D_ID
    )
;
SET ( O_ID, O_CARRIER_ID, O_ENTRY_D )
= ( SELECT O_ID
    , O_CARRIER_ID
    , O_ENTRY_D
    FROM ORDERS
    WHERE O_W_ID = ORD_C_ID.W_ID
    AND O_D_ID = ORD_C_ID.D_ID
    AND O_C_ID = ORD_C_ID.C_ID
    ORDER BY O_ID DESC
    FETCH FIRST 1 ROW ONLY
    )
;

RETURN VALUES ( VAR.O_ID
    , VAR.O_CARRIER_ID
    , VAR.O_ENTRY_D
    , VAR.C_BALANCE
    , VAR.C_FIRST
    , VAR.C_MIDDLE
    , VAR.C_LAST
    )
;

END $
--
-- PAYMENT
--
CREATE FUNCTION PAY_C_LAST( W_ID INTEGER
    , D_ID SMALLINT
    , C_W_ID INTEGER
    , C_D_ID SMALLINT
    , C_LAST VARCHAR(16)
    , H_DATE BIGINT
    , H_AMOUNT BIGINT
    , BAD_CREDIT_PREFIX VARCHAR(28)
    )
RETURNS TABLE( W_STREET_1 CHAR(20)
    , W_STREET_2 CHAR(20)
    , W_CITY CHAR(20)
    , W_STATE CHAR(2)
    , W_ZIP CHAR(9)
    , D_STREET_1 CHAR(20)
    , D_STREET_2 CHAR(20)
    , D_CITY CHAR(20)
    , D_STATE CHAR(2)
    , D_ZIP CHAR(9)
    , C_ID INTEGER
    , C_FIRST VARCHAR(16)
    , C_MIDDLE CHAR(2)
    , C_STREET_1 VARCHAR(20)
    , C_STREET_2 VARCHAR(20)
    , C_CITY VARCHAR(20)
    , C_STATE CHAR(2)
    , C_ZIP CHAR(9)
    , C_PHONE CHAR(16)
    , C_SINCE BIGINT
    , C_CREDIT CHAR(2)
    , C_CREDIT_LIM BIGINT
    , C_DISCOUNT INTEGER
    , C_BALANCE BIGINT
    , C_DATA CHAR(200)
    )

```



```

)
SPECIFIC PAY_C_LAST
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION
LANGUAGE SQL
VAR: BEGIN ATOMIC
  DECLARE W_NAME CHAR(10);
  DECLARE D_NAME CHAR(10);
  DECLARE W_STREET_1 CHAR(20);
  DECLARE W_STREET_2 CHAR(20);
  DECLARE W_CITY CHAR(20);
  DECLARE W_STATE CHAR(2);
  DECLARE W_ZIP CHAR(9);
  DECLARE D_STREET_1 CHAR(20);
  DECLARE D_STREET_2 CHAR(20);
  DECLARE D_CITY CHAR(20);
  DECLARE D_STATE CHAR(2);
  DECLARE D_ZIP CHAR(9);
  DECLARE C_ID INTEGER;
  DECLARE C_FIRST VARCHAR(16);
  DECLARE C_MIDDLE CHAR(2);
  DECLARE C_STREET_1 VARCHAR(20);
  DECLARE C_STREET_2 VARCHAR(20);
  DECLARE C_CITY VARCHAR(20);
  DECLARE C_STATE CHAR(2);
  DECLARE C_ZIP CHAR(9);
  DECLARE C_PHONE CHAR(16);
  DECLARE C_SINCE BIGINT;
  DECLARE C_CREDIT CHAR(2);
  DECLARE C_CREDIT_LIM BIGINT;
  DECLARE C_DISCOUNT INTEGER;
  DECLARE C_BALANCE BIGINT;
  DECLARE C_DATA CHAR(200);
  /* Update District and retrieve its data */
  SET ( D_NAME, D_STREET_1, D_STREET_2, D_CITY, D_STATE,
D_ZIP)
    = ( SELECT D_NAME, D_STREET_1, D_STREET_2, D_CITY,
D_STATE, D_ZIP
      FROM OLD TABLE ( UPDATE DISTRICT
        SET D_YTD = D_YTD + PAY_C_LAST.H_AMOUNT
        WHERE D_W_ID = PAY_C_LAST.W_ID
        AND D_ID = PAY_C_LAST.D_ID
      ) AS U
    )
;
/* Determine the C_ID */
SET ( C_ID )
= ( SELECT C_ID
  FROM ( SELECT C_ID
        , COUNT(*) OVER() AS COUNT
        , ROWNUMBER() OVER (ORDER BY C_FIRST) AS NUM
      FROM CUSTOMER
      WHERE C_LAST = PAY_C_LAST.C_LAST
      AND C_W_ID = PAY_C_LAST.C_W_ID
      AND C_D_ID = PAY_C_LAST.C_D_ID
    ) AS T
  WHERE NUM = (COUNT + BIGINT(1)) / BIGINT(2)
)
;
/* Update the middle customer */
SET ( C_ID, 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, C_DATA )
= ( SELECT C_ID, 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

```

```

, CASE WHEN C_CREDIT = 'BC' THEN SUBSTR(C_DATA, 1,
200) ELSE NULL END AS C_DATA
  FROM NEW TABLE ( UPDATE CUSTOMER
    SET C_BALANCE = C_BALANCE -
PAY_C_LAST.H_AMOUNT
, C_YTD_PAYMENT = C_YTD_PAYMENT +
PAY_C_LAST.H_AMOUNT
, C_PAYMENT_CNT = C_PAYMENT_CNT +
SMALLINT(1)
, C_DATA = CASE WHEN C_CREDIT = 'BC'
  THEN CHAR(C_ID) -- 11 bytes long
  ||BAD_CREDIT_PREFIX -- 28 bytes
long
  ||SUBSTR(C_DATA, 1, 461) -- 461 + 39 =
500
  ELSE C_DATA
  END
  WHERE C_W_ID = PAY_C_LAST.C_W_ID
  AND C_D_ID = PAY_C_LAST.C_D_ID
  AND C_ID = VAR.C_ID
) AS U
)
;
/* Update the warehouse */
SET ( W_NAME, W_STREET_1, W_STREET_2, W_CITY, W_STATE,
W_ZIP )
= ( SELECT W_NAME, W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP
  FROM OLD TABLE ( UPDATE WAREHOUSE
    SET W_YTD = W_YTD + PAY_C_LAST.H_AMOUNT
    WHERE W_ID = PAY_C_LAST.W_ID
  ) AS U
)
;
/* Finally insert into the warehouse */
INSERT
  INTO HISTORY ( H_C_ID, H_C_D_ID, H_C_W_ID, H_D_ID, H_W_ID,
H_DATA, H_DATE, H_AMOUNT )
  VALUES ( VAR.C_ID
, PAY_C_LAST.C_D_ID
, PAY_C_LAST.C_W_ID
, PAY_C_LAST.D_ID
, PAY_C_LAST.W_ID
, VAR.W_NAME || CHAR(' ', 4) || VAR.D_NAME
, PAY_C_LAST.H_DATE
, PAY_C_LAST.H_AMOUNT
)
;
/* Done - return the collected data */
RETURN VALUES ( W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP
, D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
, C_ID, 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, C_DATA
)
;
END
$
CREATE FUNCTION PAY_C_ID( W_ID INTEGER
, D_ID SMALLINT
, C_W_ID INTEGER
, C_D_ID SMALLINT
, C_ID INTEGER
, H_DATE BIGINT
, H_AMOUNT BIGINT
, BAD_CREDIT_PREFIX VARCHAR(34)

```

```

)
RETURNS TABLE( W_STREET_1 CHAR(20)
, W_STREET_2 CHAR(20)
, W_CITY CHAR(20)
, W_STATE CHAR(2)
, W_ZIP CHAR(9)
, D_STREET_1 CHAR(20)
, D_STREET_2 CHAR(20)
, D_CITY CHAR(20)
, D_STATE CHAR(2)
, D_ZIP CHAR(9)
, C_LAST VARCHAR(16)
, C_FIRST VARCHAR(16)
, C_MIDDLE CHAR(2)
, C_STREET_1 VARCHAR(20)
, C_STREET_2 VARCHAR(20)
, C_CITY VARCHAR(20)
, C_STATE CHAR(2)
, C_ZIP CHAR(9)
, C_PHONE CHAR(16)
, C_SINCE BIGINT
, C_CREDIT CHAR(2)
, C_CREDIT_LIM BIGINT
, C_DISCOUNT INTEGER
, C_BALANCE BIGINT
, C_DATA CHAR(200)
)
SPECIFIC PAY_C_ID
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION
LANGUAGE SQL
VAR: BEGIN ATOMIC
DECLARE W_NAME CHAR(10);
DECLARE D_NAME CHAR(10);
DECLARE W_STREET_1 CHAR(20);
DECLARE W_STREET_2 CHAR(20);
DECLARE W_CITY CHAR(20);
DECLARE W_STATE CHAR(2);
DECLARE W_ZIP CHAR(9);
DECLARE D_STREET_1 CHAR(20);
DECLARE D_STREET_2 CHAR(20);
DECLARE D_CITY CHAR(20);
DECLARE D_STATE CHAR(2);
DECLARE D_ZIP CHAR(9);
DECLARE C_LAST VARCHAR(16);
DECLARE C_FIRST VARCHAR(16);
DECLARE C_MIDDLE CHAR(2);
DECLARE C_STREET_1 VARCHAR(20);
DECLARE C_STREET_2 VARCHAR(20);
DECLARE C_CITY VARCHAR(20);
DECLARE C_STATE CHAR(2);
DECLARE C_ZIP CHAR(9);
DECLARE C_PHONE CHAR(16);
DECLARE C_SINCE BIGINT;
DECLARE C_CREDIT CHAR(2);
DECLARE C_CREDIT_LIM BIGINT;
DECLARE C_DISCOUNT INTEGER;
DECLARE C_BALANCE BIGINT;
DECLARE C_DATA CHAR(200);

/* Update District and retrieve its data */
SET ( D_NAME, D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
)
= ( SELECT D_NAME, D_STREET_1, D_STREET_2, D_CITY,
D_STATE, D_ZIP
FROM OLD TABLE ( UPDATE DISTRICT
SET D_YTD = D_YTD + PAY_C_ID.H_AMOUNT
WHERE D_W_ID = PAY_C_ID.W_ID
AND D_ID = PAY_C_ID.D_ID

```

```

) AS U
);
/* Update the middle customer */
SET ( C_LAST, 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, C_DATA )
= ( SELECT C_LAST, 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
, CASE WHEN C_CREDIT = 'BC' THEN SUBSTR(C_DATA, 1,
200) ELSE NULL END AS C_DATA
FROM NEW TABLE ( UPDATE CUSTOMER
SET C_BALANCE = C_BALANCE -
PAY_C_ID.H_AMOUNT
, C_YTD_PAYMENT = C_YTD_PAYMENT +
PAY_C_ID.H_AMOUNT
, C_PAYMENT_CNT = C_PAYMENT_CNT +
SMALLINT( 1 )
, C_DATA = CASE WHEN C_CREDIT = 'BC'
THEN BAD_CREDIT_PREFIX -- 34
bytes long ||SUBSTR( C_DATA, 1, 466 ) -- 466 + 34 =
500 bytes ELSE C_DATA
END
WHERE C_W_ID = PAY_C_ID.C_W_ID
AND C_D_ID = PAY_C_ID.C_D_ID
AND C_ID = PAY_C_ID.C_ID
) AS U
);
/* Update the warehouse */
SET ( W_NAME, W_STREET_1, W_STREET_2, W_CITY, W_STATE,
W_ZIP )
= ( SELECT W_NAME, W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP
FROM OLD TABLE ( UPDATE WAREHOUSE
SET W_YTD = W_YTD + PAY_C_ID.H_AMOUNT
WHERE W_ID = PAY_C_ID.W_ID
) AS U
);
/* Finally insert into the warehouse */
INSERT
INTO HISTORY ( H_C_ID, H_C_D_ID, H_C_W_ID, H_D_ID, H_W_ID,
H_DATA, H_DATE, H_AMOUNT )
VALUES ( PAY_C_ID.C_ID
, PAY_C_ID.C_D_ID
, PAY_C_ID.C_W_ID
, PAY_C_ID.D_ID
, PAY_C_ID.W_ID
, VAR.W_NAME || CHAR( ' ', 4 ) || VAR.D_NAME
, PAY_C_ID.H_DATE
, PAY_C_ID.H_AMOUNT
)
);
/* Done - return the collected data */
RETURN VALUES ( W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP
, D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
, C_LAST, 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, C_DATA

```

```

)
;
END
$
--
-- NEW ORDER
--
CREATE FUNCTION NEW_OL_ALL( I_ID INT
, I_QTY SMALLINT
, W_ID INT
, SUPP_W_ID INT
, O_ID INT
, D_ID SMALLINT
)
RETURNS TABLE( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, OL_DIST_INFO CHAR(24)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT
)
SPECIFIC NEW_OL_ALL
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION
LANGUAGE SQL
VAR: BEGIN ATOMIC
DECLARE I_PRICE INTEGER ;
DECLARE I_NAME CHAR(24) ;
DECLARE I_DATA VARCHAR(50) ;
DECLARE OL_DIST_INFO CHAR(24) ;
DECLARE S_DATA VARCHAR(50) ;
DECLARE S_QUANTITY SMALLINT ;

SET ( I_PRICE , I_NAME , I_DATA )
= ( SELECT
I_PRICE
, I_NAME
, I_DATA
FROM ITEM
WHERE ITEM.I_ID = NEW_OL_ALL.I_ID
);
SET ( OL_DIST_INFO , S_DATA , S_QUANTITY )
= ( SELECT OL_DIST_INFO
, S_DATA
, S_QUANTITY
FROM NEW TABLE ( UPDATE STOCK
INCLUDE ( OL_DIST_INFO CHAR( 24 ) )
SET S_QUANTITY = CASE WHEN S_QUANTITY -
NEW_OL_ALL.I_QTY >= 10
THEN S_QUANTITY -
NEW_OL_ALL.I_QTY
ELSE S_QUANTITY -
NEW_OL_ALL.I_QTY + 91
END
, S_ORDER_CNT = S_ORDER_CNT +
SMALLINT( 1 )
, S_YTD = S_YTD + NEW_OL_ALL.I_QTY
, S_REMOTE_CNT = CASE WHEN
NEW_OL_ALL.SUPP_W_ID = NEW_OL_ALL.W_ID
THEN S_REMOTE_CNT
ELSE S_REMOTE_CNT +
SMALLINT( 1 )
END
, OL_DIST_INFO = CASE D_ID WHEN
SMALLINT( 1 ) THEN S_DIST_01
WHEN SMALLINT( 2 )
THEN S_DIST_02
WHEN SMALLINT( 3 )
THEN S_DIST_03

```

```

WHEN SMALLINT( 4 )
THEN S_DIST_04
WHEN SMALLINT( 5 )
THEN S_DIST_05
WHEN SMALLINT( 6 )
THEN S_DIST_06
WHEN SMALLINT( 7 )
THEN S_DIST_07
WHEN SMALLINT( 8 )
THEN S_DIST_08
WHEN SMALLINT( 9 )
THEN S_DIST_09
WHEN SMALLINT( 10 )
THEN S_DIST_10
END
WHERE S_I_ID = NEW_OL_ALL.I_ID
AND S_W_ID = NEW_OL_ALL.SUPP_W_ID
) AS U
)
;
RETURN VALUES( VAR.I_PRICE
, VAR.I_NAME
, VAR.I_DATA
, VAR.OL_DIST_INFO
, VAR.S_DATA
, VAR.S_QUANTITY
)
;
END
$
CREATE FUNCTION NEW_OL_LOCAL( I_ID INT
, I_QTY SMALLINT
, W_ID INT
, O_ID INT
, D_ID SMALLINT
)
RETURNS TABLE( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, OL_DIST_INFO CHAR(24)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT
)
SPECIFIC NEW_OL_LOCAL
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION
LANGUAGE SQL
VAR: BEGIN ATOMIC
DECLARE I_PRICE INTEGER ;
DECLARE I_NAME CHAR(24) ;
DECLARE I_DATA VARCHAR(50) ;
DECLARE OL_DIST_INFO CHAR(24) ;
DECLARE S_DATA VARCHAR(50) ;
DECLARE S_QUANTITY SMALLINT ;

SET ( I_PRICE , I_NAME , I_DATA )
= ( SELECT
I_PRICE
, I_NAME
, I_DATA
FROM ITEM
WHERE ITEM.I_ID = NEW_OL_LOCAL.I_ID
);
SET ( OL_DIST_INFO , S_DATA , S_QUANTITY )
= ( SELECT OL_DIST_INFO
, S_DATA
, S_QUANTITY
FROM NEW TABLE ( UPDATE STOCK
INCLUDE ( OL_DIST_INFO CHAR( 24 ) )

```

```

        SET S_QUANTITY = CASE WHEN S_QUANTITY -
NEW_OL_LOCAL.I_QTY >= 10
        THEN S_QUANTITY -
NEW_OL_LOCAL.I_QTY
        ELSE S_QUANTITY -
NEW_OL_LOCAL.I_QTY + 91
        END
        ,S_ORDER_CNT = S_ORDER_CNT +
SMALLINT( 1 )
        ,S_YTD = S_YTD +
NEW_OL_LOCAL.I_QTY
SMALLINT( 1 ) THEN S_DIST_01
        WHEN SMALLINT( 2 )
THEN S_DIST_02
        WHEN SMALLINT( 3 )
THEN S_DIST_03
        WHEN SMALLINT( 4 )
THEN S_DIST_04
        WHEN SMALLINT( 5 )
THEN S_DIST_05
        WHEN SMALLINT( 6 )
THEN S_DIST_06
        WHEN SMALLINT( 7 )
THEN S_DIST_07
        WHEN SMALLINT( 8 )
THEN S_DIST_08
        WHEN SMALLINT( 9 )
THEN S_DIST_09
        WHEN SMALLINT( 10 )
THEN S_DIST_10
        END
        WHERE S_I_ID = NEW_OL_LOCAL.I_ID
        AND S_W_ID = NEW_OL_LOCAL.W_ID
        ) AS U
    )
;
RETURN VALUES( VAR.I_PRICE
    ,VAR.I_NAME
    ,VAR.I_DATA
    ,VAR.OL_DIST_INFO
    ,VAR.S_DATA
    ,VAR.S_QUANTITY
    )
;
END
$

```

```

CREATE FUNCTION NEW_WH ( O_ID    INTEGER
    ,W_ID    INTEGER
    ,D_ID    SMALLINT
    ,C_ID    INTEGER
    ,O_ENTRY_D BIGINT
    ,O_OL_CNT SMALLINT
    ,O_ALL_LOCAL SMALLINT
    )
RETURNS TABLE ( W_TAX    INTEGER
    ,C_DISCOUNT INTEGER
    ,C_LAST    VARCHAR(16)
    ,C_CREDIT CHAR(2)
    )
SPECIFIC NEW_WH
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION
LANGUAGE SQL
VAR: BEGIN ATOMIC
    DECLARE C_DISCOUNT INTEGER ;
    DECLARE C_LAST    VARCHAR(16);

```

```

DECLARE C_CREDIT CHAR(2);
DECLARE W_TAX    INTEGER;
INSERT
    INTO NEW_ORDER ( NO_O_ID, NO_D_ID, NO_W_ID )
        VALUES ( O_ID
            ,D_ID
            ,W_ID
            )
;
INSERT
    INTO ORDERS ( O_C_ID, O_ENTRY_D, O_CARRIER_ID, O_OL_CNT,
O_ALL_LOCAL, O_ID, O_W_ID, O_D_ID )
        VALUES ( C_ID
            ,O_ENTRY_D
            ,0
            ,O_OL_CNT
            ,O_ALL_LOCAL
            ,O_ID
            ,W_ID
            ,D_ID
            )
;
SET ( C_DISCOUNT, C_LAST, C_CREDIT )
= ( SELECT C_DISCOUNT, C_LAST, C_CREDIT
    FROM CUSTOMER
    WHERE C_ID = NEW_WH.C_ID
    AND C_W_ID = W_ID
    AND C_D_ID = D_ID
    )
;
SET W_TAX
= ( SELECT W_TAX
    FROM WAREHOUSE
    WHERE W_ID = NEW_WH.W_ID
    )
;
RETURN VALUES ( W_TAX , C_DISCOUNT , C_LAST , C_CREDIT );
END
$

```

Src.Srv/cat-proc.ddl

```

CREATE PROCEDURE news
    (in new_in  varchar(270) FOR BIT DATA,
    out new_out varchar(662) FOR BIT DATA)
LANGUAGE C
PARAMETER STYLE GENERAL
EXTERNAL NAME 'c:\SQLLIB\function\vrpctpc!news'
not fenced;

```

```

CREATE PROCEDURE ords
    (in ord_in  varchar(42) FOR BIT DATA,
    out ord_out varchar(446) FOR BIT DATA)
LANGUAGE C
PARAMETER STYLE GENERAL
EXTERNAL NAME 'c:\SQLLIB\function\vrpctpc!ords'
not fenced;

```

```

CREATE PROCEDURE dels
    (in del_in  varchar(22) FOR BIT DATA,
    out del_out varchar(50) FOR BIT DATA)
LANGUAGE C
PARAMETER STYLE GENERAL
EXTERNAL NAME 'c:\SQLLIB\function\vrpctpc!dels'
not fenced;

```

Src.Srv/tpcc_all_sql.sqc

```
/*
*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****
*****/
/*
* tpcc_all_sql.sqc - Client/Server code for TPCC
*/
#include <stdlib.h>
#include <errno.h>
#include "db2tpcc.h"
#include "tpccapp.h"
#include "tpccdbg.h"
#include "sqlca.h"
#include "sql.h"
#include "lval.h"
// -----
// New Order SERVER
// -----
int static is_ORIGINAL( char *string, short length );
SQL_API_RC new_order_internal( char *pin, char *pout )
{
    struct out_neword_struct *neword;
    struct in_neword_struct *in_neword;
    struct sqlca sqlca ;
    int fbadItemDetected = 0 ;
    EXEC SQL BEGIN DECLARE SECTION;
        char c_last [ 16 ] ;
        char c_credit [ 2 ] ;
        sqlint32 c_discount ;
        sqlint32 dist_tax ;
        sqlint32 ware_tax ;
        sqlint32 w_id ;
        short d_id ;
        sqlint32 c_id ;
        sqlint32 next_o_id ;
        short s_quantity ;
        sqlint32 supply_w_id ;
        short inputItemCount ;
        char stockDistrictInformation [ 24 ] ;
        char item_name[ 24 ] ;
        sqlint64 o_entry_d ;

        short allLocal ;

        sqlint32 item_price ;

        struct i_data_type { short len ; char data[ 50 ] ; } i_data ;
        struct s_data_type { short len ; char data[ 50 ] ; } s_data ;
        sqlint32 id0, id1, id2, id3, id4, id5, id6, id7 ;
        sqlint32 id8, id9, id10, id11, id12, id13, id14 ;
        sqlint32 supply_w_id0, supply_w_id1, supply_w_id2, supply_w_id3 ;
        sqlint32 supply_w_id4, supply_w_id5, supply_w_id6, supply_w_id7 ;
        sqlint32 supply_w_id8, supply_w_id9, supply_w_id10, supply_w_id11 ;
        sqlint32 supply_w_id12, supply_w_id13, supply_w_id14 ;
        short ol_quantity0, ol_quantity1, ol_quantity2, ol_quantity3 ;
```

```
        short ol_quantity4, ol_quantity5, ol_quantity6, ol_quantity7 ;
        short ol_quantity8, ol_quantity9, ol_quantity10, ol_quantity11 ;
        short ol_quantity12, ol_quantity13, ol_quantity14 ;

EXEC SQL END DECLARE SECTION;
int storedProcRc ;
int inputItemArrayIndex ;
char stockDistrictInformationArray [15][25];
#define stockDistrictInformation stockDistrictInformationArray[
inputItemArrayIndex ]
// Redirected input fields
#define w_id in_neword->s_W_ID
#define d_id in_neword->s_D_ID
#define c_id in_neword->s_C_ID
#define o_entry_d in_neword->s_O_ENTRY_D_time
#define inputItemCount in_neword->s_O_OL_CNT
#define allLocal in_neword->s_all_local
// Redirected output fields
#define c_last neword->s_C_LAST
#define c_credit neword->s_C_CREDIT
#define c_discount neword->s_C_DISCOUNT
#define ware_tax neword->s_W_TAX
#define dist_tax neword->s_D_TAX
#define s_quantity neword->item[ inputItemArrayIndex ].s_S_QUANTITY
// This output field becomes an input field to order_line

#define next_o_id neword->s_O_ID

#define item_name neword->item[ inputItemArrayIndex ].s_I_NAME
// item_price holds the integer version of this value. If the return structure was
// an integer this would not be necessary.
sqlint32 i_priceArray[ 15 ] ;
#define item_price i_priceArray[ inputItemArrayIndex ]
// Handle the generic/brand distinction
struct i_data_type i_dataArray[ 15 ] ;
struct s_data_type s_dataArray[ 15 ] ;
#define i_data i_dataArray[ inputItemArrayIndex ]
#define s_data s_dataArray[ inputItemArrayIndex ]

// Redirect hostvars to input structure
#define id0 in_neword->in_item[0].s_OL_I_ID
#define id1 in_neword->in_item[1].s_OL_I_ID
#define id2 in_neword->in_item[2].s_OL_I_ID
#define id3 in_neword->in_item[3].s_OL_I_ID
#define id4 in_neword->in_item[4].s_OL_I_ID
#define id5 in_neword->in_item[5].s_OL_I_ID
#define id6 in_neword->in_item[6].s_OL_I_ID
#define id7 in_neword->in_item[7].s_OL_I_ID
#define id8 in_neword->in_item[8].s_OL_I_ID
#define id9 in_neword->in_item[9].s_OL_I_ID
#define id10 in_neword->in_item[10].s_OL_I_ID
#define id11 in_neword->in_item[11].s_OL_I_ID
#define id12 in_neword->in_item[12].s_OL_I_ID
#define id13 in_neword->in_item[13].s_OL_I_ID
#define id14 in_neword->in_item[14].s_OL_I_ID
#define ol_quantity0 in_neword->in_item[ 0 ].s_OL_QUANTITY
#define ol_quantity1 in_neword->in_item[ 1 ].s_OL_QUANTITY
#define ol_quantity2 in_neword->in_item[ 2 ].s_OL_QUANTITY
#define ol_quantity3 in_neword->in_item[ 3 ].s_OL_QUANTITY
#define ol_quantity4 in_neword->in_item[ 4 ].s_OL_QUANTITY
#define ol_quantity5 in_neword->in_item[ 5 ].s_OL_QUANTITY
#define ol_quantity6 in_neword->in_item[ 6 ].s_OL_QUANTITY
#define ol_quantity7 in_neword->in_item[ 7 ].s_OL_QUANTITY
#define ol_quantity8 in_neword->in_item[ 8 ].s_OL_QUANTITY
#define ol_quantity9 in_neword->in_item[ 9 ].s_OL_QUANTITY
#define ol_quantity10 in_neword->in_item[ 10 ].s_OL_QUANTITY
#define ol_quantity11 in_neword->in_item[ 11 ].s_OL_QUANTITY
```

```

#define ol_quantity12 in_newword->in_item[ 12 ].s_OL_QUANTITY
#define ol_quantity13 in_newword->in_item[ 13 ].s_OL_QUANTITY
#define ol_quantity14 in_newword->in_item[ 14 ].s_OL_QUANTITY
#define supply_w_id0 in_newword->in_item[ 0 ].s_OL_SUPPLY_W_ID
#define supply_w_id1 in_newword->in_item[ 1 ].s_OL_SUPPLY_W_ID
#define supply_w_id2 in_newword->in_item[ 2 ].s_OL_SUPPLY_W_ID
#define supply_w_id3 in_newword->in_item[ 3 ].s_OL_SUPPLY_W_ID
#define supply_w_id4 in_newword->in_item[ 4 ].s_OL_SUPPLY_W_ID
#define supply_w_id5 in_newword->in_item[ 5 ].s_OL_SUPPLY_W_ID
#define supply_w_id6 in_newword->in_item[ 6 ].s_OL_SUPPLY_W_ID
#define supply_w_id7 in_newword->in_item[ 7 ].s_OL_SUPPLY_W_ID
#define supply_w_id8 in_newword->in_item[ 8 ].s_OL_SUPPLY_W_ID
#define supply_w_id9 in_newword->in_item[ 9 ].s_OL_SUPPLY_W_ID
#define supply_w_id10 in_newword->in_item[ 10 ].s_OL_SUPPLY_W_ID
#define supply_w_id11 in_newword->in_item[ 11 ].s_OL_SUPPLY_W_ID
#define supply_w_id12 in_newword->in_item[ 12 ].s_OL_SUPPLY_W_ID
#define supply_w_id13 in_newword->in_item[ 13 ].s_OL_SUPPLY_W_ID
#define supply_w_id14 in_newword->in_item[ 14 ].s_OL_SUPPLY_W_ID
EXEC SQL DECLARE ISOL_Remote_1 CURSOR FOR
  WITH DATA AS ( SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , 0 AS OL_DELIVERY_D
    , I_QTY
    , (I_PRICE * I_QTY) AS TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
    FROM ( SELECT :next_o_id as O_ID
      , :w_id AS W_ID
      , :d_id as D_ID
      , OL_NUMBER
      , I_ID
      , I_SUPPLY_W_ID
      , I_QTY
      FROM Table( VALUES
        ( SMALLINT(1) , :id0 , :ol_quantity0
, :supply_w_id0 )
        ) AS X (OL_NUMBER , I_ID , I_QTY
, I_SUPPLY_W_ID )
        ) AS ITEMLIST
    , TABLE(NEW_OL_ALL( I_ID
      , I_QTY
      , W_ID
      , I_SUPPLY_W_ID
      , O_ID
      , D_ID
      )
    ) AS NEW_OL_ALL
    WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
  )
  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
  FROM NEW TABLE ( INSERT INTO ORDER_LINE
    ( OL_O_ID
    , OL_D_ID
    , OL_W_ID
    , OL_NUMBER
    , OL_I_ID
    , OL_SUPPLY_W_ID
    , OL_DELIVERY_D
    , OL_QUANTITY
    , OL_AMOUNT
    , OL_DIST_INFO
    )

```

```

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
  FROM DATA
  ) AS INS
;
EXEC SQL DECLARE ISOL_Remote_2 CURSOR FOR
  WITH DATA AS ( SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , 0 AS OL_DELIVERY_D
    , I_QTY
    , (I_PRICE * I_QTY) AS TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
    FROM ( SELECT :next_o_id as O_ID
      , :w_id AS W_ID
      , :d_id as D_ID
      , OL_NUMBER
      , I_ID
      , I_SUPPLY_W_ID
      , I_QTY
      FROM Table( VALUES
        ( SMALLINT(1) , :id0 , :ol_quantity0
, :supply_w_id0 )
        , ( SMALLINT(2) , :id1 , :ol_quantity1
, :supply_w_id1 )
        ) AS X (OL_NUMBER , I_ID , I_QTY
, I_SUPPLY_W_ID )
        ) AS ITEMLIST
    , TABLE(NEW_OL_ALL( I_ID
      , I_QTY
      , W_ID
      , I_SUPPLY_W_ID
      , O_ID
      , D_ID
      )
    ) AS NEW_OL_ALL
    WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
  )
  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
  FROM NEW TABLE ( INSERT INTO ORDER_LINE
    ( OL_O_ID
    , OL_D_ID
    , OL_W_ID
    , OL_NUMBER
    , OL_I_ID
    , OL_SUPPLY_W_ID

```

```

,OL_DELIVERY_D
,OL_QUANTITY
,OL_AMOUNT
,OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
,I_NAME CHAR(24)
,I_DATA VARCHAR(50)
,S_DATA VARCHAR(50)
,S_QUANTITY SMALLINT )
SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,OL_DELIVERY_D
,I_QTY
,TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_3 CURSOR FOR
WITH DATA AS ( SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,0 AS OL_DELIVERY_D
,I_QTY
,(I_PRICE * I_QTY) AS TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
,:w_id as W_ID
,:d_id as D_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,I_QTY
FROM Table( VALUES
( SMALLINT(1) ,:id0 ,:ol_quantity0
,:supply_w_id0 )
,( SMALLINT(2) ,:id1 ,:ol_quantity1
,:supply_w_id1 )
,( SMALLINT(3) ,:id2 ,:ol_quantity2
,:supply_w_id2 )
) AS X (OL_NUMBER , I_ID , I_QTY
,I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE(NEW_OL_ALL( I_ID
,I_QTY
,W_ID
,I_SUPPLY_W_ID
,O_ID
,D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE

```

```

( OL_O_ID
,OL_D_ID
,OL_W_ID
,OL_NUMBER
,OL_I_ID
,OL_SUPPLY_W_ID
,OL_DELIVERY_D
,OL_QUANTITY
,OL_AMOUNT
,OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
,I_NAME CHAR(24)
,I_DATA VARCHAR(50)
,S_DATA VARCHAR(50)
,S_QUANTITY SMALLINT )
SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,OL_DELIVERY_D
,I_QTY
,TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_4 CURSOR FOR
WITH DATA AS ( SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,0 AS OL_DELIVERY_D
,I_QTY
,(I_PRICE * I_QTY) AS TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
,:w_id as W_ID
,:d_id as D_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,I_QTY
FROM Table( VALUES
( SMALLINT(1) ,:id0 ,:ol_quantity0
,:supply_w_id0 )
,( SMALLINT(2) ,:id1 ,:ol_quantity1
,:supply_w_id1 )
,( SMALLINT(3) ,:id2 ,:ol_quantity2
,:supply_w_id2 )
,( SMALLINT(4) ,:id3 ,:ol_quantity3
,:supply_w_id3 )
) AS X (OL_NUMBER , I_ID , I_QTY
,I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE(NEW_OL_ALL( I_ID
,I_QTY
,W_ID
,I_SUPPLY_W_ID
,O_ID
,D_ID

```

```

        )
        ) AS NEW_OL_ALL
        WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
    )
    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
    S_QUANTITY
    FROM NEW TABLE ( INSERT INTO ORDER_LINE
    ( OL_O_ID
    , OL_D_ID
    , OL_W_ID
    , OL_NUMBER
    , OL_I_ID
    , OL_SUPPLY_W_ID
    , OL_DELIVERY_D
    , OL_QUANTITY
    , OL_AMOUNT
    , OL_DIST_INFO
    )
    INCLUDE ( I_PRICE INTEGER
    , I_NAME CHAR(24)
    , I_DATA VARCHAR(50)
    , S_DATA VARCHAR(50)
    , S_QUANTITY SMALLINT )
    SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , OL_DELIVERY_D
    , I_QTY
    , TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE, I_NAME, I_DATA, S_DATA,
    S_QUANTITY
    FROM DATA
    ) AS INS
;
EXEC SQL DECLARE ISOL_Remote_5 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT(1) , :id0 , :ol_quantity0
, :supply_w_id0 )
, ( SMALLINT(2) , :id1 , :ol_quantity1
, :supply_w_id1 )
, ( SMALLINT(3) , :id2 , :ol_quantity2
, :supply_w_id2 )
, ( SMALLINT(4) , :id3 , :ol_quantity3
, :supply_w_id3 )

```

```

, ( SMALLINT(5) , :id4 , :ol_quantity4
, :supply_w_id4 )
) AS X ( OL_NUMBER , I_ID , I_QTY
, I_SUPPLY_W_ID )
) AS ITEM_LIST
, TABLE(NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_6 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID

```



```

        ,I_QTY
        FROM Table( VALUES
        ( SMALLINT( 1 ) ,:id0 ,:ol_quantity0
, :supply_w_id0 )
        , ( SMALLINT( 2 ) ,:id1 ,:ol_quantity1
, :supply_w_id1 )
        , ( SMALLINT( 3 ) ,:id2 ,:ol_quantity2
, :supply_w_id2 )
        , ( SMALLINT( 4 ) ,:id3 ,:ol_quantity3
, :supply_w_id3 )
        , ( SMALLINT( 5 ) ,:id4 ,:ol_quantity4
, :supply_w_id4 )
        , ( SMALLINT( 6 ) ,:id5 ,:ol_quantity5
, :supply_w_id5 )
        ) AS X (OL_NUMBER , I_ID , I_QTY
, I_SUPPLY_W_ID )
        ) AS ITEMLIST
        , TABLE(NEW_OL_ALL( I_ID
        , I_QTY
        , W_ID
        , I_SUPPLY_W_ID
        , O_ID
        , D_ID
        )
        ) AS NEW_OL_ALL
        WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
    )
    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
    S_QUANTITY
    FROM NEW TABLE ( INSERT INTO ORDER_LINE
    ( OL_O_ID
    , OL_D_ID
    , OL_W_ID
    , OL_NUMBER
    , OL_I_ID
    , OL_SUPPLY_W_ID
    , OL_DELIVERY_D
    , OL_QUANTITY
    , OL_AMOUNT
    , OL_DIST_INFO
    )
    INCLUDE( I_PRICE INTEGER
    , I_NAME CHAR(24)
    , I_DATA VARCHAR(50)
    , S_DATA VARCHAR(50)
    , S_QUANTITY SMALLINT )
    SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , OL_DELIVERY_D
    , I_QTY
    , TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE, I_NAME, I_DATA, S_DATA,
    S_QUANTITY
    FROM DATA
    ) AS INS
;
EXEC SQL DECLARE ISOL_Remote_7 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID

```

```

, I_SUPPLY_W_ID
, O AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) ,:id0 ,:ol_quantity0
, :supply_w_id0 )
, ( SMALLINT( 2 ) ,:id1 ,:ol_quantity1
, :supply_w_id1 )
, ( SMALLINT( 3 ) ,:id2 ,:ol_quantity2
, :supply_w_id2 )
, ( SMALLINT( 4 ) ,:id3 ,:ol_quantity3
, :supply_w_id3 )
, ( SMALLINT( 5 ) ,:id4 ,:ol_quantity4
, :supply_w_id4 )
, ( SMALLINT( 6 ) ,:id5 ,:ol_quantity5
, :supply_w_id5 )
, ( SMALLINT( 7 ) ,:id6 ,:ol_quantity6
, :supply_w_id6 )
) AS X (OL_NUMBER , I_ID , I_QTY
, I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE(NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_DIST_INFO
)
INCLUDE( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D

```

```

        ,I_QTY
        ,TOTAL_PRICE
        ,OL_DIST_INFO
        ,I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
    FROM DATA

    ) AS INS
;
EXEC SQL DECLARE ISOL_Remote_8 CURSOR FOR
    WITH DATA AS ( SELECT O_ID
        ,D_ID
        ,W_ID
        ,OL_NUMBER
        ,I_ID
        ,I_SUPPLY_W_ID
        ,O AS OL_DELIVERY_D
        ,I_QTY
        ,(I_PRICE * I_QTY) AS TOTAL_PRICE
        ,OL_DIST_INFO
        ,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
        FROM ( SELECT :next_o_id as O_ID
            ,:w_id AS W_ID
            ,:d_id as D_ID
            ,OL_NUMBER
            ,I_ID
            ,I_SUPPLY_W_ID
            ,I_QTY
        FROM Table( VALUES
            ( SMALLINT( 1) ,:id0 ,:ol_quantity0
, :supply_w_id0 )
            , ( SMALLINT( 2) ,:id1 ,:ol_quantity1
, :supply_w_id1 )
            , ( SMALLINT( 3) ,:id2 ,:ol_quantity2
, :supply_w_id2 )
            , ( SMALLINT( 4) ,:id3 ,:ol_quantity3
, :supply_w_id3 )
            , ( SMALLINT( 5) ,:id4 ,:ol_quantity4
, :supply_w_id4 )
            , ( SMALLINT( 6) ,:id5 ,:ol_quantity5
, :supply_w_id5 )
            , ( SMALLINT( 7) ,:id6 ,:ol_quantity6
, :supply_w_id6 )
            , ( SMALLINT( 8) ,:id7 ,:ol_quantity7
, :supply_w_id7 )
        ) AS X (OL_NUMBER, I_ID, I_QTY
        ) AS ITEMLIST
        ,TABLE(NEW_OL_ALL( I_ID
            ,I_QTY
            ,W_ID
            ,I_SUPPLY_W_ID
            ,O_ID
            ,D_ID
        )
        ) AS NEW_OL_ALL
        WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
    )
    SELECT I_PRICE, I_NAME, I_DATA, OL_DIST_INFO, S_DATA,
S_QUANTITY
    FROM NEW TABLE ( INSERT INTO ORDER_LINE
        ( OL_O_ID
        ,OL_D_ID
        ,OL_W_ID
        ,OL_NUMBER
        ,OL_I_ID

```

```

        ,OL_SUPPLY_W_ID
        ,OL_DELIVERY_D
        ,OL_QUANTITY
        ,OL_AMOUNT
        ,OL_DIST_INFO
    )
    INCLUDE ( I_PRICE INTEGER
        ,I_NAME CHAR(24)
        ,I_DATA VARCHAR(50)
        ,S_DATA VARCHAR(50)
        ,S_QUANTITY SMALLINT )
    SELECT O_ID
        ,D_ID
        ,W_ID
        ,OL_NUMBER
        ,I_ID
        ,I_SUPPLY_W_ID
        ,OL_DELIVERY_D
        ,I_QTY
        ,TOTAL_PRICE
        ,OL_DIST_INFO
        ,I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
    FROM DATA
    ) AS INS
;
EXEC SQL DECLARE ISOL_Remote_9 CURSOR FOR
    WITH DATA AS ( SELECT O_ID
        ,D_ID
        ,W_ID
        ,OL_NUMBER
        ,I_ID
        ,I_SUPPLY_W_ID
        ,O AS OL_DELIVERY_D
        ,I_QTY
        ,(I_PRICE * I_QTY) AS TOTAL_PRICE
        ,OL_DIST_INFO
        ,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
        FROM ( SELECT :next_o_id as O_ID
            ,:w_id AS W_ID
            ,:d_id as D_ID
            ,OL_NUMBER
            ,I_ID
            ,I_SUPPLY_W_ID
            ,I_QTY
        FROM Table( VALUES
            ( SMALLINT( 1) ,:id0 ,:ol_quantity0
, :supply_w_id0 )
            , ( SMALLINT( 2) ,:id1 ,:ol_quantity1
, :supply_w_id1 )
            , ( SMALLINT( 3) ,:id2 ,:ol_quantity2
, :supply_w_id2 )
            , ( SMALLINT( 4) ,:id3 ,:ol_quantity3
, :supply_w_id3 )
            , ( SMALLINT( 5) ,:id4 ,:ol_quantity4
, :supply_w_id4 )
            , ( SMALLINT( 6) ,:id5 ,:ol_quantity5
, :supply_w_id5 )
            , ( SMALLINT( 7) ,:id6 ,:ol_quantity6
, :supply_w_id6 )
            , ( SMALLINT( 8) ,:id7 ,:ol_quantity7
, :supply_w_id7 )
            , ( SMALLINT( 9) ,:id8 ,:ol_quantity8
, :supply_w_id8 )
        ) AS X (OL_NUMBER, I_ID, I_QTY
        ,I_SUPPLY_W_ID )
        ) AS ITEMLIST
        ,TABLE(NEW_OL_ALL( I_ID

```

```

        ,I_QTY
        ,W_ID
        ,I_SUPPLY_W_ID
        ,O_ID
        ,D_ID
    )
    ) AS NEW_OL_ALL
    WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_10 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT(1) , :id0 , :ol_quantity0
, :supply_w_id0 )
, ( SMALLINT(2) , :id1 , :ol_quantity1

```

```

, ( SMALLINT(3) , :id2 , :ol_quantity2
, :supply_w_id2 )
, ( SMALLINT(4) , :id3 , :ol_quantity3
, :supply_w_id3 )
, ( SMALLINT(5) , :id4 , :ol_quantity4
, :supply_w_id4 )
, ( SMALLINT(6) , :id5 , :ol_quantity5
, :supply_w_id5 )
, ( SMALLINT(7) , :id6 , :ol_quantity6
, :supply_w_id6 )
, ( SMALLINT(8) , :id7 , :ol_quantity7
, :supply_w_id7 )
, ( SMALLINT(9) , :id8 , :ol_quantity8
, :supply_w_id8 )
, ( SMALLINT(10) , :id9 ,
:ol_quantity9 , :supply_w_id9 )
) AS X (OL_NUMBER , I_ID , I_QTY
, I_SUPPLY_W_ID )
) AS ITEM LIST
, TABLE(NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_11 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID

```

```

, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
      , :w_id AS W_ID
      , :d_id as D_ID
      , OL_NUMBER
      , I_ID
      , I_SUPPLY_W_ID
      , I_QTY
FROM Table( VALUES
      ( SMALLINT( 1 ) , :id0 , :ol_quantity0
      , :supply_w_id0 )
      , ( SMALLINT( 2 ) , :id1 , :ol_quantity1
      , :supply_w_id1 )
      , ( SMALLINT( 3 ) , :id2 , :ol_quantity2
      , :supply_w_id2 )
      , ( SMALLINT( 4 ) , :id3 , :ol_quantity3
      , :supply_w_id3 )
      , ( SMALLINT( 5 ) , :id4 , :ol_quantity4
      , :supply_w_id4 )
      , ( SMALLINT( 6 ) , :id5 , :ol_quantity5
      , :supply_w_id5 )
      , ( SMALLINT( 7 ) , :id6 , :ol_quantity6
      , :supply_w_id6 )
      , ( SMALLINT( 8 ) , :id7 , :ol_quantity7
      , :supply_w_id7 )
      , ( SMALLINT( 9 ) , :id8 , :ol_quantity8
      , :supply_w_id8 )
      , ( SMALLINT( 10 ) , :id9 ,
      :ol_quantity9 , :supply_w_id9 )
      , ( SMALLINT( 11 ) , :id10 ,
      :ol_quantity10 , :supply_w_id10 )
) AS X (OL_NUMBER , I_ID , I_QTY
, I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE(NEW_OL_ALL( I_ID
      , I_QTY
      , W_ID
      , I_SUPPLY_W_ID
      , O_ID
      , D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE( I_PRICE INTEGER
, I_NAME CHAR(24)

```

```

, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT)
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_12 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
      , :w_id AS W_ID
      , :d_id as D_ID
      , OL_NUMBER
      , I_ID
      , I_SUPPLY_W_ID
      , I_QTY
FROM Table( VALUES
      ( SMALLINT( 1 ) , :id0 , :ol_quantity0
      , :supply_w_id0 )
      , ( SMALLINT( 2 ) , :id1 , :ol_quantity1
      , :supply_w_id1 )
      , ( SMALLINT( 3 ) , :id2 , :ol_quantity2
      , :supply_w_id2 )
      , ( SMALLINT( 4 ) , :id3 , :ol_quantity3
      , :supply_w_id3 )
      , ( SMALLINT( 5 ) , :id4 , :ol_quantity4
      , :supply_w_id4 )
      , ( SMALLINT( 6 ) , :id5 , :ol_quantity5
      , :supply_w_id5 )
      , ( SMALLINT( 7 ) , :id6 , :ol_quantity6
      , :supply_w_id6 )
      , ( SMALLINT( 8 ) , :id7 , :ol_quantity7
      , :supply_w_id7 )
      , ( SMALLINT( 9 ) , :id8 , :ol_quantity8
      , :supply_w_id8 )
      , ( SMALLINT( 10 ) , :id9 ,
      :ol_quantity9 , :supply_w_id9 )
      , ( SMALLINT( 11 ) , :id10 ,
      :ol_quantity10 , :supply_w_id10 )
      , ( SMALLINT( 12 ) , :id11 ,
      :ol_quantity11 , :supply_w_id11 )
) AS X (OL_NUMBER , I_ID , I_QTY
, I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE(NEW_OL_ALL( I_ID
      , I_QTY
      , W_ID

```

```

        ,I_SUPPLY_W_ID
        ,O_ID
        ,D_ID
    )
    ) AS NEW_OL_ALL
    WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
    ( OL_O_ID
    , OL_D_ID
    , OL_W_ID
    , OL_NUMBER
    , OL_I_ID
    , OL_SUPPLY_W_ID
    , OL_DELIVERY_D
    , OL_QUANTITY
    , OL_AMOUNT
    , OL_DIST_INFO
    )
    INCLUDE ( I_PRICE INTEGER
    , I_NAME CHAR(24)
    , I_DATA VARCHAR(50)
    , S_DATA VARCHAR(50)
    , S_QUANTITY SMALLINT )
    SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , OL_DELIVERY_D
    , I_QTY
    , TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
    FROM DATA
    ) AS INS
;
EXEC SQL DECLARE ISOL_Remote_13 CURSOR FOR
WITH DATA AS ( SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , 0 AS OL_DELIVERY_D
    , I_QTY
    , (I_PRICE * I_QTY) AS TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
    FROM ( SELECT :next_o_id as O_ID
    , :w_id AS W_ID
    , :d_id as D_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , I_QTY
    FROM Table( VALUES
        ( SMALLINT( 1 ) , :id0 , :ol_quantity0
, :supply_w_id0 )
        , ( SMALLINT( 2 ) , :id1 , :ol_quantity1
, :supply_w_id1 )
        , ( SMALLINT( 3 ) , :id2 , :ol_quantity2
, :supply_w_id2 )

```

```

        , ( SMALLINT( 4 ) , :id3 , :ol_quantity3
, :supply_w_id3 )
        , ( SMALLINT( 5 ) , :id4 , :ol_quantity4
, :supply_w_id4 )
        , ( SMALLINT( 6 ) , :id5 , :ol_quantity5
, :supply_w_id5 )
        , ( SMALLINT( 7 ) , :id6 , :ol_quantity6
, :supply_w_id6 )
        , ( SMALLINT( 8 ) , :id7 , :ol_quantity7
, :supply_w_id7 )
        , ( SMALLINT( 9 ) , :id8 , :ol_quantity8
, :supply_w_id8 )
        , ( SMALLINT( 10 ) , :id9 ,
:ol_quantity9 , :supply_w_id9 )
        , ( SMALLINT( 11 ) , :id10 ,
:ol_quantity10 , :supply_w_id10 )
        , ( SMALLINT( 12 ) , :id11 ,
:ol_quantity11 , :supply_w_id11 )
        , ( SMALLINT( 13 ) , :id12 ,
:ol_quantity12 , :supply_w_id12 )
    ) AS X ( OL_NUMBER , I_ID , I_QTY
, I_SUPPLY_W_ID )
    ) AS ITEM_LIST
    , TABLE( NEW_OL_ALL ( I_ID
    , I_QTY
    , W_ID
    , I_SUPPLY_W_ID
    , O_ID
    , D_ID
    )
    ) AS NEW_OL_ALL
    WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
    ( OL_O_ID
    , OL_D_ID
    , OL_W_ID
    , OL_NUMBER
    , OL_I_ID
    , OL_SUPPLY_W_ID
    , OL_DELIVERY_D
    , OL_QUANTITY
    , OL_AMOUNT
    , OL_DIST_INFO
    )
    INCLUDE ( I_PRICE INTEGER
    , I_NAME CHAR(24)
    , I_DATA VARCHAR(50)
    , S_DATA VARCHAR(50)
    , S_QUANTITY SMALLINT )
    SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , OL_DELIVERY_D
    , I_QTY
    , TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
    FROM DATA
    ) AS INS
;

```

```

EXEC SQL DECLARE ISOL_Remote_14 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0
, :supply_w_id0 )
, ( SMALLINT( 2 ) , :id1 , :ol_quantity1
, :supply_w_id1 )
, ( SMALLINT( 3 ) , :id2 , :ol_quantity2
, :supply_w_id2 )
, ( SMALLINT( 4 ) , :id3 , :ol_quantity3
, :supply_w_id3 )
, ( SMALLINT( 5 ) , :id4 , :ol_quantity4
, :supply_w_id4 )
, ( SMALLINT( 6 ) , :id5 , :ol_quantity5
, :supply_w_id5 )
, ( SMALLINT( 7 ) , :id6 , :ol_quantity6
, :supply_w_id6 )
, ( SMALLINT( 8 ) , :id7 , :ol_quantity7
, :supply_w_id7 )
, ( SMALLINT( 9 ) , :id8 , :ol_quantity8
, :supply_w_id8 )
, ( SMALLINT( 10 ) , :id9 ,
:ol_quantity9 , :supply_w_id9 )
, ( SMALLINT( 11 ) , :id10 ,
:ol_quantity10 , :supply_w_id10 )
, ( SMALLINT( 12 ) , :id11 ,
:ol_quantity11 , :supply_w_id11 )
, ( SMALLINT( 13 ) , :id12 ,
:ol_quantity12 , :supply_w_id12 )
, ( SMALLINT( 14 ) , :id13 ,
:ol_quantity13 , :supply_w_id13 )
) AS X ( OL_NUMBER , I_ID , I_QTY
, I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE(NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID

```

```

, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_15 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0
, :supply_w_id0 )
, ( SMALLINT( 2 ) , :id1 , :ol_quantity1
, :supply_w_id1 )
, ( SMALLINT( 3 ) , :id2 , :ol_quantity2
, :supply_w_id2 )
, ( SMALLINT( 4 ) , :id3 , :ol_quantity3
, :supply_w_id3 )
, ( SMALLINT( 5 ) , :id4 , :ol_quantity4
, :supply_w_id4 )
, ( SMALLINT( 6 ) , :id5 , :ol_quantity5
, :supply_w_id5 )
, ( SMALLINT( 7 ) , :id6 , :ol_quantity6
, :supply_w_id6 )
, ( SMALLINT( 8 ) , :id7 , :ol_quantity7
, :supply_w_id7 )
, ( SMALLINT( 9 ) , :id8 , :ol_quantity8
, :supply_w_id8 )
, ( SMALLINT( 10 ) , :id9 ,
:ol_quantity9 , :supply_w_id9 )

```

```

        , ( SMALLINT( 11 ) , :id10 ,
:ol_quantity10 , :supply_w_id10 )
        , ( SMALLINT( 12 ) , :id11 ,
:ol_quantity11 , :supply_w_id11 )
        , ( SMALLINT( 13 ) , :id12 ,
:ol_quantity12 , :supply_w_id12 )
        , ( SMALLINT( 14 ) , :id13 ,
:ol_quantity13 , :supply_w_id13 )
        , ( SMALLINT( 15 ) , :id14 ,
:ol_quantity14 , :supply_w_id14 )
        ) AS X ( OL_NUMBER , I_ID , I_QTY
, I_SUPPLY_W_ID )
        ) AS ITEMLIST
        , TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
        ) AS NEW_OL_ALL
        WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
    )
    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
    FROM NEW TABLE ( INSERT INTO ORDER_LINE
    ( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
    INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
    SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
    FROM DATA
    ) AS INS
;
EXEC SQL DECLARE ISOL_Local_1 CURSOR FOR
    WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, O AS OL_DELIVERY_D
, I_QTY

```

```

        , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
    FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
    FROM Table( VALUES
        ( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
        ) AS X ( OL_NUMBER , I_ID , I_QTY
        ) AS ITEMLIST
        , TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
        ) AS NEW_OL_LOCAL
        WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
    )
    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
    FROM NEW TABLE ( INSERT INTO ORDER_LINE
    ( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
    INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
    SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
    FROM DATA
    ) AS INS
;
EXEC SQL DECLARE ISOL_Local_2 CURSOR FOR
    WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID

```

```

, O AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT(1) , :id0 , :ol_quantity0 )
, ( SMALLINT(2) , :id1 , :ol_quantity1 )
) AS X (OL_NUMBER, I_ID, I_QTY
)
) AS ITEMLIST
, TABLE(NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
)
SELECT I_PRICE, I_NAME, I_DATA, OL_DIST_INFO, S_DATA,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_3 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID

```

```

, W_ID AS I_SUPPLY_W_ID
, O AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT(1) , :id0 , :ol_quantity0 )
, ( SMALLINT(2) , :id1 , :ol_quantity1 )
, ( SMALLINT(3) , :id2 , :ol_quantity2 )
) AS X (OL_NUMBER, I_ID, I_QTY
)
) AS ITEMLIST
, TABLE(NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
)
SELECT I_PRICE, I_NAME, I_DATA, OL_DIST_INFO, S_DATA,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_4 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID

```



```

,W_ID
,OL_NUMBER
,I_ID
,W_ID AS I_SUPPLY_W_ID
,0 AS OL_DELIVERY_D
,I_QTY
,(I_PRICE * I_QTY) AS TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE,I_NAME, I_DATA, S_DATA, S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
      ,:w_id AS W_ID
      ,:d_id as D_ID
      ,OL_NUMBER
      ,I_ID
      ,I_QTY
FROM Table( VALUES
      ( SMALLINT(1) ,:id0 ,:ol_quantity0 )
      ,( SMALLINT(2) ,:id1 ,:ol_quantity1 )
      ,( SMALLINT(3) ,:id2 ,:ol_quantity2 )
      ,( SMALLINT(4) ,:id3 ,:ol_quantity3 )
      ) AS X (OL_NUMBER , I_ID , I_QTY
)
) AS ITEMLIST
, TABLE(NEW_OL_LOCAL( I_ID
      ,I_QTY
      ,W_ID
      ,O_ID
      ,D_ID
      )
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
,OL_D_ID
,OL_W_ID
,OL_NUMBER
,OL_I_ID
,OL_SUPPLY_W_ID
,OL_DELIVERY_D
,OL_QUANTITY
,OL_AMOUNT
,OL_DIST_INFO
)
INCLUDE( I_PRICE INTEGER
,I_NAME CHAR(24)
,I_DATA VARCHAR(50)
,S_DATA VARCHAR(50)
,S_QUANTITY SMALLINT )
SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,OL_DELIVERY_D
,I_QTY
,TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE,I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;

```

```

EXEC SQL DECLARE ISOL_Local_5 CURSOR FOR

WITH DATA AS ( SELECT O_ID
      ,D_ID
      ,W_ID
      ,OL_NUMBER
      ,I_ID
      ,W_ID AS I_SUPPLY_W_ID
      ,0 AS OL_DELIVERY_D
      ,I_QTY
      ,(I_PRICE * I_QTY) AS TOTAL_PRICE
      ,OL_DIST_INFO
      ,I_PRICE,I_NAME, I_DATA, S_DATA, S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
      ,:w_id AS W_ID
      ,:d_id as D_ID
      ,OL_NUMBER
      ,I_ID
      ,I_QTY
FROM Table( VALUES
      ( SMALLINT(1) ,:id0 ,:ol_quantity0 )
      ,( SMALLINT(2) ,:id1 ,:ol_quantity1 )
      ,( SMALLINT(3) ,:id2 ,:ol_quantity2 )
      ,( SMALLINT(4) ,:id3 ,:ol_quantity3 )
      ,( SMALLINT(5) ,:id4 ,:ol_quantity4 )
      ) AS X (OL_NUMBER , I_ID , I_QTY
)
) AS ITEMLIST
, TABLE(NEW_OL_LOCAL( I_ID
      ,I_QTY
      ,W_ID
      ,O_ID
      ,D_ID
      )
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
,OL_D_ID
,OL_W_ID
,OL_NUMBER
,OL_I_ID
,OL_SUPPLY_W_ID
,OL_DELIVERY_D
,OL_QUANTITY
,OL_AMOUNT
,OL_DIST_INFO
)
INCLUDE( I_PRICE INTEGER
,I_NAME CHAR(24)
,I_DATA VARCHAR(50)
,S_DATA VARCHAR(50)
,S_QUANTITY SMALLINT )
SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,OL_DELIVERY_D
,I_QTY
,TOTAL_PRICE
) AS INS
;

```

```

        ,OL_DIST_INFO
        ,I_PRICE,I_NAME,I_DATA,S_DATA,
S_QUANTITY
        FROM DATA
        ) AS INS
;
EXEC SQL DECLARE ISOL_Local_6 CURSOR FOR

WITH DATA AS ( SELECT O_ID
        ,D_ID
        ,W_ID
        ,OL_NUMBER
        ,I_ID
        ,W_ID AS I_SUPPLY_W_ID
        ,0 AS OL_DELIVERY_D
        ,I_QTY
        ,(I_PRICE * I_QTY) AS TOTAL_PRICE
        ,OL_DIST_INFO
        ,I_PRICE,I_NAME,I_DATA,S_DATA,S_QUANTITY

        FROM ( SELECT :next_o_id as O_ID
                ,:w_id AS W_ID
                ,:d_id as D_ID
                ,OL_NUMBER
                ,I_ID
                ,I_QTY
                FROM Table( VALUES
                    ( SMALLINT(1) ,:id0 ,:ol_quantity0 )
                    ,( SMALLINT(2) ,:id1 ,:ol_quantity1 )
                    ,( SMALLINT(3) ,:id2 ,:ol_quantity2 )
                    ,( SMALLINT(4) ,:id3 ,:ol_quantity3 )
                    ,( SMALLINT(5) ,:id4 ,:ol_quantity4 )
                    ,( SMALLINT(6) ,:id5 ,:ol_quantity5 )
                ) AS X (OL_NUMBER , I_ID , I_QTY
        )

        ) AS ITEMLIST
        ,TABLE(NEW_OL_LOCAL( I_ID
                ,I_QTY
                ,W_ID
                ,O_ID
                ,D_ID
                )
        ) AS NEW_OL_LOCAL
        WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
    )

    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
    FROM NEW TABLE ( INSERT INTO ORDER_LINE
        ( OL_O_ID
        ,OL_D_ID
        ,OL_W_ID
        ,OL_NUMBER
        ,OL_I_ID
        ,OL_SUPPLY_W_ID
        ,OL_DELIVERY_D
        ,OL_QUANTITY
        ,OL_AMOUNT
        ,OL_DIST_INFO
        )
    INCLUDE ( I_PRICE INTEGER
        ,I_NAME CHAR(24)
        ,I_DATA VARCHAR(50)
        ,S_DATA VARCHAR(50)
        ,S_QUANTITY SMALLINT )
    SELECT O_ID
        ,D_ID
        ,W_ID

```

```

        ,OL_NUMBER
        ,I_ID
        ,I_SUPPLY_W_ID
        ,OL_DELIVERY_D
        ,I_QTY
        ,TOTAL_PRICE
        ,OL_DIST_INFO
        ,I_PRICE,I_NAME,I_DATA,S_DATA,
S_QUANTITY
        FROM DATA
        ) AS INS
;
EXEC SQL DECLARE ISOL_Local_7 CURSOR FOR

WITH DATA AS ( SELECT O_ID
        ,D_ID
        ,W_ID
        ,OL_NUMBER
        ,I_ID
        ,W_ID AS I_SUPPLY_W_ID
        ,0 AS OL_DELIVERY_D
        ,I_QTY
        ,(I_PRICE * I_QTY) AS TOTAL_PRICE
        ,OL_DIST_INFO
        ,I_PRICE,I_NAME,I_DATA,S_DATA,S_QUANTITY

        FROM ( SELECT :next_o_id as O_ID
                ,:w_id AS W_ID
                ,:d_id as D_ID
                ,OL_NUMBER
                ,I_ID
                ,I_QTY
                FROM Table( VALUES
                    ( SMALLINT(1) ,:id0 ,:ol_quantity0 )
                    ,( SMALLINT(2) ,:id1 ,:ol_quantity1 )
                    ,( SMALLINT(3) ,:id2 ,:ol_quantity2 )
                    ,( SMALLINT(4) ,:id3 ,:ol_quantity3 )
                    ,( SMALLINT(5) ,:id4 ,:ol_quantity4 )
                    ,( SMALLINT(6) ,:id5 ,:ol_quantity5 )
                    ,( SMALLINT(7) ,:id6 ,:ol_quantity6 )
                ) AS X (OL_NUMBER , I_ID , I_QTY
        )

        ) AS ITEMLIST
        ,TABLE(NEW_OL_LOCAL( I_ID
                ,I_QTY
                ,W_ID
                ,O_ID
                ,D_ID
                )
        ) AS NEW_OL_LOCAL
        WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
    )

    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
    FROM NEW TABLE ( INSERT INTO ORDER_LINE
        ( OL_O_ID
        ,OL_D_ID
        ,OL_W_ID
        ,OL_NUMBER
        ,OL_I_ID
        ,OL_SUPPLY_W_ID
        ,OL_DELIVERY_D
        ,OL_QUANTITY
        ,OL_AMOUNT
        ,OL_DIST_INFO
        )
    INCLUDE ( I_PRICE INTEGER

```

```

, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_8 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT(1) , :id0 , :ol_quantity0 )
, ( SMALLINT(2) , :id1 , :ol_quantity1 )
, ( SMALLINT(3) , :id2 , :ol_quantity2 )
, ( SMALLINT(4) , :id3 , :ol_quantity3 )
, ( SMALLINT(5) , :id4 , :ol_quantity4 )
, ( SMALLINT(6) , :id5 , :ol_quantity5 )
, ( SMALLINT(7) , :id6 , :ol_quantity6 )
, ( SMALLINT(8) , :id7 , :ol_quantity7 )
) AS X (OL_NUMBER , I_ID , I_QTY
)
) AS ITEMLIST
, TABLE(NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER

```

```

, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_9 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT(1) , :id0 , :ol_quantity0 )
, ( SMALLINT(2) , :id1 , :ol_quantity1 )
, ( SMALLINT(3) , :id2 , :ol_quantity2 )
, ( SMALLINT(4) , :id3 , :ol_quantity3 )
, ( SMALLINT(5) , :id4 , :ol_quantity4 )
, ( SMALLINT(6) , :id5 , :ol_quantity5 )
, ( SMALLINT(7) , :id6 , :ol_quantity6 )
, ( SMALLINT(8) , :id7 , :ol_quantity7 )
, ( SMALLINT(9) , :id8 , :ol_quantity8 )
) AS X (OL_NUMBER , I_ID , I_QTY
)
) AS ITEMLIST
, TABLE(NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL

```

```

        WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
    )
    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
    S_QUANTITY
    FROM NEW TABLE ( INSERT INTO ORDER_LINE
        ( OL_O_ID
        , OL_D_ID
        , OL_W_ID
        , OL_NUMBER
        , OL_I_ID
        , OL_SUPPLY_W_ID
        , OL_DELIVERY_D
        , OL_QUANTITY
        , OL_AMOUNT
        , OL_DIST_INFO
        )
    INCLUDE( I_PRICE INTEGER
        , I_NAME CHAR(24)
        , I_DATA VARCHAR(50)
        , S_DATA VARCHAR(50)
        , S_QUANTITY SMALLINT )
    SELECT O_ID
        , D_ID
        , W_ID
        , OL_NUMBER
        , I_ID
        , I_SUPPLY_W_ID
        , OL_DELIVERY_D
        , I_QTY
        , TOTAL_PRICE
        , OL_DIST_INFO
        , I_PRICE, I_NAME, I_DATA, S_DATA,
    S_QUANTITY
    FROM DATA
    ) AS INS
;
EXEC SQL DECLARE ISOL_Local_10 CURSOR FOR

WITH DATA AS ( SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , W_ID AS I_SUPPLY_W_ID
    , 0 AS OL_DELIVERY_D
    , I_QTY
    , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

    FROM ( SELECT :next_o_id as O_ID
        , :w_id AS W_ID
        , :d_id as D_ID
        , OL_NUMBER
        , I_ID
        , I_QTY
    FROM Table( VALUES
        ( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
        , ( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
        , ( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
        , ( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
        , ( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
        , ( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
        , ( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
        , ( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
        , ( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
        , ( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
    )

```

```

        ) AS X ( OL_NUMBER , I_ID , I_QTY
    ) AS ITEMELIST
    , TABLE( NEW_OL_LOCAL( I_ID
        , I_QTY
        , W_ID
        , O_ID
        , D_ID
        )
    ) AS NEW_OL_LOCAL
    WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
    )
    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
    S_QUANTITY
    FROM NEW TABLE ( INSERT INTO ORDER_LINE
        ( OL_O_ID
        , OL_D_ID
        , OL_W_ID
        , OL_NUMBER
        , OL_I_ID
        , OL_SUPPLY_W_ID
        , OL_DELIVERY_D
        , OL_QUANTITY
        , OL_AMOUNT
        , OL_DIST_INFO
        )
    INCLUDE( I_PRICE INTEGER
        , I_NAME CHAR(24)
        , I_DATA VARCHAR(50)
        , S_DATA VARCHAR(50)
        , S_QUANTITY SMALLINT )
    SELECT O_ID
        , D_ID
        , W_ID
        , OL_NUMBER
        , I_ID
        , I_SUPPLY_W_ID
        , OL_DELIVERY_D
        , I_QTY
        , TOTAL_PRICE
        , OL_DIST_INFO
        , I_PRICE, I_NAME, I_DATA, S_DATA,
    S_QUANTITY
    FROM DATA
    ) AS INS
;
EXEC SQL DECLARE ISOL_Local_11 CURSOR FOR

WITH DATA AS ( SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , W_ID AS I_SUPPLY_W_ID
    , 0 AS OL_DELIVERY_D
    , I_QTY
    , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

    FROM ( SELECT :next_o_id as O_ID
        , :w_id AS W_ID
        , :d_id as D_ID
        , OL_NUMBER
        , I_ID
        , I_QTY
    FROM Table( VALUES
        ( SMALLINT( 1 ) , :id0 , :ol_quantity0 )

```

```

        , ( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
        , ( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
        , ( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
        , ( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
        , ( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
        , ( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
        , ( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
        , ( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
        , ( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
        , ( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
    ) AS X ( OL_NUMBER , I_ID , I_QTY
)
    ) AS ITEM LIST
    , TABLE( NEW_OL_LOCAL( I_ID
        , I_QTY
        , W_ID
        , O_ID
        , D_ID
    )
    ) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_12 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO

```

```

, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
, ( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
, ( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
, ( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
, ( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
, ( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
, ( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
, ( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
, ( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
, ( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
, ( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
, ( SMALLINT( 12 ) , :id11 , :ol_quantity11 )
) AS X ( OL_NUMBER , I_ID , I_QTY
) AS ITEM LIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS

```

```

;
EXEC SQL DECLARE ISOL_Local_13 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT(1) , :id0 , :ol_quantity0 )
, ( SMALLINT(2) , :id1 , :ol_quantity1 )
, ( SMALLINT(3) , :id2 , :ol_quantity2 )
, ( SMALLINT(4) , :id3 , :ol_quantity3 )
, ( SMALLINT(5) , :id4 , :ol_quantity4 )
, ( SMALLINT(6) , :id5 , :ol_quantity5 )
, ( SMALLINT(7) , :id6 , :ol_quantity6 )
, ( SMALLINT(8) , :id7 , :ol_quantity7 )
, ( SMALLINT(9) , :id8 , :ol_quantity8 )
, ( SMALLINT(10) , :id9 , :ol_quantity9 )
, ( SMALLINT(11) , :id10 , :ol_quantity10 )
, ( SMALLINT(12) , :id11 , :ol_quantity11 )
, ( SMALLINT(13) , :id12 , :ol_quantity12 )
) AS X (OL_NUMBER , I_ID , I_QTY

)
) AS ITEMLIST
, TABLE(NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
)
INCLUDE( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID

```

```

, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_14 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT(1) , :id0 , :ol_quantity0 )
, ( SMALLINT(2) , :id1 , :ol_quantity1 )
, ( SMALLINT(3) , :id2 , :ol_quantity2 )
, ( SMALLINT(4) , :id3 , :ol_quantity3 )
, ( SMALLINT(5) , :id4 , :ol_quantity4 )
, ( SMALLINT(6) , :id5 , :ol_quantity5 )
, ( SMALLINT(7) , :id6 , :ol_quantity6 )
, ( SMALLINT(8) , :id7 , :ol_quantity7 )
, ( SMALLINT(9) , :id8 , :ol_quantity8 )
, ( SMALLINT(10) , :id9 , :ol_quantity9 )
, ( SMALLINT(11) , :id10 , :ol_quantity10 )
, ( SMALLINT(12) , :id11 , :ol_quantity11 )
, ( SMALLINT(13) , :id12 , :ol_quantity12 )
, ( SMALLINT(14) , :id13 , :ol_quantity13 )
) AS X (OL_NUMBER , I_ID , I_QTY

)
) AS ITEMLIST
, TABLE(NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER

```

```

,OL_I_ID
,OL_SUPPLY_W_ID
,OL_DELIVERY_D
,OL_QUANTITY
,OL_AMOUNT
,OL_DIST_INFO
)
INCLUDE( I_PRICE INTEGER
,I_NAME CHAR(24)
,I_DATA VARCHAR(50)
,S_DATA VARCHAR(50)
,S_QUANTITY SMALLINT)
SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,OL_DELIVERY_D
,I_QTY
,TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE,I_NAME,I_DATA,S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_15 CURSOR FOR

WITH DATA AS ( SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,W_ID AS I_SUPPLY_W_ID
,0 AS OL_DELIVERY_D
,I_QTY
,(I_PRICE * I_QTY) AS TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE,I_NAME,I_DATA,S_DATA,S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
,:w_id AS W_ID
,:d_id as D_ID
,OL_NUMBER
,I_ID
,I_QTY
FROM Table( VALUES
( SMALLINT(1) ,:id0 ,:ol_quantity0 )
,( SMALLINT(2) ,:id1 ,:ol_quantity1 )
,( SMALLINT(3) ,:id2 ,:ol_quantity2 )
,( SMALLINT(4) ,:id3 ,:ol_quantity3 )
,( SMALLINT(5) ,:id4 ,:ol_quantity4 )
,( SMALLINT(6) ,:id5 ,:ol_quantity5 )
,( SMALLINT(7) ,:id6 ,:ol_quantity6 )
,( SMALLINT(8) ,:id7 ,:ol_quantity7 )
,( SMALLINT(9) ,:id8 ,:ol_quantity8 )
,( SMALLINT(10) ,:id9 ,:ol_quantity9 )
,( SMALLINT(11) ,:id10 ,:ol_quantity10 )
,( SMALLINT(12) ,:id11 ,:ol_quantity11 )
,( SMALLINT(13) ,:id12 ,:ol_quantity12 )
,( SMALLINT(14) ,:id13 ,:ol_quantity13 )
,( SMALLINT(15) ,:id14 ,:ol_quantity14 )
) AS X(OL_NUMBER ,I_ID ,I_QTY

)
) AS ITEMLIST
, TABLE(NEW_OL_LOCAL( I_ID
,I_QTY

```

```

,W_ID
,O_ID
,D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA ,
S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
,OL_D_ID
,OL_W_ID
,OL_NUMBER
,OL_I_ID
,OL_SUPPLY_W_ID
,OL_DELIVERY_D
,OL_QUANTITY
,OL_AMOUNT
,OL_DIST_INFO
)
INCLUDE( I_PRICE INTEGER
,I_NAME CHAR(24)
,I_DATA VARCHAR(50)
,S_DATA VARCHAR(50)
,S_QUANTITY SMALLINT)

SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,OL_DELIVERY_D
,I_QTY
,TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE,I_NAME,I_DATA,S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
// Start processing
in_neword = (struct in_neword_struct *) pin ;
neword = (struct out_neword_struct *) pout ;
#ifdef DEBUGIT
new_debug( neword, in_neword, "SP upon entry");
#endif
// Using I_PRICE == 0 as a flag to the client that the ITEM was not fetched
(hence bad).
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
in_neword->s_O_OL_CNT ; inputItemArrayIndex++)
{
i_priceArray[ inputItemArrayIndex ] = 0 ;
}
neword->deadlocks = -1 ;
retry_tran:
neword->deadlocks++ ;
EXEC SQL
SELECT D_TAX,D_NEXT_O_ID INTO :dist_tax , :next_o_id
FROM OLD TABLE ( UPDATE DISTRICT
SET D_NEXT_O_ID = D_NEXT_O_ID + 1
WHERE D_W_ID = :w_id
AND D_ID = :d_id
) AS OT
;

```

```

if( sqlca.sqlcode != 0 )
{
DLCHK( retry_tran );
sqlerror( NEWORD_SQL, "DISTRICT", __FILE__, __LINE__, &sqlca );
goto error;
}
// Invalid I_ID will give a +100, now that we've changed the cursor definitions
// to include a 'WHERE I_PRICE NOT NULL' clause.
#define NEW_CURSOR_OPEN_ERROR \
{
if( sqlca.sqlcode != 0 ) \
{ \
goto sql_error ; \
} \
}
#define NEW_CURSOR_ERROR \
{
if( sqlca.sqlcode == 0 ) \
{ \
neword->s_O_OL_CNT ++; \
} \
else \
if( sqlca.sqlcode == +100 ) \
{ \
break ; \
} \
else \
goto sql_error ; \
}
if( allLocal )
{
switch( inputItemCount )
{
case 1:
EXEC SQL OPEN ISOL_Local_1 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_1 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation , :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 2:
EXEC SQL OPEN ISOL_Local_2 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_2 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation , :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 3:
EXEC SQL OPEN ISOL_Local_3 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_3 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation , :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 4:
EXEC SQL OPEN ISOL_Local_4 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_4 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation , :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 5:
EXEC SQL OPEN ISOL_Local_5 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_5 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation , :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 6:
EXEC SQL OPEN ISOL_Local_6 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_6 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation , :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 7:
EXEC SQL OPEN ISOL_Local_7 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_7 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation , :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 8:
EXEC SQL OPEN ISOL_Local_8 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_8 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation , :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 9:
EXEC SQL OPEN ISOL_Local_9 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_9 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation , :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 10:
EXEC SQL OPEN ISOL_Local_10 ;
NEW_CURSOR_OPEN_ERROR

```



```

        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Local_10 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 11:
        EXEC SQL OPEN ISOL_Local_11 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Local_11 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 12:
        EXEC SQL OPEN ISOL_Local_12 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Local_12 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 13:
        EXEC SQL OPEN ISOL_Local_13 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Local_13 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 14:
        EXEC SQL OPEN ISOL_Local_14 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Local_14 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 15:
        EXEC SQL OPEN ISOL_Local_15 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Local_15 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    default:
        sqlerror(NEWORD_SQL, "Default switch on local
orderline/stock/index", __FILE__, __LINE__, &sqlca);
        goto ferror;
    }
}
else
{
    switch( inputItemCount )
    {
        case 1:
            EXEC SQL OPEN ISOL_Remote_1 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Remote_1 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
                NEW_CURSOR_ERROR
            }
            break ;
        case 2:
            EXEC SQL OPEN ISOL_Remote_2 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Remote_2 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
                NEW_CURSOR_ERROR
            }
            break ;
        case 3:
            EXEC SQL OPEN ISOL_Remote_3 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Remote_3 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
                NEW_CURSOR_ERROR
            }
            break ;
        case 4:
            EXEC SQL OPEN ISOL_Remote_4 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Remote_4 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
                NEW_CURSOR_ERROR
            }
            break ;
        case 5:
            EXEC SQL OPEN ISOL_Remote_5 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Remote_5 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
                NEW_CURSOR_ERROR
            }
            break ;
        case 6:
            EXEC SQL OPEN ISOL_Remote_6 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
            {

```

```

EXEC SQL FETCH ISOL_Remote_6 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 7:
EXEC SQL OPEN ISOL_Remote_7 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_7 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 8:
EXEC SQL OPEN ISOL_Remote_8 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_8 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 9:
EXEC SQL OPEN ISOL_Remote_9 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_9 INTO :item_price, :item_name,
:i_data, :stockDistrictInformation, :s_data, :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 10:
EXEC SQL OPEN ISOL_Remote_10 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_10 INTO :item_price,
:item_name, :i_data, :stockDistrictInformation, :s_data, :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 11:
EXEC SQL OPEN ISOL_Remote_11 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_11 INTO :item_price,
:item_name, :i_data, :stockDistrictInformation, :s_data, :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 12:
EXEC SQL OPEN ISOL_Remote_12 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_12 INTO :item_price,
:item_name, :i_data, :stockDistrictInformation, :s_data, :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 13:
EXEC SQL OPEN ISOL_Remote_13 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_13 INTO :item_price,
:item_name, :i_data, :stockDistrictInformation, :s_data, :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 14:
EXEC SQL OPEN ISOL_Remote_14 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_14 INTO :item_price,
:item_name, :i_data, :stockDistrictInformation, :s_data, :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 15:
EXEC SQL OPEN ISOL_Remote_15 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_15 INTO :item_price,
:item_name, :i_data, :stockDistrictInformation, :s_data, :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
default:
sqlerror(NEWORD_SQL, "Default switch on remote
orderline/stock/index", __FILE__, __LINE__, &sqlca);
goto ferror;
}
}
for ( inputItemArrayIndex = 0 ;
inputItemArrayIndex < in_neword->s_O_OL_CNT // from input
&& i_priceArray[ inputItemArrayIndex ] != 0 ;
inputItemArrayIndex++ )
{
// s_I_NAME, and s_S_QUANTITY already set as output host variables
neword->item[ inputItemArrayIndex ].s_I_PRICE = i_priceArray[
inputItemArrayIndex ] ;
if ( is_ORIGINAL( s_dataArray[ inputItemArrayIndex ].data,
s_dataArray[ inputItemArrayIndex ].len )
&& is_ORIGINAL( i_dataArray[ inputItemArrayIndex ].data,
i_dataArray[ inputItemArrayIndex ].len ) )
{
neword->item[ inputItemArrayIndex ].s_brand_generic = 'B';
}
else
{
neword->item[ inputItemArrayIndex ].s_brand_generic = 'G';
}
}
EXEC SQL
SELECT W_TAX, C_DISCOUNT, C_LAST, C_CREDIT
INTO :ware_tax, :c_discount, :c_last, :c_credit

```

```

FROM TABLE (NEW_WH ( :next_o_id
                ,:w_id
                ,:d_id
                ,:c_id
                ,:o_entry_d
                ,:inputItemCount
                ,:allLocal
            )
        ) AS NEW_WH_TABLE
;
if( sqlca.sqlcode == 0 )
{
    if( neword->s_O_OL_CNT == in_neword->s_O_OL_CNT )
    {
        neword->s_transtatus = TRAN_OK ;
        EXEC SQL COMMIT;

        if( sqlca.sqlcode != 0 )
        {
            sqlerror(NEWORD_SQL, "COMMIT", __FILE__, __LINE__, &sqlca )
        }
        goto ferror;
    }
}
else
{
    neword->s_transtatus = INVALID_ITEM ;

    EXEC SQL ROLLBACK WORK ;

    if( sqlca.sqlcode != 0 )
    {
        neword->s_transtatus = FATAL_SQLERROR;
        sqlerror(NEWORD_SQL, "ROLLBACK FAILED (INVALID ITEM)",
        __FILE__, __LINE__, &sqlca);
        // no point in ferror
    }
}
else
{
    DLCHK( retry_tran );
    sqlerror( NEWORD_SQL, "NEW_WH", __FILE__, __LINE__, &sqlca);
    goto ferror;
}
/*-----*/
/* Return to client */
/*-----*/
mexit:
if( sqlca.sqlcode >= 0 )
{
    storedProcRc = SQLZ_HOLD_PROC ;
}
else
{
    storedProcRc = SQLZ_DISCONNECT_PROC ;
}
#ifdef DEBUGIT
    new_debug( neword, in_neword, "SP prior to return");
#endif
return ( storedProcRc ) ;
sql_error:
{
    char tempstr[ 4096 ] ;
    DLCHK( retry_tran ) ;
    sprintf( tempstr, "inputItemCount=%d, :next_o_id=%d, :d_id=%d,
:w_id=%d", inputItemCount, next_o_id, d_id, w_id ) ;
    sqlerror( NEWORD_SQL, tempstr, __FILE__, __LINE__, &sqlca ) ;
}

```

```

}
ferror:
neword->s_transtatus = FATAL_SQLERROR;
EXEC SQL ROLLBACK WORK;
if( sqlca.sqlcode != 0 )
{
    sqlerror( NEWORD_SQL, "ROLLBACK FAILED", __FILE__, __LINE__
, &sqlca ) ;
}
}
goto mexit ;
}
/*
** A little function to search for the string "ORIGINAL" given a string and
** it's length
*/
static unsigned char skip[256] = {8,8,8,8,8,8,8,8, /*0-9*/
                                8,8,8,8,8,8,8,8, /*10-19*/
                                8,8,8,8,8,8,8,8, /*20-29*/
                                8,8,8,8,8,8,8,8, /*30-39*/
                                8,8,8,8,8,8,8,8, /*40-49*/
                                8,8,8,8,8,8,8,8, /*50-59*/
                                8,8,8,8,1,8,8,8, /*60-69*/
                                8,4,8,3,8,8,0,8,2,7, /*70-79*/
                                8,8,6,8,8,8,8,8,8, /*80-89*/
                                8,8,8,8,8,8,8,8,8, /*90-99*/
                                8,8,8,8,8,8,8,8,8, /*100-109*/
                                8,8,8,8,8,8,8,8,8, /*110-119*/
                                8,8,8,8,8,8,8,8,8, /*120-129*/
                                8,8,8,8,8,8,8,8,8, /*130-139*/
                                8,8,8,8,8,8,8,8,8, /*140-149*/
                                8,8,8,8,8,8,8,8,8, /*150-159*/
                                8,8,8,8,8,8,8,8,8, /*160-169*/
                                8,8,8,8,8,8,8,8,8, /*170-179*/
                                8,8,8,8,8,8,8,8,8, /*180-189*/
                                8,8,8,8,8,8,8,8,8, /*190-199*/
                                8,8,8,8,8,8,8,8,8, /*200-209*/
                                8,8,8,8,8,8,8,8,8, /*210-219*/
                                8,8,8,8,8,8,8,8,8, /*220-229*/
                                8,8,8,8,8,8,8,8,8, /*230-239*/
                                8,8,8,8,8,8,8,8,8, /*240-249*/
                                8,8,8,8,8}; /*250-254*/
static int is_ORIGINAL( char *string, short length )
{
    char *cur_string;
    char *end_string;
    unsigned char *skips;
    int skip_dist;
    int result = 0;
    cur_string = string+7;
    end_string = string + length;
    skips = skip;
    while (cur_string < end_string)
    {
        skip_dist = skips[*cur_string];
        while ( (skip_dist > 0) && (cur_string < end_string) )
        {
            skip_dist = skips[*cur_string += skip_dist];
        }
        if (cur_string >= end_string)
            goto exit;
        if ( cur_string[-4] != 'G' )
            goto noMatch;
        if ( memcmp( cur_string-7, "ORIGINAL", 8 ) == 0 )
        {
            result = 1;
            goto exit;
        }
    }
}

```

```

noMatch:
    cur_string += 8;
    } /* end while */
exit:
    return ( result );
}
// -----
// Order Status SERVER
// -----
#undef w_id
#undef d_id
#undef c_id_input
#undef o_id
#undef o_entry_d
#undef o_carrier_d
#undef c_id
#undef c_first
#undef c_middle
#undef c_last
#undef c_balance
SQL_API_RC order_status_internal( char *pin, char *pout )
{
    struct in_ordstat_struct * in_ordstat = (struct in_ordstat_struct *) pin ;
    struct out_ordstat_struct * ordstat = (struct out_ordstat_struct *) pout ;
    struct sqlca sqlca ;
    EXEC SQL BEGIN DECLARE SECTION;
    // From input values
    ///sqlint32 w_id ;
    ///short d_id;
    sqlint32 c_id_input ;
    struct s_data_type { short len ; char data[ 16 ] ; } c_last_input ;
    // From queries
    // From initial query
    sqlint32 o_id ;
    ///sqlint32 c_id ;
    short o_carrier_id ;
    ///sqlint64 o_entry_d ;
    char c_first[ 16 ] ;
    char c_middle[ 2 ] ;
    ///char c_last[ 16 ] ;
    sqlint64 c_balance ;
    // From cursor
    sqlint32 ol_i_id ;
    sqlint32 ol_supply_w_id ;
    short ol_quantity ;
    sqlint32 ol_amount ;
    sqlint64 ol_delivery_d ;
    EXEC SQL END DECLARE SECTION;
    // NOTE: this varchar would normally live inside the declare section
    // but this package already delcared the same field higher up. Need the field
    // within this scope though.
    ///struct s_data_type { short len ; char data[ 16 ] ; } c_last_input ;
    int storedProcRc ;
    int itemArrayIndex = 0 ;
    #define w_id in_ordstat->s_W_ID ;
    #define d_id in_ordstat->s_D_ID ;
    #define c_id_input in_ordstat->s_C_ID
    #define o_id ordstat->s_O_ID
    #define o_entry_d ordstat->s_O_ENTRY_D_time
    #define o_carrier_id ordstat->s_O_CARRIER_ID
    #define c_id ordstat->s_C_ID
    #define c_first ordstat->s_C_FIRST
    #define c_middle ordstat->s_C_MIDDLE
    #define c_last ordstat->s_C_LAST
    #define c_balance ordstat->s_C_BALANCE
    EXEC SQL DECLARE read_orderline_cur CURSOR FOR
    SELECT OL_I_ID, OL_SUPPLY_W_ID, OL_QUANTITY,
    OL_AMOUNT, OL_DELIVERY_D

```

```

FROM ORDER_LINE
WHERE OL_W_ID = :w_id
AND OL_D_ID = :d_id
AND OL_O_ID = :o_id
FOR FETCH ONLY ;
ordstat->deadlocks = -1 ;
#endif
ord_debug(ordstat, in_ordstat, "SP upon entry");
#endif
retry_tran:
    ordstat->deadlocks ++ ;
    if ( c_id_input == 0 )
    {
        c_last_input.len = strlen( in_ordstat->s_C_LAST ) ;
        memcpy( c_last_input.data , in_ordstat->s_C_LAST , c_last_input.len ) ;
        EXEC SQL
        SELECT O_ID, O_CARRIER_ID, O_ENTRY_D, C_BALANCE,
        C_FIRST, C_MIDDLE, C_ID
        INTO :o_id, :o_carrier_id, :o_entry_d, :c_balance, :c_first, :c_middle,
        :c_id

        FROM TABLE ( ORD_C_LAST( :w_id
        , :d_id
        , :c_last_input
        )
        ) AS ORD_C_LAST
        ;
    }
    else
    {
        EXEC SQL
        SELECT O_ID, O_CARRIER_ID, O_ENTRY_D, C_BALANCE,
        C_FIRST, C_MIDDLE, C_LAST
        INTO :o_id, :o_carrier_id, :o_entry_d, :c_balance, :c_first, :c_middle,
        :c_last

        FROM TABLE ( ORD_C_ID( :w_id
        , :d_id
        , :c_id_input
        )
        ) AS ORD_C_ID
        ;
    }
    if ( sqlca.sqlcode != 0 )
    {
        DLCHK( retry_tran );
        sqlerror( ORDSTAT_SQL, "READ CUST and ORDERS", __FILE__,
        __LINE__, &sqlca );
        goto ferror;
    }
    /*-----*/
    /* Read ORDER_LINES */
    /*-----*/
    EXEC SQL OPEN read_orderline_cur ;
    if ( sqlca.sqlcode != 0 )
    {
        DLCHK( retry_tran );
        sqlerror(ORDSTAT_SQL, "OPEN CURSOR read_orderline_cur",
        __FILE__, __LINE__, &sqlca ) ;
        goto ferror;
    }
    itemArrayIndex = 0 ;
    {
        do
        {
            EXEC SQL FETCH read_orderline_cur
            INTO :ol_i_id, :ol_supply_w_id, :ol_quantity, :ol_amount,
            :ol_delivery_d ;

```

```

if( sqlca.sqlcode == 0 )
{
    ordstat->item[ itemArrayIndex ].s_OL_I_ID      = ol_i_id ;
    ordstat->item[ itemArrayIndex ].s_OL_SUPPLY_W_ID =
ol_supply_w_id ;
    ordstat->item[ itemArrayIndex ].s_OL_QUANTITY  = ol_quantity ;
    ordstat->item[ itemArrayIndex ].s_OL_AMOUNT   = ol_amount ;
    ordstat->item[ itemArrayIndex ].s_OL_DELIVERY_D_time =
ol_delivery_d ;

    itemArrayIndex++;
}
else
if( sqlca.sqlcode < 0 )
{
    DLCHK( retry_tran ) ;
    sqlerror( ORDSTAT_SQL, "FETCH CURSOR read_orderline_cur",
__FILE__, __LINE__, &sqlca ) ;
    goto ferror ;
}
}
while ( sqlca.sqlcode == 0 ) ;
}
ordstat->s_ol_cnt = itemArrayIndex ;
EXEC SQL COMMIT ;
if( sqlca.sqlcode == 0 )
{
    ordstat->s_transtatus = TRAN_OK ;
}
else
{
    DLCHK( retry_tran ) ;
    sqlerror(ORDSTAT_SQL, "COMMIT", __FILE__, __LINE__, &sqlca);
    goto ferror ;
}
mexit:
if ( sqlca.sqlcode >= 0 )
{
    storedProcRc = SQLZ_HOLD_PROC ;
}
else
{
    storedProcRc = SQLZ_DISCONNECT_PROC ;
}
#ifdef DEBUGIT
    ord_debug(ordstat, in_ordstat, "SP prior to return");
#endif
return ( storedProcRc ) ;
ferror:
ordstat->s_transtatus = FATAL_SQLERROR ;
EXEC SQL ROLLBACK WORK ;
if( sqlca.sqlcode != 0 )
{
    sqlerror(ORDSTAT_SQL, "ROLLBACK FAILED", __FILE__, __LINE__,
&sqlca);
}
goto mexit;
}
// -----
// Delivery SERVER
// -----

#undef d_id
#undef c_id
#undef w_id
#undef o_carrier_id
#undef ol_delivery_d

```

```

SQL_API_RC delivery_internal ( char * pin, char * pout )
{
    struct in_delivery_struct * in_delivery = (struct in_delivery_struct *) pin ;
    struct out_delivery_struct * delivery = (struct out_delivery_struct *) pout ;
    struct sqlca sqlca ;
    int storedProcRc ;
    short district_id ;
    sqlint32 customer_id ;

    EXEC SQL BEGIN DECLARE SECTION;
    // input
    ///sqlint32 w_id ;
    ///short d_id ;
    ///sqlint32 c_id ;
    ///short o_carrier_id ;
    ///sqlint64 ol_delivery_d ;
    // output
    short no_o_id_indicator = 0 ;
    sqlint32 no_o_id ;
    EXEC SQL END DECLARE SECTION;
#define d_id      district_id
#define c_id      customer_id
#define w_id      in_delivery->s_W_ID
#define o_carrier_id in_delivery->s_O_CARRIER_ID
#define ol_delivery_d in_delivery->s_O_DELIVERY_D_time
    delivery->deadlocks = -1 ;
#ifdef DEBUGIT
    del_debug( delivery, in_delivery, "SP upon entry");
#endif
    // Deadlock Handling
    // -----
    // Since we COMMIT inside the for() loop, we must take special
    // care while handling deadlocks. This is best explained by
    // an example.
    //
    // Assume we deadlock on d_id=6. This means that an order from the
    // first 5 districts have already been delivered. We will then
    // restart the loop (retry_tran). However, the loop will restart
    // at d_id = 1! This means that the second (and all subsequent)
    // time through the loop, we will deliver orders for districts that
    // have already been delivered, with the net result being more than
    // 10 orders being delivered.
    //
    // The solution to this problem is to initialize the starting point
    // of the loop *before* the retry_tran label. This will ensure that
    // if we deadlock, we will restart the loop with the same district
    // that we deadlocked on, and we won't deliver any extra orders.
    //
    // NOTE: If we ever change this back to one COMMIT per transaction
    // (instead of one COMMIT per iteration), then the initialization
    // of d_id must be moved back into the for loop. (A rollback due
    // to deadlock in this case would rollback all delivered orders so
    // far, so we'd need to re-deliver them all on the next iteration.)
    d_id = 1;
retry_tran:
    delivery->deadlocks++;
    for ( ; d_id <= DISTRICTS_PER_WAREHOUSE ; d_id++)
    {
        no_o_id = 0 ;
        no_o_id_indicator = 0 ;
        EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC
        SELECT O_ID

                INTO :no_o_id :no_o_id_indicator

                FROM TABLE ( DEL(:w_id, :d_id, :o_carrier_id, :ol_delivery_d) )
    AS T ;

```

```

        COMMIT ;
    END COMPOUND ;

if ( sqlca.sqlcode == 0 )
{
    /* Refer to clause 2.7.4.2, bullet 3 in spec.*/
    /* Need to report if more than 1 or 1% of */
    /* no_o_id will remain 0 if null returned, so just treat the same way */
    delivery->s_O_ID[ d_id - 1 ] = no_o_id ;
}
else
{
    DLCHK( retry_tran );

    sqlerror( DELIVERY_SQL , "DELIVERY", __FILE__ , __LINE__ ,
&sqlca);
    goto ferror ;
}

delivery->s_transtatus = TRAN_OK ;
mexit:
if ( sqlca.sqlcode >= 0 )
{
    storedProcRc = SQLZ_HOLD_PROC ;
}
else
{
    storedProcRc = SQLZ_DISCONNECT_PROC ;
}
#ifdef DEBUGIT
    del_debug( delivery, in_delivery, "SP prior to return");
#endif
return ( storedProcRc ) ;
ferror:
delivery->s_transtatus = FATAL_SQLERROR ;
EXEC SQL ROLLBACK WORK ;
if ( sqlca.sqlcode != 0 )
{
    sqlerror( DELIVERY_SQL, "ROLLBACK FAILED", __FILE__ , __LINE__
, &sqlca ) ;
}

goto mexit ;
}
// -----
// Stored Procedure Stubs
// -----
SQL_API_RC SQL_API_FN news( char *pin, char *pout )
{
    return new_order_internal( pin, pout ) ;
}
SQL_API_RC SQL_API_FN ords( char *pin, char *pout )
{
    return order_status_internal( pin, pout ) ;
}
SQL_API_RC SQL_API_FN dels ( char * pin, char * pout )
{
    return delivery_internal( pin, pout ) ;
}

```

Src.Srv/uncat_func.ddl

```

-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.

```

```

--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
-----
-- uncat-func.ddl - Drop table function DDL
--
-- STOCK LEVEL
DROP SPECIFIC FUNCTION STOCK_LEVEL $
-- DELIVERY
DROP SPECIFIC FUNCTION DELIVERY $
-- ORDER STATUS
DROP SPECIFIC FUNCTION ORD_C_LAST $
DROP SPECIFIC FUNCTION ORD_C_ID $
-- PAYMENT
DROP SPECIFIC FUNCTION PAY_C_LAST $
DROP SPECIFIC FUNCTION PAY_C_ID $
-- NEW ORDER
DROP SPECIFIC FUNCTION NEW_OL_ALL $
DROP SPECIFIC FUNCTION NEW_OL_LOCAL $
DROP SPECIFIC FUNCTION NEW_WH $

```

Src.Srv/uncat_proc.ddl

```

DROP PROCEDURE news
    (varchar(270),varchar(662));
DROP PROCEDURE news;

DROP PROCEDURE pays;
DROP PROCEDURE ords
    (varchar(42),varchar(446));
DROP PROCEDURE ords;
DROP PROCEDURE dels
    (varchar(22),varchar(50));
DROP PROCEDURE dels;
DROP PROCEDURE stks
    (varchar(18),varchar(14));
DROP PROCEDURE stks;

```

Src.Srv/rpctpcc.def

```

LIBRARY rpctpcc
DESCRIPTION "Library of TPC-C Transactions (Stored Procedures)"
EXPORTS
news
ords
dels

```

utils/EXPLAIN.ddl

```

-- *- sql *-
--
-- Sample DDL to create Explain tables for Version 5.0
--
-- -> assumes db2start issued
-- -> assumes connection to a database exists
-- -> assumes called by "db2 -tf EXPLAIN.DDL"
--
--
-- To remind users how to use this file!
--
ECHO          ;
ECHO ***** IMPORTANT ***** ;

```

```

ECHO          ;
ECHO USAGE: db2 -tf EXPLAIN.DDL ;
ECHO          ;
ECHO ***** IMPORTANT ***** ;
ECHO          ;
ECHO          ;
--
--
-- Set autocommit off
--
UPDATE COMMAND OPTIONS USING C OFF;
--
-- EXPLAIN INSTANCE
--
-- (must be defined first due to referential integrity defintions)
--
CREATE TABLE EXPLAIN_INSTANCE ( EXPLAIN_REQUESTER
VARCHAR(128) NOT NULL,
        EXPLAIN_TIME  TIMESTAMP NOT NULL,
        SOURCE_NAME   VARCHAR(128) NOT NULL,
        SOURCE_SCHEMA VARCHAR(128) NOT NULL,
        SOURCE_VERSION VARCHAR(64) NOT NULL,
        EXPLAIN_OPTION CHAR(1) NOT NULL,
        SNAPSHOT_TAKEN CHAR(1) NOT NULL,
        DB2_VERSION   CHAR(7) NOT NULL,
        SQL_TYPE      CHAR(1) NOT NULL,
        QUERYOPT      INTEGER NOT NULL,
        BLOCK         CHAR(1) NOT NULL,
        ISOLATION     CHAR(2) NOT NULL,
        BUFPAGE       INTEGER NOT NULL,
        AVG_APPLS     INTEGER NOT NULL,
        SORTHEAP      INTEGER NOT NULL,
        LOCKLIST      INTEGER NOT NULL,
        MAXLOCKS      SMALLINT NOT NULL,
        LOCKS_AVAIL   INTEGER NOT NULL,
        CPU_SPEED     DOUBLE NOT NULL,
        REMARKS       VARCHAR(254),
        DBHEAP        INTEGER NOT NULL,
        COMM_SPEED    DOUBLE NOT NULL,
        PARALLELISM   CHAR(2) NOT NULL,
        DATAJOINER   CHAR(1) NOT NULL,
        PRIMARY KEY (EXPLAIN_REQUESTER,
        EXPLAIN_TIME,
        SOURCE_NAME,
        SOURCE_SCHEMA,
        SOURCE_VERSION))

IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_STATEMENT
--
CREATE TABLE EXPLAIN_STATEMENT ( EXPLAIN_REQUESTER
VARCHAR(128) NOT NULL,
        EXPLAIN_TIME  TIMESTAMP NOT NULL,
        SOURCE_NAME   VARCHAR(128) NOT NULL,
        SOURCE_SCHEMA VARCHAR(128) NOT NULL,
        SOURCE_VERSION VARCHAR(64) NOT NULL,
        EXPLAIN_LEVEL CHAR(1) NOT NULL,
        STMTNO        INTEGER NOT NULL,
        SECTNO        INTEGER NOT NULL,
        QUERYNO       INTEGER NOT NULL,
        QUERYTAG      CHAR(20) NOT NULL,
        STATEMENT_TYPE CHAR(2) NOT NULL,
        UPDATABLE     CHAR(1) NOT NULL,
        DELETABLE     CHAR(1) NOT NULL,
        TOTAL_COST    DOUBLE NOT NULL,
        STATEMENT_TEXT CLOB(2M) NOT NULL NOT
LOGGED,

```

```

        SNAPSHOT      BLOB(10M) NOT LOGGED,
        QUERY_DEGREE  INTEGER NOT NULL,
        PRIMARY KEY (EXPLAIN_REQUESTER,
        EXPLAIN_TIME,
        SOURCE_NAME,
        SOURCE_SCHEMA,
        SOURCE_VERSION,
        EXPLAIN_LEVEL,
        STMTNO,
        SECTNO),
        FOREIGN KEY (EXPLAIN_REQUESTER,
        EXPLAIN_TIME,
        SOURCE_NAME,
        SOURCE_SCHEMA,
        SOURCE_VERSION)
        REFERENCES EXPLAIN_INSTANCE
        ON DELETE CASCADE)

IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_ARGUMENTS
--
CREATE TABLE EXPLAIN_ARGUMENT ( EXPLAIN_REQUESTER
VARCHAR(128) NOT NULL,
        EXPLAIN_TIME  TIMESTAMP NOT NULL,
        SOURCE_NAME   VARCHAR(128) NOT NULL,
        SOURCE_SCHEMA VARCHAR(128) NOT NULL,
        SOURCE_VERSION VARCHAR(64) NOT NULL,
        EXPLAIN_LEVEL CHAR(1) NOT NULL,
        STMTNO        INTEGER NOT NULL,
        SECTNO        INTEGER NOT NULL,
        OPERATOR_ID   INTEGER NOT NULL,
        ARGUMENT_TYPE CHAR(8) NOT NULL,
        ARGUMENT_VALUE VARCHAR(1024),
        LONG_ARGUMENT_VALUE CLOB(2M) NOT
LOGGED,
        FOREIGN KEY (EXPLAIN_REQUESTER,
        EXPLAIN_TIME,
        SOURCE_NAME,
        SOURCE_SCHEMA,
        SOURCE_VERSION,
        EXPLAIN_LEVEL,
        STMTNO,
        SECTNO)
        REFERENCES EXPLAIN_STATEMENT
        ON DELETE CASCADE)

IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_OBJECT
--
CREATE TABLE EXPLAIN_OBJECT ( EXPLAIN_REQUESTER
VARCHAR(128) NOT NULL,
        EXPLAIN_TIME  TIMESTAMP NOT NULL,
        SOURCE_NAME   VARCHAR(128) NOT NULL,
        SOURCE_SCHEMA VARCHAR(128) NOT NULL,
        SOURCE_VERSION VARCHAR(64) NOT NULL,
        EXPLAIN_LEVEL CHAR(1) NOT NULL,
        STMTNO        INTEGER NOT NULL,
        SECTNO        INTEGER NOT NULL,
        OBJECT_SCHEMA VARCHAR(128) NOT NULL,
        OBJECT_NAME   VARCHAR(128) NOT NULL,
        OBJECT_TYPE   CHAR(2) NOT NULL,
        CREATE_TIME   TIMESTAMP,
        STATISTICS_TIME TIMESTAMP,
        COLUMN_COUNT  SMALLINT NOT NULL,
        ROW_COUNT     BIGINT NOT NULL,
        WIDTH         INTEGER NOT NULL,

```

```

PAGES          INTEGER NOT NULL,
DISTINCT       CHAR(1) NOT NULL,
TABLESPACE_NAME VARCHAR(128),
OVERHEAD       DOUBLE NOT NULL,
TRANSFER_RATE  DOUBLE NOT NULL,
PREFETCHSIZE   INTEGER NOT NULL,
EXTENTS_SIZE   INTEGER NOT NULL,
CLUSTER        DOUBLE NOT NULL,
NLEAF          INTEGER NOT NULL,
NLEVELS        INTEGER NOT NULL,
FULLKEYCARD    BIGINT NOT NULL,
OVERFLOW       INTEGER NOT NULL,
FIRSTKEYCARD   BIGINT NOT NULL,
FIRST2KEYCARD  BIGINT NOT NULL,
FIRST3KEYCARD  BIGINT NOT NULL,
FIRST4KEYCARD  BIGINT NOT NULL,
SEQUENTIAL_PAGES INTEGER NOT NULL,
DENSITY        INTEGER NOT NULL,
STATS_SRC      CHAR(1) NOT NULL,
AVERAGE_SEQUENCE_GAP DOUBLE NOT
NULL,
AVERAGE_SEQUENCE_FETCH_GAP DOUBLE NOT
NULL,
AVERAGE_SEQUENCE_PAGES DOUBLE NOT
NULL,
AVERAGE_SEQUENCE_FETCH_PAGES DOUBLE
NOT NULL,
AVERAGE_RANDOM_PAGES DOUBLE NOT
NULL,
AVERAGE_RANDOM_FETCH_PAGES DOUBLE NOT
NULL,
NUMRIDS        BIGINT NOT NULL,
NUMRIDS_DELETED BIGINT NOT NULL,
NUM_EMPTY_LEAFS BIGINT NOT NULL,
ACTIVE_BLOCKS  BIGINT NOT NULL,
FOREIGN KEY (EXPLAIN_REQUESTER,
EXPLAIN_TIME,
SOURCE_NAME,
SOURCE_SCHEMA,
SOURCE_VERSION,
EXPLAIN_LEVEL,
STMTNO,
SECTNO)
REFERENCES EXPLAIN_STATEMENT
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_OPERATOR
--
CREATE TABLE EXPLAIN_OPERATOR ( EXPLAIN_REQUESTER
VARCHAR(128) NOT NULL,
EXPLAIN_TIME TIMESTAMP NOT NULL,
SOURCE_NAME VARCHAR(128) NOT NULL,
SOURCE_SCHEMA VARCHAR(128) NOT NULL,
SOURCE_VERSION VARCHAR(64) NOT NULL,
EXPLAIN_LEVEL CHAR(1) NOT NULL,
STMTNO INTEGER NOT NULL,
SECTNO INTEGER NOT NULL,
OPERATOR_ID INTEGER NOT NULL,
OPERATOR_TYPE CHAR(6) NOT NULL,
TOTAL_COST DOUBLE NOT NULL,
IO_COST DOUBLE NOT NULL,
CPU_COST DOUBLE NOT NULL,
FIRST_ROW_COST DOUBLE NOT NULL,
RE_TOTAL_COST DOUBLE NOT NULL,
RE_IO_COST DOUBLE NOT NULL,
RE_CPU_COST DOUBLE NOT NULL,
COMM_COST DOUBLE NOT NULL,
FIRST_COMM_COST DOUBLE NOT NULL,
BUFFERS DOUBLE NOT NULL,
REMOTE_TOTAL_COST DOUBLE NOT NULL,
REMOTE_COMM_COST DOUBLE NOT NULL,
FOREIGN KEY (EXPLAIN_REQUESTER,
EXPLAIN_TIME,
SOURCE_NAME,
SOURCE_SCHEMA,
SOURCE_VERSION,
EXPLAIN_LEVEL,
STMTNO,
SECTNO)
REFERENCES EXPLAIN_STATEMENT
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_PREDICATE
--
CREATE TABLE EXPLAIN_PREDICATE ( EXPLAIN_REQUESTER
VARCHAR(128) NOT NULL,
EXPLAIN_TIME TIMESTAMP NOT NULL,
SOURCE_NAME VARCHAR(128) NOT NULL,
SOURCE_SCHEMA VARCHAR(128) NOT NULL,
SOURCE_VERSION VARCHAR(64) NOT NULL,
EXPLAIN_LEVEL CHAR(1) NOT NULL,
STMTNO INTEGER NOT NULL,
SECTNO INTEGER NOT NULL,
OPERATOR_ID INTEGER NOT NULL,
PREDICATE_ID INTEGER NOT NULL,
HOW_APPLIED CHAR(5) NOT NULL,
WHEN_EVALUATED CHAR(3) NOT NULL,
RELOP_TYPE CHAR(2) NOT NULL,
SUBQUERY CHAR(1) NOT NULL,
FILTER_FACTOR DOUBLE NOT NULL,
PREDICATE_TEXT CLOB(2M) NOT LOGGED,
FOREIGN KEY (EXPLAIN_REQUESTER,
EXPLAIN_TIME,
SOURCE_NAME,
SOURCE_SCHEMA,
SOURCE_VERSION,
EXPLAIN_LEVEL,
STMTNO,
SECTNO)
REFERENCES EXPLAIN_STATEMENT
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_STREAM
--
CREATE TABLE EXPLAIN_STREAM ( EXPLAIN_REQUESTER
VARCHAR(128) NOT NULL,
EXPLAIN_TIME TIMESTAMP NOT NULL,
SOURCE_NAME VARCHAR(128) NOT NULL,
SOURCE_SCHEMA VARCHAR(128) NOT NULL,
SOURCE_VERSION VARCHAR(64) NOT NULL,
EXPLAIN_LEVEL CHAR(1) NOT NULL,
STMTNO INTEGER NOT NULL,
SECTNO INTEGER NOT NULL,
STREAM_ID INTEGER NOT NULL,
SOURCE_TYPE CHAR(1) NOT NULL,
SOURCE_ID INTEGER NOT NULL,
TARGET_TYPE CHAR(1) NOT NULL,
TARGET_ID INTEGER NOT NULL,
OBJECT_SCHEMA VARCHAR(128),
OBJECT_NAME VARCHAR(128),

```



```

STREAM_COUNT DOUBLE NOT NULL,
COLUMN_COUNT SMALLINT NOT NULL,
PREDICATE_ID INTEGER NOT NULL,
COLUMN_NAMES CLOB(2M) NOT LOGGED,
PMID SMALLINT NOT NULL,
SINGLE_NODE CHAR(5),
PARTITION_COLUMNS CLOB(2M) NOT LOGGED,
FOREIGN KEY (EXPLAIN_REQUESTER,
EXPLAIN_TIME,
SOURCE_NAME,
SOURCE_SCHEMA,
SOURCE_VERSION,
EXPLAIN_LEVEL,
STMTNO,
SECTNO)
REFERENCES EXPLAIN_STATEMENT
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- ADVISE TABLES
--
-- ADVISE_INSTANCE
--
-- (must be defined first due to referential integrity defintiions)
--
CREATE TABLE ADVISE_INSTANCE (
START_TIME TIMESTAMP NOT NULL WITH DEFAULT
CURRENT_TIMESTAMP,
END_TIME TIMESTAMP NOT NULL WITH DEFAULT
CURRENT_TIMESTAMP,
MODE VARCHAR(4) NOT NULL WITH DEFAULT "",
WKLD_COMPRESSION CHAR(4) NOT NULL WITH DEFAULT
'NONE',
STATUS CHAR(9) NOT NULL WITH DEFAULT "",
PRIMARY KEY (START_TIME))
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- ADVISE_INDEX
--
CREATE TABLE ADVISE_INDEX(
EXPLAIN_REQUESTER VARCHAR(128) NOT NULL WITH
DEFAULT "",
EXPLAIN_TIME TIMESTAMP NOT NULL WITH DEFAULT
CURRENT_TIMESTAMP,
SOURCE_NAME VARCHAR(128) NOT NULL WITH DEFAULT "",
SOURCE_SCHEMA VARCHAR(128) NOT NULL WITH DEFAULT
",
SOURCE_VERSION VARCHAR(64) NOT NULL WITH DEFAULT
",
EXPLAIN_LEVEL CHAR(1) NOT NULL WITH DEFAULT "",
STMTNO INTEGER NOT NULL WITH DEFAULT 0,
SECTNO INTEGER NOT NULL WITH DEFAULT 0,
QUERYNO INTEGER NOT NULL WITH DEFAULT 0,
QUERYTAG CHAR(20) NOT NULL WITH DEFAULT "",
NAME VARCHAR(128) NOT NULL,
CREATOR VARCHAR(128) NOT NULL WITH DEFAULT "",
TBNAME VARCHAR(128) NOT NULL,
TBCREATOR VARCHAR(128) NOT NULL WITH DEFAULT "",
COLNAMES CLOB(2M) NOT NULL,
UNIQUERULE CHAR(1) NOT NULL WITH DEFAULT "",
COLCOUNT SMALLINT NOT NULL WITH DEFAULT 0,
IID SMALLINT NOT NULL WITH DEFAULT 0,
NLEAF INTEGER NOT NULL WITH DEFAULT 0,
NLEVELS SMALLINT NOT NULL WITH DEFAULT 0,
FIRSTKEYCARD BIGINT NOT NULL WITH DEFAULT 0,
FULLKEYCARD BIGINT NOT NULL WITH DEFAULT 0,
CLUSTERRATIO SMALLINT NOT NULL WITH DEFAULT 0,
CLUSTERFACTOR DOUBLE NOT NULL WITH DEFAULT 0,
USERDEFINED SMALLINT NOT NULL WITH DEFAULT 0,
SYSTEM_REQUIRED SMALLINT NOT NULL WITH DEFAULT
0,
CREATE_TIME TIMESTAMP NOT NULL WITH DEFAULT
CURRENT_TIMESTAMP,
STATS_TIME TIMESTAMP WITH DEFAULT CURRENT
TIMESTAMP,
PAGE_FETCH_PAIRS VARCHAR(254) NOT NULL WITH
DEFAULT "",
REMARKS VARCHAR(254) WITH DEFAULT "",
DEFINER VARCHAR(128) NOT NULL WITH DEFAULT "",
CONVERTED CHAR(1) NOT NULL WITH DEFAULT "",
SEQUENTIAL_PAGES INTEGER NOT NULL WITH DEFAULT 0,
DENSITY INTEGER NOT NULL WITH DEFAULT 0,
FIRST2KEYCARD BIGINT NOT NULL WITH DEFAULT 0,
FIRST3KEYCARD BIGINT NOT NULL WITH DEFAULT 0,
FIRST4KEYCARD BIGINT NOT NULL WITH DEFAULT 0,
PCTFREE SMALLINT NOT NULL WITH DEFAULT -1,
UNIQUE_COLCOUNT SMALLINT NOT NULL WITH DEFAULT
-1,
MINPCTUSED SMALLINT NOT NULL WITH DEFAULT 0,
REVERSE_SCANS CHAR(1) NOT NULL WITH DEFAULT 'N',
USE_INDEX CHAR(1),
CREATION_TEXT CLOB(2M) NOT NULL NOT LOGGED WITH
DEFAULT "",
PACKED_DESC BLOB(1M) NOT LOGGED,
RUN_ID TIMESTAMP,
INDEXTYPE VARCHAR(4) NOT NULL WITH DEFAULT "",
EXISTS CHAR(1) NOT NULL WITH DEFAULT 'N',
RIDTOBLOCK CHAR(1) NOT NULL WITH DEFAULT 'N',
FOREIGN KEY (RUN_ID)
REFERENCES ADVISE_INSTANCE (START_TIME)
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- ADVISE_WORKLOAD
--
CREATE TABLE ADVISE_WORKLOAD (
WORKLOAD_NAME CHAR(128) NOT NULL WITH DEFAULT
'WK0',
STATEMENT_NO INTEGER NOT NULL WITH DEFAULT 1,
STATEMENT_TEXT CLOB(2M) NOT NULL NOT LOGGED,
STATEMENT_TAG VARCHAR(256) NOT NULL WITH DEFAULT ""
),
FREQUENCY INTEGER NOT NULL WITH DEFAULT 1,
IMPORTANCE DOUBLE NOT NULL WITH DEFAULT 1,
WEIGHT DOUBLE NOT NULL WITH DEFAULT 1,
COST_BEFORE DOUBLE,
COST_AFTER DOUBLE,
COMPILABLE CHAR(17))
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- ADVISE_MQT
--
CREATE TABLE ADVISE_MQT (
EXPLAIN_REQUESTER VARCHAR(128) NOT NULL WITH
DEFAULT "",
EXPLAIN_TIME TIMESTAMP NOT NULL WITH DEFAULT
CURRENT_TIMESTAMP,
SOURCE_NAME VARCHAR(128) NOT NULL WITH DEFAULT "",
SOURCE_SCHEMA VARCHAR(128) NOT NULL WITH DEFAULT
",

```

```

SOURCE_VERSION VARCHAR(64) NOT NULL WITH DEFAULT
",
EXPLAIN_LEVEL CHAR(1) NOT NULL WITH DEFAULT ",
STMTNO INTEGER NOT NULL WITH DEFAULT 0,
SECTNO INTEGER NOT NULL WITH DEFAULT 0,
NAME VARCHAR(128) NOT NULL,
CREATOR VARCHAR(128) NOT NULL WITH DEFAULT ",
IID SMALLINT NOT NULL WITH DEFAULT 0,
CREATE_TIME TIMESTAMP NOT NULL WITH DEFAULT
CURRENT_TIMESTAMP,
STATS_TIME TIMESTAMP WITH DEFAULT CURRENT
TIMESTAMP,
NUMROWS DOUBLE NOT NULL WITH DEFAULT 0,
NUMCOLS SMALLINT NOT NULL WITH DEFAULT 0,
ROWSIZE DOUBLE NOT NULL WITH DEFAULT 0,
BENEFIT FLOAT NOT NULL WITH DEFAULT 0.0,
USE_MQT CHAR(1),
MQT_SOURCE CHAR(1),
QUERY_TEXT CLOB(2M) NOT NULL NOT LOGGED WITH
DEFAULT ",
CREATION_TEXT CLOB(2M) NOT NULL NOT LOGGED WITH
DEFAULT ",
SAMPLE_TEXT CLOB(2M) NOT NULL NOT LOGGED WITH
DEFAULT ",
COLSTATS CLOB(2M) NOT NULL NOT LOGGED WITH
DEFAULT ",
EXTRA_INFO BLOB(2M) NOT NULL NOT LOGGED with
default BLOB(""),
TBSPACE VARCHAR(128) NOT NULL WITH DEFAULT ",
RUN_ID TIMESTAMP,
REFRESH_TYPE CHAR(1) NOT NULL WITH DEFAULT ",
EXISTS CHAR(1) NOT NULL WITH DEFAULT 'N',
FOREIGN KEY (RUN_ID)
REFERENCES ADVISE_INSTANCE (START_TIME)
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- ADVISE_PARTITION
--
CREATE TABLE ADVISE_PARTITION (
EXPLAIN_REQUESTER VARCHAR(128) NOT NULL WITH
DEFAULT ",
EXPLAIN_TIME TIMESTAMP NOT NULL WITH DEFAULT
CURRENT_TIMESTAMP,
SOURCE_NAME VARCHAR(128) NOT NULL WITH DEFAULT ",
SOURCE_SCHEMA VARCHAR(128) NOT NULL WITH DEFAULT
",
SOURCE_VERSION VARCHAR(64) NOT NULL WITH DEFAULT
",
EXPLAIN_LEVEL CHAR(1) NOT NULL WITH DEFAULT ",
STMTNO INTEGER NOT NULL WITH DEFAULT 0,
SECTNO INTEGER NOT NULL WITH DEFAULT 0,
QUERYNO INTEGER NOT NULL WITH DEFAULT 0,
QUERYTAG CHAR(20) NOT NULL WITH DEFAULT ",
TBNAME VARCHAR(128) NOT NULL,
TB_CREATOR VARCHAR(128) NOT NULL WITH DEFAULT ",
PMID SMALLINT NOT NULL,
TBSPACE VARCHAR(128) NOT NULL WITH DEFAULT ",
COLNAMES CLOB(2M) NOT NULL NOT LOGGED WITH
DEFAULT ",
COLCOUNT SMALLINT NOT NULL WITH DEFAULT 0,
REPLICATE CHAR(1) NOT NULL WITH DEFAULT 'N',
COST DOUBLE NOT NULL,
USEIT CHAR(1),
RUN_ID TIMESTAMP,
FOREIGN KEY (RUN_ID)
REFERENCES ADVISE_INSTANCE (START_TIME)
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- ADVISE_TABLE
--
CREATE TABLE ADVISE_TABLE (
RUN_ID TIMESTAMP,
TABLE_NAME VARCHAR(128) NOT NULL,
TABLE_SCHEMA VARCHAR(128) NOT NULL WITH DEFAULT
",
TABLESPACE VARCHAR(128) NOT NULL WITH DEFAULT ",
SELECTION_FLAG VARCHAR(8) NOT NULL WITH DEFAULT ",
TABLE_EXISTS CHAR(1) NOT NULL WITH DEFAULT ",
USE_TABLE CHAR(1) NOT NULL WITH DEFAULT ",
GEN_COLUMNS CLOB(2M) NOT NULL NOT LOGGED
WITH DEFAULT ",
ORGANIZE_BY CLOB(2M) NOT NULL NOT LOGGED
WITH DEFAULT ",
CREATION_TEXT CLOB(2M) NOT NULL NOT LOGGED WITH
DEFAULT ",
ALTER_COMMAND CLOB(2M) NOT NULL NOT LOGGED
WITH DEFAULT ",
DISKUSE DOUBLE NOT NULL WITH DEFAULT 0,
FOREIGN KEY (RUN_ID)
REFERENCES ADVISE_INSTANCE (START_TIME)
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- Commit work
--
COMMIT WORK;
--
-- Optional Indexes: The following indexes are recommended for improved
performance
-- of explain-related utilities. These create index statements can be deleted, or
-- the indexes dropped if space is a problem.
--
CREATE INDEX STMT_I1 on
EXPLAIN_STATEMENT(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO);
CREATE INDEX ARG_I1 on
EXPLAIN_ARGUMENT(EXPLAIN_TIME, EXPLAIN_LEVEL, STMTNO,
SECTNO, OPERATOR_ID);
CREATE INDEX PRD_I1 on
EXPLAIN_PREDICATE(EXPLAIN_TIME, EXPLAIN_LEVEL, STMTNO,
SECTNO, OPERATOR_ID);
CREATE INDEX OPR_I1 on
EXPLAIN_OPERATOR(EXPLAIN_TIME, EXPLAIN_LEVEL, STMTNO,
SECTNO, OPERATOR_ID);
CREATE INDEX STM_I1 on
EXPLAIN_STREAM(EXPLAIN_TIME, EXPLAIN_LEVEL, STMTNO,
SECTNO);
CREATE INDEX OBJ_I1 on
EXPLAIN_OBJECT(EXPLAIN_TIME, EXPLAIN_LEVEL, STMTNO,
SECTNO);
CREATE INDEX IDX_I1 on
ADVISE_INDEX (EXPLAIN_TIME);
CREATE INDEX IDX_I2 on
ADVISE_INDEX (NAME, EXPLAIN_TIME);
CREATE INDEX MQT_I1 on
ADVISE_MQT (EXPLAIN_TIME);
CREATE INDEX MQT_I2 on
ADVISE_MQT (NAME, EXPLAIN_TIME);
CREATE INDEX PRT_I1 on
ADVISE_PARTITION (EXPLAIN_TIME);

```

```

--
-- Commit work
--
COMMIT WORK;
utils/UNEXPLAIN.ddl
-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

```

```

DROP INDEX STMT_I1;
DROP INDEX ARG_I1;
DROP INDEX PRD_I1;
DROP INDEX OPR_I1;
DROP INDEX STM_I1;
DROP INDEX OBJ_I1;
DROP TABLE EXPLAIN_INSTANCE;
DROP TABLE EXPLAIN_STATEMENT;
DROP TABLE EXPLAIN_ARGUMENT;
DROP TABLE EXPLAIN_OBJECT;
DROP TABLE EXPLAIN_OPERATOR;
DROP TABLE EXPLAIN_PREDICATE;
DROP TABLE EXPLAIN_STREAM;
DROP TABLE ADVISE_INDEX;
DROP TABLE ADVISE_WORKLOAD;

```

tpccCom/comreg.h

```

// compreg.h : Declaration of the CCompReg
#pragma once
#include "resource.h" // main symbols
#include "tpccCom.h"

// CCompReg
class ATL_NO_VTABLE CCompReg :
public CComObjectRootEx<CComSingleThreadModel>,
public CComCoClass<CCompReg, &CLSID_CompReg>,
public IDispatchImpl<IComponentRegistrar,
&IID_IComponentRegistrar, &LIBID_tpccComLib, /*wMajor=*/ 1, /*wMinor=*/ 0>
{
public:
    CCompReg()
    {
    }
    DECLARE_NO_REGISTRY()
    BEGIN_COM_MAP(CCompReg)
        COM_INTERFACE_ENTRY(IComponentRegistrar)
        COM_INTERFACE_ENTRY(IDispatch)
    END_COM_MAP()
    // IComponentRegistrar
public:
    STDMETHODCALLTYPE(BSTR bstrPath)
    {
        return S_OK;
    }
    STDMETHODCALLTYPE(RegisterAll())
    {
        return _AtlComModule.RegisterServer(TRUE);
    }

```

```

}
STDMETHOD(UnregisterAll())
{
    _AtlComModule.UnregisterServer(TRUE);
    return S_OK;
}
STDMETHOD(GetComponents)(SAFEARRAY **ppCLSIDs,
SAFEARRAY **ppDescriptions)
{
    if( ppCLSIDs == NULL || ppDescriptions == NULL )
        return E_POINTER;
    int nComponents = 0;
    for( _ATL_OBJMAP_ENTRY** ppEntry =
_AtlComModule.m_ppAutoObjMapFirst; ppEntry <
_AtlComModule.m_ppAutoObjMapLast; ppEntry++)
    {
        if(*ppEntry != NULL)
        {
            _ATL_OBJMAP_ENTRY* pEntry
= *ppEntry;
            if( pEntry->pclsid != NULL)
            {
                LPCTSTR
pszDescription = pEntry->pfnGetObjectDescription();
                if( pszDescription)
nComponents++;
            }
        }
    }
    SAFEARRAYBOUND rgBound[1];
    rgBound[0].lLbound = 0;
    rgBound[0].cElements = nComponents;
    *ppCLSIDs = SafeArrayCreate(VT_BSTR, 1, rgBound);
    if( *ppCLSIDs == NULL )
        return AtlHresultFromLastError();
    *ppDescriptions = SafeArrayCreate(VT_BSTR, 1,
rgBound);
    if( *ppDescriptions == NULL )
        return AtlHresultFromLastError();
    LONG i = 0;
    for( _ATL_OBJMAP_ENTRY** ppEntry =
_AtlComModule.m_ppAutoObjMapFirst; ppEntry <
_AtlComModule.m_ppAutoObjMapLast; ppEntry++)
    {
        if(*ppEntry != NULL)
        {
            _ATL_OBJMAP_ENTRY* pEntry
= *ppEntry;
            if( pEntry->pclsid != NULL)
            {
                LPCTSTR
pszDescription = pEntry->pfnGetObjectDescription();
                if( pszDescription)
                    LPOLESTR
pszCLSID;
                StringFromCLSID(*pEntry->pclsid, &pszCLSID);
                BSTR
pBSTR = OLE2BSTR(pszCLSID);
                if( pBSTR
== NULL )
            {
                CoTaskMemFree(pszCLSID);
                return E_OUTOFMEMORY;
            }

```

```

    }
    HRESULT
    HRESULT = SafeArrayPutElement(*ppCLSIDs, &i, pBSTR);

    CoTaskMemFree(pszCLSID);

    FAILED(hResult)

    return hResult;

    T2BSTR_EX(pszDescription);

    == NULL )

    return E_OUTOFMEMORY;

    SafeArrayPutElement(*ppDescriptions, &i, pBSTR);

    FAILED(hResult)

    return hResult;

    }
    }
    }
    }
    }
    }
    return S_OK;
}
STDMETHOD(RegisterComponent)(BSTR bstrCLSID)
{
    CLSID clsid;
    CLSIDFromString(bstrCLSID, &clsid);
    _AtlComModule.RegisterServer(TRUE, &clsid);
    return S_OK;
}
STDMETHOD(UnregisterComponent)(BSTR bstrCLSID)
{
    CLSID clsid;
    CLSIDFromString(bstrCLSID, &clsid);
    _AtlComModule.UnregisterServer(FALSE, &clsid);
    return S_OK;
}
};
OBJECT_ENTRY_AUTO(CLSID_CompReg, CCompReg)

```

tpccCom/dlldatax.h

```

#pragma once
#ifdef _MERGE_PROXYSTUB
extern "C"
{
    BOOL WINAPI PrxDllMain(HINSTANCE hInstance, DWORD dwReason,
        LPVOID lpReserved);
    STDAPI PrxDllCanUnloadNow(void);
    STDAPI PrxDllGetClassObject(REFCLSID rclsid, REFIID riid, LPVOID*
        ppv);
    STDAPI PrxDllRegisterServer(void);
    STDAPI PrxDllUnregisterServer(void);
}
#endif

```

tpccCom/Resource.h

```

//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.

```

```

// Used by tpccCom.rc
//
#define IDS_PROJNAME           100
#define IDR_TPCCCOM           101
#define IDR_TPCC_COM           102
// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifdef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE  201
#define _APS_NEXT_COMMAND_VALUE   32768
#define _APS_NEXT_CONTROL_VALUE   201
#define _APS_NEXT_SYMED_VALUE     103
#endif
#endif

tpccCom/stdafx.h
// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently,
// but are changed infrequently
#pragma once
#ifdef STRICT
#define STRICT
#endif
// Modify the following defines if you have to target a platform prior to the ones
// specified below.
// Refer to MSDN for the latest info on corresponding values for different
// platforms.
#ifdef WINVER // Allow use of features
// specific to Windows 95 and Windows NT 4 or later.
#define WINVER 0x0400 // Change this to the appropriate
// value to target Windows 98 and Windows 2000 or later.
#endif
#ifdef _WIN32_WINNT // Allow use of features specific to
// Windows NT 4 or later.
#define _WIN32_WINNT 0x0400 // Change this to the appropriate
// value to target Windows 2000 or later.
#endif
#ifdef _WIN32_WINDOWS // Allow use of features specific to
// Windows 98 or later.
#define _WIN32_WINDOWS 0x0410 // Change this to the appropriate value to
// target Windows Me or later.
#endif
#ifdef _WIN32_IE // Allow use of features specific to
// IE 4.0 or later.
#define _WIN32_IE 0x0400 // Change this to the appropriate value to
// target IE 5.0 or later.
#endif

```

```

#define _ATL_APARTMENT_THREADED
#define _ATL_NO_AUTOMATIC_NAMESPACE
#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS // some
// CString constructors will be explicit
// turns off ATL's hiding of some common and often safely ignored warning
// messages
#define _ATL_ALL_WARNINGS

```

```

#include <comsvcs.h>
#include "resource.h"
#include <atlbase.h>
#include <atlcom.h>
using namespace ATL;

```

```

#include <comsvcs.h>
#include "resource.h"
#include <atlbase.h>
#include <atlcom.h>
using namespace ATL;

```

tpccCom/tpccCom.h

```

// tpcc_com.h : Declaration of the Ctpcc_com
#pragma once
#include "tpccCom.h"

```

```

#include "resource.h"// main symbols
#include <comsvcs.h>
#include "..\tpcc\api\tpcc.h"
#include <db2tpcc.h>
#include <tpcc.h>
#define NULL_DB "nullDB"
static HINSTANCE dbInstance = NULL;
static CRITICAL_SECTION debugMutex;
static CRITICAL_SECTION errorMutex;
static int comServerID = 0;
static ofstream debugStream;
static ofstream errorStream;
static int debugFileOpen = 0;
static int errorFileOpen = 0;
static int nullDB = 0;
static char dbType[32];
static char dbName[32];
typedef INT (*NORD_PTR)(nord_wrapper *nord,void *connectHandle);
typedef INT (*PYMT_PTR)(paym_wrapper *pymt,void *connectHandle);
typedef INT (*ORDS_PTR)(ords_wrapper *ords,void *connectHandle);
typedef INT (*STOK_PTR)(stok_wrapper *stok,void *connectHandle);
typedef INT (*CONNECT_PTR)(char *dbName,void **connectHandle);
typedef INT (*DISCONNECT_PTR)(void *connectHandle);
NORD_PTR do_nord;
PYMT_PTR do_pymt;
ORDS_PTR do_ords;
STOK_PTR do_stok;
CONNECT_PTR do_connection;
DISCONNECT_PTR do_disconnect;

// Ctpcc_com
class ATL_NO_VTABLE Ctpcc_com :
public CComObjectRootEx<CComMultiThreadModel>,
public IObjectControl,
public CComCoClass<Ctpcc_com, &CLSID_tpcc_com>,
public Itpcc_com
{
public:
Ctpcc_com()
{
int rc = ERR;
connected = 0;
connectHandleInUse = 0;
if(debugFlag)
{
if(!debugFileOpen)
{
InitializeCriticalSection(&debugMutex);
//open comLog
char comLogFile[128];

sprintf(comLogFile,"C:\\inetpub\\wwwroot\\tpcc\\comLog_debug.txt");
debugStream.rdbuf(
)->open(comLogFile,ios_base::in | ios_base::out | ios_base::app);
debugFileOpen = 1;
}
}

//open error log file
if(!errorFileOpen)
{
InitializeCriticalSection(&errorMutex);
char errorLogFile[128];

sprintf(errorLogFile,"C:\\inetpub\\wwwroot\\tpcc\\comLog_err.txt");

```

```

errorStream.rdbuf(
)->open(errorLogFile,ios_base::in | ios_base::out | ios_base::app);
errorFileOpen=1;
}
//get registry values
if(rc = readRegistry()) != OK)
{
ERRORMSG("Unable to open registry key "
<< REGISTRY_SUB_KEY << " rc:" << rc <<endl);
return;
}

DEBUGMSG("nullDB:" <<nullDB<<"
dbType:"<<dbType<<" dbName:"<<dbName<<endl);

//load library based on registry
if( rc = loadLibrary()) != OK)
{
ERRORMSG("load library failure rc:" << rc
<< endl);
return;
}

DEBUGMSG("dbtype:"<<dbType<<" instance:" <<
DEBUGADDRESS(dbInstance) << " loaded." << endl);

//connect to db
EnterCriticalSection(&errorMutex);
if(rc = connectDB()) != OK)
{
ERRORMSG("unable to connect to db
"<<dbName<<" rc :"<<rc <<endl);
LeaveCriticalSection(&errorMutex);
return;
}
LeaveCriticalSection(&errorMutex);

DEBUGMSG("connected to db " <<dbName<< " rc:"<<
rc << " context:" <<DEBUGADDRESS(connectHandle) << endl);
}
DECLARE_PROTECT_FINAL_CONSTRUCT()
HRESULT FinalConstruct()
{
return S_OK;
}

void FinalRelease()
{
}
}
DECLARE_REGISTRY_RESOURCEID(IDR_TPCC_COM)
BEGIN_COM_MAP(Ctpcc_com)
COM_INTERFACE_ENTRY(Itpcc_com)
COM_INTERFACE_ENTRY(IObjectControl)
END_COM_MAP()
// IObjectControl
public:
STDMETHOD(Activate)();
STDMETHOD_(BOOL, CanBePooled)();
STDMETHOD_(void, Deactivate)();
CComPtr<IObjectContext> m_spObjectContext;

// Itpcc_com
public:
STDMETHOD(doStockLevel)(INT *size, UCHAR **buffer);
STDMETHOD(doNewOrder)(INT* size, UCHAR** buffer);
STDMETHOD(doPayment)(INT* size, UCHAR** buffer);
STDMETHOD(doOrderStatus)(INT* size, UCHAR** buffer);
STDMETHOD(doDBInfo)(void);

```

```

        STDMETHODCALLTYPE(doSetComplete)(void);
        int connected;
        int connectHandleInUse;
private:
        //db2 specific context
        void *connectHandle;
        int loadLibrary();
        int readRegistry();
        int connectDB();

};
OBJECT_ENTRY_AUTO(__uuidof(tpcc_com), Ctpcc_com)
tpccCom/tpcc_com.h

// tpcc_com.h : Declaration of the Ctpcc_com
#pragma once
#include "tpccCom.h"
#include "resource.h"// main symbols
#include <comsvcs.h>
#include "..\tpccapi\tpcc.h"
#include <db2tpcc.h>
#include <tpcc.h>
#define NULL_DB "nullDB"
static HINSTANCE dbInstance = NULL;
static CRITICAL_SECTION debugMutex;
static CRITICAL_SECTION errorMutex;
static int comServerID = 0;
static ofstream debugStream;
static ofstream errorStream;
static int debugFileOpen = 0;
static int errorFileOpen = 0;
static int nullDB = 0;
static char dbType[32];
static char dbName[32];
typedef INT (*NORD_PTR)(nord_wrapper *nord,void *connectHandle);
typedef INT (*PYMT_PTR)(paym_wrapper *pymt,void *connectHandle);
typedef INT (*ORDS_PTR)(ords_wrapper *ords,void *connectHandle);
typedef INT (*STOK_PTR)(stok_wrapper *stok,void *connectHandle);
typedef INT (*CONNECT_PTR)(char *dbName,void **connectHandle);
typedef INT (*DISCONNECT_PTR)(void *connectHandle);
NORD_PTR do_nord;
PYMT_PTR do_pymt;
ORDS_PTR do_ords;
STOK_PTR do_stok;
CONNECT_PTR do_connection;
DISCONNECT_PTR do_disconnect;

// Ctpcc_com
class ATL_NO_VTABLE Ctpcc_com :
public CComObjectRootEx<CComMultiThreadModel>,
public IObjectControl,
public CComCoClass<Ctpcc_com, &CLSID_tpcc_com>,
public Itpc_com
{
public:
        Ctpcc_com()
        {
                int rc = ERR;
                connected = 0;
                connectHandleInUse = 0;
                if(debugFlag)
                {
                        if(!debugFileOpen)
                }
        }
        InitializeCriticalSection(&debugMutex);
        //open comLog

```

```

        char comLogFile[128];
        sprintf(comLogFile,"C:\\inetpub\\wwwroot\\tpcc\\comLog_debug.txt");
        debugStream.rdbuf(
)->open(comLogFile,ios_base::in | ios_base::out | ios_base::app);
        debugFileOpen = 1;
        }
        //open error log file
        if(!errorFileOpen)
        {
                InitializeCriticalSection(&errorMutex);
                char errorLogFile[128];
        }
        sprintf(errorLogFile,"C:\\inetpub\\wwwroot\\tpcc\\comLog_err.txt");
        errorStream.rdbuf(
)->open(errorLogFile,ios_base::in | ios_base::out | ios_base::app);
        errorFileOpen=1;
        }
        //get registry values
        if(rc = readRegistry() != OK)
        {
                ERRORMSG("Unable to open registry key "
<< REGISTRY_SUB_KEY << " rc:" << rc <<endl);
                return;
        }
        DEBUGMSG("nullDB:" <<nullDB<<"
dbType:"<<dbType<<" dbName:"<<dbName<<endl);
        //load library based on registry
        if( rc = loadLibrary() != OK)
        {
                ERRORMSG("load library failure rc:" << rc
<< endl);
                return;
        }
        DEBUGMSG("dbtype:"<<dbType<<" instance:" <<
DEBUGADDRESS(dbInstance) << " loaded." << endl);
        //connect to db
        EnterCriticalSection(&errorMutex);
        if(rc = connectDB() != OK)
        {
                ERRORMSG("unable to connect to db
"<<dbName<<" rc :"<<rc <<endl);
                LeaveCriticalSection(&errorMutex);
                return;
        }
        LeaveCriticalSection(&errorMutex);
        DEBUGMSG("connected to db " <<dbName<< " rc:"<<
rc << " context:" <<DEBUGADDRESS(connectHandle) << endl);
        }
        DECLARE_PROTECT_FINAL_CONSTRUCT()
        HRESULT FinalConstruct()
        {
                return S_OK;
        }
        void FinalRelease()
        {
        }
        DECLARE_REGISTRY_RESOURCEID(IDR_TPCC_COM)
        BEGIN_COM_MAP(Ctpcc_com)
                COM_INTERFACE_ENTRY(Itpc_com)

```

```

        COM_INTERFACE_ENTRY(IObjectControl)
END_COM_MAP()
// IObjectControl
public:
    STDMETHOD(Activate)();
    STDMETHOD_(BOOL, CanBePooled)();
    STDMETHOD_(void, Deactivate)();
    CComPtr<IObjectContext> m_spObjectContext;

// Itpcc_com
public:
    STDMETHOD(doStockLevel)(INT *size, UCHAR **buffer);
    STDMETHOD(doNewOrder)(INT* size, UCHAR** buffer);
    STDMETHOD(doPayment)(INT* size, UCHAR** buffer);
    STDMETHOD(doOrderStatus)(INT* size, UCHAR** buffer);
    STDMETHOD(doDBInfo)(void);
    STDMETHOD(doSetComplete)(void);
    int connected;
    int connectHandleInUse;

private:
    //db2 specific context
    void *connectHandle;
    int loadLibrary();
    int readRegistry();
    int connectDB();

};
OBJECT_ENTRY_AUTO(__uuidof(tpcc_com), Ctpcc_com)

```

tpccCom/tpccCom.def

; tpccCom.def : Declares the module parameters.

```

LIBRARY "tpccCom.DLL"
EXPORTS
    DllCanUnloadNow PRIVATE
    DllGetClassObject PRIVATE
    DllRegisterServer PRIVATE
    DllUnregisterServer PRIVATE

```

tpccCom/tpccCom.idl

```

// tpccCom.idl : IDL source for tpccCom
//
// This file will be processed by the MIDL tool to
// produce the type library (tpccCom.tlb) and marshalling code.
import "oaidl.idl";
import "ocidl.idl";
//this is test.
[
    object,
    uuid(a817e7a2-43fa-11d0-9e44-00aa00b6770a),
    dual,
    helpstring("IComponentRegistrar Interface"),
    pointer_default(unique)
]
interface IComponentRegistrar : IDispatch
{
    [id(1)] HRESULT Attach([in] BSTR bstrPath);
    [id(2)] HRESULT RegisterAll();
    [id(3)] HRESULT UnregisterAll();
    [id(4)] HRESULT GetComponents([out]
SAFEARRAY(BSTR)* pbsrCLSIDs, [out] SAFEARRAY(BSTR)*
pbsrDescriptions);
    [id(5)] HRESULT RegisterComponent([in] BSTR bstrCLSID);
    [id(6)] HRESULT UnregisterComponent([in] BSTR bstrCLSID);
};
[
    object,

```

```

    uuid(5B4FA473-2E68-4D79-A626-F38B30B8196E),
    helpstring("Itpcc_com Interface"),
    pointer_default(unique)
]
interface Itpcc_com : IUnknown{
    [helpstring("method doStockLevel")] HRESULT doStockLevel([in]
INT *size, [in,out, size_is(*size)] UCHAR **buffer);
    [helpstring("method doNewOrder")] HRESULT doNewOrder([in]
INT* size, [in,out,size_is(*size)] UCHAR** buffer);
    [helpstring("method doPayment")] HRESULT doPayment([in] INT*
size, [in,out,size_is(*size)] UCHAR** buffer);
    [helpstring("method doOrderStatus")] HRESULT
doOrderStatus([in] INT* size, [in,out,size_is(*size)] UCHAR** buffer);
    [helpstring("method doDBInfo")] HRESULT doDBInfo(void);
    [helpstring("method doSetComplete")] HRESULT
doSetComplete(void);
};
[
    uuid(91F1B8B0-89E9-457B-A228-3E2D6CE3E752),
    version(1.0),
    helpstring("tpccCom 1.0 Type Library"),

custom(a817e7a1-43fa-11d0-9e44-00aa00b6770a,"{90EEDAFF-F8D3-4711-99
A9-8AC3C0FE5DB9}")
]
library tpccComLib
{
    importlib("stdole2.tlb");
    [
        uuid(90EEDAFF-F8D3-4711-99A9-8AC3C0FE5DB9),
        helpstring("ComponentRegistrar Class")
    ]
    coclass CompReg
    {
        [default] interface IComponentRegistrar;
    };
    [
        uuid(5F752BF2-F739-43D4-8492-44C19581C0A1),
        helpstring("tpcc_com Class")
    ]
    coclass tpcc_com
    {
        [default] interface Itpcc_com;
    };
};

```

tpccCom/tpcc_com.rgs

```

HKCR
{
    tpccCom.tpcc_com.1 = s 'tpcc_com Class'
    {
        CLSID = s
        '{5F752BF2-F739-43D4-8492-44C19581C0A1}'
    }
    tpccCom.tpcc_com = s 'tpcc_com Class'
    {
        CLSID = s
        '{5F752BF2-F739-43D4-8492-44C19581C0A1}'
        CurVer = s 'tpccCom.tpcc_com.1'
    }
    NoRemove CLSID
    {
        ForceRemove
        {5F752BF2-F739-43D4-8492-44C19581C0A1} = s 'tpcc_com Class'
        {
            ProgID = s 'tpccCom.tpcc_com.1'
            VersionIndependentProgID = s
            'tpccCom.tpcc_com'
        }
    }
}

```

```

        InprocServer32 = s '%MODULE%'
        {
            val ThreadingModel = s 'Both'
        }
        val AppID = s '%APPID%'
        'TypeLib' = s
        '{91F1B8B0-89E9-457B-A228-3E2D6CE3E752}'
    }
}

```

tpccCom/comreg.cpp

```

// compreg.cpp : Implementation of CCompReg
#include "stdafx.h"
#include "comreg.h"

```

```
// CCompReg
```

tpccCom/stdafx.cpp

```

// stdafx.cpp : source file that includes just the standard includes
// tpccCom.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information
#include "stdafx.h"

```

tpccCom/tpccCom.cpp

```

// tpccCom.cpp : Implementation of DLL Exports.
//
// Note: COM+ 1.0 Information:
// Please remember to run Microsoft Transaction Explorer to install the
// component(s).
// Registration is not done by default.
#include "stdafx.h"
#include "resource.h"
#include "tpccCom.h"
#include "comreg.h"
#include "dlldatax.h"
class CtpccComModule : public CAtlDllModuleT< CtpccComModule >
{
public :
    DECLARE_LIBID(LIBID_tpccComLib)

```

```

    DECLARE_REGISTRY_APPID_RESOURCEID(IDR_TPCCCOM,
    "{11ED2355-1A27-42F1-ADFF-F201F5E82BCE}");
};
CtpccComModule _AtlModule;

```

```

// DLL Entry Point
extern "C" BOOL WINAPI DllMain(HINSTANCE hInstance, DWORD
dwReason, LPVOID lpReserved)
{
#ifdef _MERGE_PROXYSTUB
    if (!PrxDllMain(hInstance, dwReason, lpReserved))
        return FALSE;
#endif
    hInstance;
    return _AtlModule.DllMain(dwReason, lpReserved);
}

```

```

// Used to determine whether the DLL can be unloaded by OLE
STDAPI DllCanUnloadNow(void)
{
#ifdef _MERGE_PROXYSTUB
    HRESULT hr = PrxDllCanUnloadNow();
    if (FAILED(hr))
        return hr;
#endif
}

```

```

return _AtlModule.DllCanUnloadNow();
}

// Returns a class factory to create an object of the requested type
STDAPI DllGetClassObject(REFCLSID rclsid, REFIID riid, LPVOID* ppv)
{
#ifdef _MERGE_PROXYSTUB
    if (PrxDllGetClassObject(rclsid, riid, ppv) == S_OK)
        return S_OK;
#endif
    return _AtlModule.DllGetClassObject(rclsid, riid, ppv);
}

// DllRegisterServer - Adds entries to the system registry
STDAPI DllRegisterServer(void)
{
    // registers object, typelib and all interfaces in typelib
    HRESULT hr = _AtlModule.DllRegisterServer();
#ifdef _MERGE_PROXYSTUB
    if (FAILED(hr))
        return hr;
    hr = PrxDllRegisterServer();
#endif
    return hr;
}

```

```

// DllUnregisterServer - Removes entries from the system registry
STDAPI DllUnregisterServer(void)
{
    HRESULT hr = _AtlModule.DllUnregisterServer();
#ifdef _MERGE_PROXYSTUB
    if (FAILED(hr))
        return hr;
    hr = PrxDllRegisterServer();
    if (FAILED(hr))
        return hr;
    hr = PrxDllUnregisterServer();
#endif
    return hr;
}

```

tpccCom/tpcc_com.cpp

```

// tpcc_com.cpp : Implementation of Ctpcc_com
#include "stdafx.h"
#include "tpcc_com.h"
#include ".\tpcc_com.h"
#include <db2tpcc.h>
// Ctpcc_com
HRESULT Ctpcc_com::Activate()
{
    HRESULT hr = GetObjectContext(&m_spObjectContext);
    if (SUCCEEDED(hr))
    {
        DEBUGMSG("Object assigned to thread."<<endl);
        return S_OK;
    }
    return hr;
}
BOOL Ctpcc_com::CanBePooled()
{
    DEBUGMSG("CanBePooled() returning true"<<endl);
    return TRUE;
}
void Ctpcc_com::Deactivate()
{
}

```



```

        DEBUGMSG("deactivated() releasing object back into
pool"<<endl);
        m_spObjectContext.Release();
    }
    /*
    *****
    ** Name          :          doSetComplete
    ** Description    :
    **               Release object back
    into com pool
    ** Parameters    :
    ** Returns       :
    **               int - return code
    ** Comments      :
    **               Calls SetComplete on
    the object that the com
    **               pool manager returned
    to the caller(isapi thread)
    *****
    */
    STDMETHODCALLTYPE Ctpcc_com::doSetComplete(void)
    {
        // TODO: Add your implementation code here
        HRESULT hres = m_spObjectContext->SetComplete();
        if (SUCCEEDED(hres))
        {
            DEBUGMSG("SetComplete successful. object bit set to
release object into pool."<<endl);
        }
        else
        {
            DEBUGMSG("SetComplete failed. object bit set to
release object into pool."<<endl);
            ERRORMSG("SetComplete() failed,
code:"<<HRESULT_CODE(hres)<<"
facility:"<<HRESULT_FACILITY(hres)<<" hres:"<<hex<<hres<<endl);
        }
        return S_OK;
    }
    /*
    *****
    ** Name          :          doStockLevel
    ** Description    :
    **               Call db2 dll entry point
    to execute txn
    ** Parameters    :
    **               int*
    size of UCHAR buffer to pay attention to
    **               UCHAR**
    char buffer that holds txn wrapper struct
    ** Returns       :
    **               int - return code
    ** Comments      :
    *****
    */
    STDMETHODCALLTYPE Ctpcc_com::doStockLevel(INT *size, UCHAR **buffer)
    {
        stok_wrapper * stok;

        stok = (stok_wrapper *) *buffer;

        if(!connectHandleInUse)
        {
            DEBUGMSG("Setting Context handle in use to
true"<<endl);
            connectHandleInUse = 1;
        }
    }

```

```

    else
    {
        DEBUGMSG("Context handle in use."<<endl);
        ERRORMSG("Context handle in use."<<endl);
        return ERR_HANDLE_IN_USE;
    }
    DEBUGMSG("Calling do_stok call using
connectHandle:"<<DEBUGADDRESS(connectHandle)<<"
w_id:"<<stok->in_stok.s_W_ID<<" d_id:"<<stok->in_stok.s_D_ID<<"
"
s_transtatus:"<<stok->out_stok.s_transtatus<<endl);

    do_stok(stok,connectHandle);
    DEBUGMSG("Return from do_stok call using
connectHandle:"<<DEBUGADDRESS(connectHandle)<<"
w_id:"<<stok->in_stok.s_W_ID<<" d_id:"<<stok->in_stok.s_D_ID<<"
"
s_transtatus:"<<stok->out_stok.s_transtatus<<endl);
    DEBUGMSG("Connection handle set to free" <<endl);
    connectHandleInUse = 0;
    return S_OK;
}
/*
*****
** Name          :          doNewOrder
** Description    :
**               Call db2 dll entry point
to execute txn
** Parameters    :
**               int*
size of UCHAR buffer to pay attention to
**               UCHAR**
char buffer that holds txn wrapper struct
** Returns       :
**               int - return code
** Comments      :
*****
*/
 STDMETHODCALLTYPE Ctpcc_com::doNewOrder(INT* size, UCHAR** buffer)
    {
        nord_wrapper *nord;
        nord = (nord_wrapper *) *buffer;
        if(!connectHandleInUse)
        {
            DEBUGMSG("Setting Context handle in use to
true"<<endl);
            connectHandleInUse = 1;
        }
        else
        {
            DEBUGMSG("Context handle in use."<<endl);
            ERRORMSG("Context handle in use."<<endl);
            return ERR_HANDLE_IN_USE;
        }
    }
    DEBUGMSG("Calling do_nord call using
connectHandle:"<<DEBUGADDRESS(connectHandle)<<"
w_id:"<<nord->in_nord.s_W_ID<<" d_id:"<<nord->in_nord.s_D_ID<<"
"
s_transtatus:"<<nord->out_nord.s_transtatus<<endl);
    do_nord(nord,connectHandle);

    DEBUGMSG("Return from do_nord call using
connectHandle:"<<DEBUGADDRESS(connectHandle)<<"
w_id:"<<nord->in_nord.s_W_ID<<" d_id:"<<nord->in_nord.s_D_ID<<"
"
s_transtatus:"<<nord->out_nord.s_transtatus<<endl);

```

```

        DEBUGMSG("Connection handle set to free" <<endl);
        connectHandleInUse = 0;
        return S_OK;
    }
}
/*
*****
** Name          :          doPayment
** Description    :
**               Call db2 dll entry point
to execute txn
** Parameters    :
**               int*
size of UCHAR buffer to pay attention to
**               UCHAR**
char buffer that holds txn wrapper struct
** Returns      :
**               int - return code
** Comments     :
**
*****
*/
STDMETHODIMP Ctpcc_com::doPayment(INT* size, UCHAR** buffer)
{
    paym_wrapper    *pymt;
    pymt = (paym_wrapper *) *buffer;
    if(!connectHandleInUse)
    {
        DEBUGMSG("Setting Context handle in use to
true" <<endl);
        connectHandleInUse = 1;
    }
    else
    {
        DEBUGMSG("Context handle in use." <<endl);
        ERRORMSG("Context handle in use." <<endl);
        return ERR_HANDLE_IN_USE;
    }

    DEBUGMSG("Calling do_pymt call using
connectHandle:" <<DEBUGADDRESS(connectHandle) <<"
w_id:" <<pymt->in_paym.s_W_ID <<" d_id:" <<pymt->in_paym.s_D_ID <<"
"
s_transtatus:" <<pymt->out_paym.s_transtatus <<endl);
    do_pymt(pymt,connectHandle);
    DEBUGMSG("Return from do_pymt call using
connectHandle:" <<DEBUGADDRESS(connectHandle) <<"
w_id:" <<pymt->in_paym.s_W_ID <<" d_id:" <<pymt->in_paym.s_D_ID <<"
"
s_transtatus:" <<pymt->out_paym.s_transtatus <<endl);
    DEBUGMSG("Connection handle set to free" <<endl);
    connectHandleInUse = 0;

    return S_OK;
}
}
/*
*****
** Name          :          doOrderStatus
** Description    :
**               Call db2 dll entry point
to execute txn
** Parameters    :
**               int*
size of UCHAR buffer to pay attention to
**               UCHAR**
char buffer that holds txn wrapper struct
** Returns      :
**               int - return code
**
*****

```

```

** Comments     :
**
*****
*/
STDMETHODIMP Ctpcc_com::doOrderStatus(INT* size, UCHAR** buffer)
{
    ords_wrapper    *ords;
    ords = (ords_wrapper *) *buffer;
    if(!connectHandleInUse)
    {
        DEBUGMSG("Setting Context handle in use to
true" <<endl);
        connectHandleInUse = 1;
    }
    else
    {
        DEBUGMSG("Context handle in use." <<endl);
        ERRORMSG("Context handle in use." <<endl);
        return ERR_HANDLE_IN_USE;
    }

    DEBUGMSG("Calling do_ords call using
connectHandle:" <<DEBUGADDRESS(connectHandle) <<"
w_id:" <<ords->in_ords.s_W_ID <<" d_id:" <<ords->in_ords.s_D_ID <<"
"
s_transtatus:" <<ords->out_ords.s_transtatus <<endl);
    do_ords(ords,connectHandle);
    DEBUGMSG("Return from do_ords call using
connectHandle:" <<DEBUGADDRESS(connectHandle) <<"
w_id:" <<ords->in_ords.s_W_ID <<" d_id:" <<ords->in_ords.s_D_ID <<"
"
s_transtatus:" <<ords->out_ords.s_transtatus <<endl);
    DEBUGMSG("Connection handle set to free" <<endl);
    connectHandleInUse = 0;

    return S_OK;
}
}
/*
*****
** Name          :          doDBInfo
** Description    :
**               Function to test com
interface
** Parameters    :
** Returns      :
**               int - return code
** Comments     :
**
*****
*/
STDMETHODIMP Ctpcc_com::doDBInfo(void)
{
    DEBUGMSG("Stub function to warm object pool" <<endl);
    return S_OK;
}
}
/*
*****
** Name          :          loadLibrary
** Description    :
**               Function loads
appropriate db library based on
**               registry setting
** Parameters    :
** Returns      :
**               int - return code
** Comments     :
**
*****

```

```

*/
Ctpcc_com::loadLibrary()
{
    DEBUGMSG("Entered loadLibrary function"<<endl);
    //check to see if dbInstance is already loaded
    if(!dbInstance)
    {
        DEBUGMSG("Database dll not loaded. Loading
dll."<<endl);
        if (nullDB)
        {
            DEBUGMSG("Loading "<<dbType << "
nulldb dll." << endl);
            dbInstance =
LoadLibrary("c:\inetpub\wwwroot\tpcc\nullDB.dll");
            if(dbInstance == NULL)
            {
                DEBUGMSG("Unable to load null
db dll, rc:"<<GetLastError());
                ERRORMSG("Unable to load null
db dll, rc:"<<GetLastError());
                return
ERR_NULL_DLL_NOT_LOADED;
            }
            DEBUGMSG(dbType << " nulldb dll
loaded"<<endl);
        }
        else if(strcmp(dbType,"DB2") == 0)
        {
            DEBUGMSG("Loading "<<dbType << " dll."
<< endl);
            dbInstance =
LoadLibrary("c:\inetpub\wwwroot\tpcc\tpccDB2glue.dll");
            if(dbInstance == NULL)
            {
                DEBUGMSG("Unable to load
library."<<endl);
                ERRORMSG("Unable to load com
dll, rc:" << GetLastError() << endl);
                return
ERR_DB2_DLL_NOT_LOADED;
            }
            DEBUGMSG(dbType<< " dll
loaded"<<endl);
        }
        else if( strcmp(dbType,"ORACLE") == 0 )
        {
            DEBUGMSG("Unable to load oracle
dll"<<endl);
            ERRORMSG("Unable to load oracle dll,
rc:"<<GetLastError()<<endl);
            return
ERR_ORACLE_DLL_NOT_LOADED;
        }
        else
        {
            DEBUGMSG("Unknown database type
dll:"<<dbType<<endl);
            ERRORMSG("Unknown database type
dll:"<<dbType<<endl);
            return ERR_UNKNOWN_DB;
        }
        //retrieve function addresses from instance loaded.
        DEBUGMSG("Getting do_connection function address
from "<<dbType<<" dll"<<endl);
        if (do_connection =
(CONNECT_PTR)GetProcAddress(dbInstance,"connect_db") == NULL )
            return
ERR_CONNECT_ADDRESS_NOT_FOUND;
        DEBUGMSG("do_connection
address:"<<DEBUGADDRESS(do_connection)<<endl);
        DEBUGMSG("Getting do_disconnect function address
from "<<dbType<<" dll"<<endl);
        if (do_disconnect =
(DISCONNECT_PTR)GetProcAddress(dbInstance,"disconnect_db") ==
NULL )
            return
ERR_DISCONNECT_ADDRESS_NOT_FOUND;
        DEBUGMSG("do_disconnect
address:"<<DEBUGADDRESS(do_disconnect)<<endl);
        DEBUGMSG("Getting do_nord function address from
"<<dbType<<" dll"<<endl);
        if (do_nord = (NORD_PTR)
GetProcAddress(dbInstance,"do_nord") == NULL)
            return
ERR_NORD_ADDRESS_NOT_FOUND;
        DEBUGMSG("do_nord function
address:"<<DEBUGADDRESS(do_nord)<<endl);
        DEBUGMSG("Getting do_pymt function address from
"<<dbType<<" dll"<<endl);
        if (do_pymt = (PYMT_PTR)
GetProcAddress(dbInstance,"do_pymt") == NULL)
            return
ERR_PYMT_ADDRESS_NOT_FOUND;
        DEBUGMSG("do_pymt function
address:"<<DEBUGADDRESS(do_pymt)<<endl);
        DEBUGMSG("Getting do_ords function address from
"<<dbType<<" dll"<<endl);
        if (do_ords = (ORDS_PTR)
GetProcAddress(dbInstance,"do_ords") == NULL)
            return
ERR_ORDS_ADDRESS_NOT_FOUND;
        DEBUGMSG("do_ords function
address:"<<DEBUGADDRESS(do_ords)<<endl);
        DEBUGMSG("Getting do_stok function address from
"<<dbType<<"
dll"<<endl);
        if (do_stok = (STOK_PTR)
GetProcAddress(dbInstance,"do_stok") == NULL)
            return
ERR_STOK_ADDRESS_NOT_FOUND;
        DEBUGMSG("do_stok function
address:"<<DEBUGADDRESS(do_stok)<<endl);
        DEBUGMSG("All function addresses retrieved
successfully."<<endl);
        }
        return OK;
    }
}
/*
*****
** Name          :          readRegistry()
** Description   :
**              :          Function reads registry
value
** Parameters   :
** Returns      :
**              :          int - return code
** Comments     :
**              :          Values retrieved from
registry
**              :          dbName, dbUserName,
and dbUserPassword
*/

```

```

*****
*/
Ctpcc_com::readRegistry()
{
    //open registry key
    HKEY    registryKey;
    DWORD   regType;
    char    value[MAX_STRING_LEN];
    DWORD   regValue;
    DWORD   regValueSize = MAX_STRING_LEN;
    DEBUGMSG("Entered readRegistry(), opening key:"<<
REGISTRY_SUB_KEY <<endl);
    //open up registry key

    if(RegOpenKeyEx(HKEY_LOCAL_MACHINE,REGISTRY_SUB_KEY,0,K
EY_READ,&registryKey) == ERROR_SUCCESS)
    {
        DEBUGMSG(REGISTRY_SUB_KEY<<" open,
getting database type from key"<<endl);
        regValueSize = sizeof(value);
        if
        (RegQueryValueEx(registryKey,DB_TYPE,0,&regType,(BYTE *)
&value,&regValueSize) == ERROR_SUCCESS )
            strcpy(dbType,value);
        DEBUGMSG("Database type:"<<dbType<<" from
registry key."<<endl);
        DEBUGMSG("Getting database name from registry
key."<<endl);
        regValueSize = sizeof(value);
        if
        (RegQueryValueEx(registryKey,DB_NAME,0,&regType,(BYTE *)
&value,&regValueSize) == ERROR_SUCCESS )
            strcpy(dbName,value);
        DEBUGMSG("Database name:"<<dbName<<endl);
        DEBUGMSG("Getting null database flag from
key."<<endl);
        regValueSize = sizeof(regValue);

        if(RegQueryValueEx(registryKey,NULL_DB,0,&regType,(BYTE
*)&regValue,&regValueSize) == ERROR_SUCCESS)
            nullDB = regValue;
        DEBUGMSG("Null database flag:"<<nullDB<<endl);
        return OK;
    }
    DEBUGMSG("Error, unable to open registry key."<<endl);
    return ERR_UNABLE_TO_OPEN_REG;
}
/*
*****
** Name          :          connectDB
** Description   :
**              :          Function connects to
the db
** Parameters   :
** Returns      :
**              :          int - return code
** Comments     :
**
*****
*/
Ctpcc_com::connectDB()
{
    DEBUGMSG("Entered connectDB(), checking if object is
connected."<<endl);
    if(!connected)
    {
        DEBUGMSG("Object not connected, calling
do_connection with dbName:"<<dbName<<" connectHandle:"<<

```

```

DEBUGADDRESS(connectHandle)<<endl);
        if(!connectHandleInUse)
        {
            DEBUGMSG("Setting Context handle in use
to true"<<endl);
            connectHandleInUse = 1;
            connected =
do_connection(dbName,&connectHandle);
            if(connected != OK)
            {
                DEBUGMSG("Object do_connect
failed, rc:"<<connected<<endl);
                ERRORMSG("Object do_connect
failed, rc:"<<connected<<endl);
                return connected;
            }
            DEBUGMSG("Object connection complete,
connectHandle:"<<DEBUGADDRESS(connectHandle)<<endl);
            connectHandleInUse = 0;
            return OK;
        }
        else
        {
            DEBUGMSG("Object's connectHandle
already in use, connect failed"<<endl);
            ERRORMSG("Object's connectHandle
already in use, connect failed"<<endl);
            return ERR_HANDLE_IN_USE;
        }
    }
    DEBUGMSG("Object already has connection established."<<endl);
    return OK;
}

```

tpccCom/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
*****/
#define PROXY_DELEGATION
#include <rpcproxy.h>
#ifdef _cplusplus

extern "C" {
#endif
EXTERN_PROXY_FILE( tpccCom )

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

DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )
#ifdef _cplusplus
} /*extern "C" */
#endif
/* end of generated dlldata file */

```

tpccCom/dlldatax.c

```

// wrapper for dlldata.c
#ifdef _MERGE_PROXYSTUB // merge proxy stub DLL

```

```

#define REGISTER_PROXY_DLL //DllRegisterServer, etc.
#define _WIN32_WINNT 0x0500 //for Win2000, change it to 0x0400
for NT4 or Win95 with DCOM
#define USE_STUBLESS_PROXY //defined only with MIDL switch
/Oicf
#pragma comment(lib, "rpcns4.lib")
#pragma comment(lib, "rpert4.lib")
#define ENTRY_PREFIX Prx
#include "dldata.c"
#include "tpccCom_p.c"
#endif // _MERGE_PROXYSTUB

```

tpccCom/tpccCom_i.c

```

/* 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 6.00.0361 */
/* at Wed Feb 11 08:32:46 2004
*/
/* Compiler settings for \tpccCom.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
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( )
#if !defined(_M_IA64) && !defined(_M_AMD64)

```

```

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

```

```

#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,b3,b4,b5,b6,b7,b8) \
    DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)
#else // !_MIDL_USE_GUIDDEF_
#ifndef __IID_DEFINED__
#define __IID_DEFINED__
typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char c[8];
} IID;
#endif // __IID_DEFINED__
#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED
#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
    const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}
#endif // !_MIDL_USE_GUIDDEF_

```

```

MIDL_DEFINE_GUID(IID,
IID_IComponentRegistrar,0xa817e7a2,0x43fa,0x11d0,0x9e,0x44,0x00,0xaa,0x
00,0xb6,0x77,0x0a);

```

```

MIDL_DEFINE_GUID(IID,
IID_Itpcc_com,0x5B4FA473,0x2E68,0x4D79,0xA6,0x26,0xF3,0x8B,0x30,0x
B8,0x19,0x6E);

```

```

MIDL_DEFINE_GUID(IID,
LIBID_tpccComLib,0x91F1B8B0,0x89E9,0x457B,0xA2,0x28,0x3E,0x2D,0x6
C,0xE3,0xE7,0x52);

```

```

MIDL_DEFINE_GUID(CLSID,
CLSID_CompReg,0x90EEDAFF,0xF8D3,0x4711,0x99,0xA9,0x8A,0xC3,0xC
0,0xFE,0x5D,0xB9);

```

```

MIDL_DEFINE_GUID(CLSID,
CLSID_tpcc_com,0x5F752BF2,0xF739,0x43D4,0x84,0x92,0x44,0xC1,0x95,0
x81,0xC0,0xA1);
#ifdef _MIDL_DEFINE_GUID
#ifndef __cplusplus
}
#endif
#endif

```

```

#endif /* !defined(_M_IA64) && !defined(_M_AMD64) */

```

tpccCom/tpccCom_p.c

```

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

```

```

/* File created by MIDL compiler version 6.00.0361 */
/* at Wed Feb 11 08:32:46 2004
*/
/* Compiler settings for \tpccCom.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
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( )
#if !defined(_M_IA64) && !defined(_M_AMD64)

```

```

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

```

```

#ifdef _MSC_VER >= 1200
#pragma warning(push)
#endif

```

```

#pragma warning( disable: 4100 ) /* unreferenced arguments in x86 call */
#pragma warning( disable: 4211 ) /* redefine extent to static */
#pragma warning( disable: 4232 ) /* dllimport identity */
#define USE_STUBLESS_PROXY

```

```

/* verify that the <rpcproxy.h> version is high enough to compile this file
*/

```

```

#ifdef _REDQ_RPCPROXY_H_VERSION_
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif

```

```

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

```

```

#include "tpccCom.h"
#define TYPE_FORMAT_STRING_SIZE 1089
#define PROC_FORMAT_STRING_SIZE 409

```

```

#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 2
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 IComponentRegistrar_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO
IComponentRegistrar_ProxyInfo;

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO Itpcc_com_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO Itpcc_com_ProxyInfo;

extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];
#if !defined(__RPC_WIN32_)
#error Invalid build platform for this stub.
#endif
#if !(TARGET_IS_NT50_OR_LATER)
#error You need a Windows 2000 or later to run this stub because it uses these
features:
#error /robust command line switch.
#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 =
{
    {
        /* Procedure Attach */
        0x33, /*
FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object,
Oi2 */
        /* 2 */ NdrFcLong( 0x0 ), /* 0 */
        /* 6 */ NdrFcShort( 0x7 ), /* 7 */
        /* 8 */ NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
        /* 10 */ NdrFcShort( 0x0 ), /* 0 */
        /* 12 */ NdrFcShort( 0x8 ), /* 8 */
        /* 14 */ 0x46, /* Oi2 Flags: clt must size, has return, has
ext, */
        0x2, /* 2 */
        /* 16 */ 0x8, /* 8 */
        0x5, /* Ext Flags: new corr
desc, srv corr check, */
        /* 18 */ NdrFcShort( 0x0 ), /* 0 */
        /* 20 */ NdrFcShort( 0x1 ), /* 1 */
    }
}

```

```

/* 22 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter bstrPath */
/* 24 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
/* 26 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 28 */ NdrFcShort( 0x1c ), /* Type Offset=28 */
/* Return value */
/* 30 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 32 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 34 */ 0x8, /* FC_LONG */
0x0, /* 0 */
/* Procedure doSetComplete */
/* Procedure RegisterAll */
/* 36 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 38 */ NdrFcLong( 0x0 ), /* 0 */
/* 42 */ NdrFcShort( 0x8 ), /* 8 */
/* 44 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 46 */ NdrFcShort( 0x0 ), /* 0 */
/* 48 */ NdrFcShort( 0x8 ), /* 8 */
/* 50 */ 0x44, /* Oi2 Flags: has return, has ext, */
0x1, /* 1 */
/* 52 */ 0x8, /* 8 */
0x1, /* Ext Flags: new corr
desc, */
/* 54 */ NdrFcShort( 0x0 ), /* 0 */
/* 56 */ NdrFcShort( 0x0 ), /* 0 */
/* 58 */ NdrFcShort( 0x0 ), /* 0 */
/* Return value */
/* Return value */
/* 60 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 62 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 64 */ 0x8, /* FC_LONG */
0x0, /* 0 */
/* Procedure UnregisterAll */
/* 66 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 68 */ NdrFcLong( 0x0 ), /* 0 */
/* 72 */ NdrFcShort( 0x9 ), /* 9 */
/* 74 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 76 */ NdrFcShort( 0x0 ), /* 0 */
/* 78 */ NdrFcShort( 0x8 ), /* 8 */
/* 80 */ 0x44, /* Oi2 Flags: has return, has ext, */
0x1, /* 1 */
/* 82 */ 0x8, /* 8 */
0x1, /* Ext Flags: new corr
desc, */
/* 84 */ NdrFcShort( 0x0 ), /* 0 */
/* 86 */ NdrFcShort( 0x0 ), /* 0 */
/* 88 */ NdrFcShort( 0x0 ), /* 0 */
/* Return value */
/* 90 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 92 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 94 */ 0x8, /* FC_LONG */
0x0, /* 0 */
/* Procedure GetComponents */
/* 96 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 98 */ NdrFcLong( 0x0 ), /* 0 */
/* 102 */ NdrFcShort( 0xa ), /* 10 */
/* 104 */ NdrFcShort( 0x10 ), /* x86 Stack size/offset = 16 */
/* 106 */ NdrFcShort( 0x0 ), /* 0 */
/* 108 */ NdrFcShort( 0x8 ), /* 8 */

```

```

/* 110 */ 0x45, /* Oi2 Flags: srv must size, has return, has
ext, */
/* 112 */ 0x8, /* 8 */
/* 3 */
0x3, /* Ext Flags: new corr
desc, clt corr check, */
/* 114 */ NdrFcShort( 0x24 ), /* 36 */
/* 116 */ NdrFcShort( 0x0 ), /* 0 */
/* 118 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter pbstrCLSIDs */
/* 120 */ NdrFcShort( 0x2113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=8 */
/* 122 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 124 */ NdrFcShort( 0x41e ), /* Type Offset=1054 */
/* Parameter pbstrDescriptions */
/* 126 */ NdrFcShort( 0x2113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=8 */
/* 128 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 130 */ NdrFcShort( 0x41e ), /* Type Offset=1054 */
/* Return value */
/* 132 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 134 */ NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 136 */ 0x8, /* FC_LONG */
0x0, /* 0 */
/* Procedure RegisterComponent */
/* 138 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 140 */ NdrFcLong( 0x0 ), /* 0 */
/* 144 */ NdrFcShort( 0xb ), /* 11 */
/* 146 */ NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 148 */ NdrFcShort( 0x0 ), /* 0 */
/* 150 */ NdrFcShort( 0x8 ), /* 8 */
/* 152 */ 0x46, /* Oi2 Flags: clt must size, has return, has
ext, */
0x2, /* 2 */
/* 154 */ 0x8, /* 8 */
0x5, /* Ext Flags: new corr
desc, srv corr check, */
/* 156 */ NdrFcShort( 0x0 ), /* 0 */
/* 158 */ NdrFcShort( 0x1 ), /* 1 */
/* 160 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter bstrCLSID */
/* 162 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
/* 164 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 166 */ NdrFcShort( 0x1c ), /* Type Offset=28 */
/* Return value */
/* 168 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 170 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 172 */ 0x8, /* FC_LONG */
0x0, /* 0 */
/* Procedure UnregisterComponent */
/* 174 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 176 */ NdrFcLong( 0x0 ), /* 0 */
/* 180 */ NdrFcShort( 0xc ), /* 12 */
/* 182 */ NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 184 */ NdrFcShort( 0x0 ), /* 0 */
/* 186 */ NdrFcShort( 0x8 ), /* 8 */
/* 188 */ 0x46, /* Oi2 Flags: clt must size, has return, has
ext, */
0x2, /* 2 */
/* 190 */ 0x8, /* 8 */
0x5, /* Ext Flags: new corr
desc, srv corr check, */
/* 192 */ NdrFcShort( 0x0 ), /* 0 */
/* 194 */ NdrFcShort( 0x1 ), /* 1 */
/* 196 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter bstrCLSID */
/* 198 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
/* 200 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 202 */ NdrFcShort( 0x1c ), /* Type Offset=28 */
/* Return value */
/* 204 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 206 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 208 */ 0x8, /* FC_LONG */
0x0, /* 0 */
/* Procedure doStockLevel */
/* 210 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 212 */ NdrFcLong( 0x0 ), /* 0 */
/* 216 */ NdrFcShort( 0x3 ), /* 3 */
/* 218 */ NdrFcShort( 0x10 ), /* x86 Stack size/offset = 16 */
/* 220 */ NdrFcShort( 0x1c ), /* 28 */
/* 222 */ NdrFcShort( 0x8 ), /* 8 */
/* 224 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
0x3, /* 3 */
/* 226 */ 0x8, /* 8 */
0x7, /* Ext Flags: new corr
desc, clt corr check, srv corr check, */
/* 228 */ NdrFcShort( 0x1 ), /* 1 */
/* 230 */ NdrFcShort( 0x1 ), /* 1 */
/* 232 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter size */
/* 234 */ NdrFcShort( 0x148 ), /* Flags: in, base type, simple ref, */
/* 236 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 238 */ 0x8, /* FC_LONG */
0x0, /* 0 */
/* Parameter buffer */
/* 240 */ NdrFcShort( 0x201b ), /* Flags: must size, must free, in,
out, srv alloc size=8 */
/* 242 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 244 */ NdrFcShort( 0x42c ), /* Type Offset=1068 */
/* Return value */
/* 246 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 248 */ NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 250 */ 0x8, /* FC_LONG */
0x0, /* 0 */
/* Procedure doNewOrder */
/* 252 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 254 */ NdrFcLong( 0x0 ), /* 0 */
/* 258 */ NdrFcShort( 0x4 ), /* 4 */
/* 260 */ NdrFcShort( 0x10 ), /* x86 Stack size/offset = 16 */
/* 262 */ NdrFcShort( 0x1c ), /* 28 */
/* 264 */ NdrFcShort( 0x8 ), /* 8 */
/* 266 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
0x3, /* 3 */
/* 268 */ 0x8, /* 8 */
0x7, /* Ext Flags: new corr
desc, clt corr check, srv corr check, */
/* 270 */ NdrFcShort( 0x1 ), /* 1 */
/* 272 */ NdrFcShort( 0x1 ), /* 1 */
/* 274 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter size */
/* 276 */ NdrFcShort( 0x148 ), /* Flags: in, base type, simple ref, */
/* 278 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 280 */ 0x8, /* FC_LONG */
0x0, /* 0 */
/* Parameter buffer */

```

```

/* 282 */ NdrFcShort( 0x201b ), /* Flags: must size, must free, in,
out, srv alloc size=8 */
/* 284 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 286 */ NdrFcShort( 0x42c ), /* Type Offset=1068 */
/* Return value */
/* 288 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 290 */ NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 292 */ 0x8, /* FC_LONG */
/* Procedure doPayment */
/* 294 */ 0x33, /* FC_AUTO_HANDLE */
/* Old Flags: object,
0x0, /* 0 */
0x0, /* 0 */
0x6c, /* Old Flags: object,
Oi2 */
/* 296 */ NdrFcLong( 0x0 ), /* 0 */
/* 300 */ NdrFcShort( 0x5 ), /* 5 */
/* 302 */ NdrFcShort( 0x10 ), /* x86 Stack size/offset = 16 */
/* 304 */ NdrFcShort( 0x1c ), /* 28 */
/* 306 */ NdrFcShort( 0x8 ), /* 8 */
/* 308 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
/* 310 */ 0x8, /* 3 */
/* 8 */
0x7, /* Ext Flags: new corr
desc, clt corr check, srv corr check, */
/* 312 */ NdrFcShort( 0x1 ), /* 1 */
/* 314 */ NdrFcShort( 0x1 ), /* 1 */
/* 316 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter size */
/* 318 */ NdrFcShort( 0x148 ), /* Flags: in, base type, simple ref, */
/* 320 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 322 */ 0x8, /* FC_LONG */
/* 0x0, /* 0 */
/* Parameter buffer */
/* 324 */ NdrFcShort( 0x201b ), /* Flags: must size, must free, in,
out, srv alloc size=8 */
/* 326 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 328 */ NdrFcShort( 0x42c ), /* Type Offset=1068 */
/* Return value */
/* 330 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 332 */ NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 334 */ 0x8, /* FC_LONG */
/* 0x0, /* 0 */
/* Procedure doOrderStatus */
/* 336 */ 0x33, /* FC_AUTO_HANDLE */
/* Old Flags: object,
0x6c, /* Old Flags: object,
Oi2 */
/* 338 */ NdrFcLong( 0x0 ), /* 0 */
/* 342 */ NdrFcShort( 0x6 ), /* 6 */
/* 344 */ NdrFcShort( 0x10 ), /* x86 Stack size/offset = 16 */
/* 346 */ NdrFcShort( 0x1c ), /* 28 */
/* 348 */ NdrFcShort( 0x8 ), /* 8 */
/* 350 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
/* 352 */ 0x8, /* 3 */
/* 8 */
0x7, /* Ext Flags: new corr
desc, clt corr check, srv corr check, */
/* 354 */ NdrFcShort( 0x1 ), /* 1 */
/* 356 */ NdrFcShort( 0x1 ), /* 1 */
/* 358 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter size */
/* 360 */ NdrFcShort( 0x148 ), /* Flags: in, base type, simple ref, */
/* 362 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 364 */ 0x8, /* FC_LONG */
/* 0x0, /* 0 */
/* Parameter buffer */
/* 366 */ NdrFcShort( 0x201b ), /* Flags: must size, must free, in,
out, srv alloc size=8 */
/* 368 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 370 */ NdrFcShort( 0x42c ), /* Type Offset=1068 */
/* Return value */
/* 372 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 374 */ NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 376 */ 0x8, /* FC_LONG */
/* 0x0, /* 0 */
/* Procedure doDBInfo */
/* 378 */ 0x33, /* FC_AUTO_HANDLE */
/* Old Flags: object,
0x6c, /* Old Flags: object,
Oi2 */
/* 380 */ NdrFcLong( 0x0 ), /* 0 */
/* 384 */ NdrFcShort( 0x7 ), /* 7 */
/* 386 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 388 */ NdrFcShort( 0x0 ), /* 0 */
/* 390 */ NdrFcShort( 0x8 ), /* 8 */
/* 392 */ 0x44, /* Oi2 Flags: has return, has ext, */
/* 0x1, /* 1 */
/* 8 */
0x1, /* Ext Flags: new corr
desc, */
/* 396 */ NdrFcShort( 0x0 ), /* 0 */
/* 398 */ NdrFcShort( 0x0 ), /* 0 */
/* 400 */ NdrFcShort( 0x0 ), /* 0 */
/* Return value */
/* 402 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 404 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 406 */ 0x8, /* FC_LONG */
/* 0x0, /* 0 */
0x0
}
};
static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /* 0 */
/* 2 */
0x12, 0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0xe ), /* Offset= 14 (18) */
/* 6 */
0x1b, /* FC_CARRAY */
0x1, /* 1 */
/* 8 */ NdrFcShort( 0x2 ), /* 2 */
/* 10 */ 0x9, /* Corr desc: FC_ULONG */
0x0, /* */
/* 12 */ NdrFcShort( 0xffc ), /* -4 */
/* 14 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 16 */ 0x6, /* FC_SHORT */
0x5b, /* FC_END */
/* 18 */
0x17, /* FC_CSTRUCT */
0x3, /* 3 */
/* 20 */ NdrFcShort( 0x8 ), /* 8 */
/* 22 */ NdrFcShort( 0xffff0 ), /* Offset= -16 (6) */
/* 24 */ 0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 26 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 28 */ 0xb4, /* FC_USER_MARSHAL */
0x83, /* 131 */
/* 30 */ NdrFcShort( 0x0 ), /* 0 */
/* 32 */ NdrFcShort( 0x4 ), /* 4 */
/* 34 */ NdrFcShort( 0x0 ), /* 0 */
/* 36 */ NdrFcShort( 0xffde ), /* Offset= -34 (2) */
/* 38 */
0x11, 0x4, /* FC_RP [allocated_on_stack] */
/* 40 */ NdrFcShort( 0x3f6 ), /* Offset= 1014 (1054) */

```



```

/* 42 */
[pointer_deref] */
/* 44 */ NdrFcShort( 0x2 ), /* Offset= 2 (46) */
/* 46 */
0x13, 0x0, /* FC_OP */
/* 48 */ NdrFcShort( 0x3dc ), /* Offset= 988 (1036) */
/* 50 */
0x2a, /*
FC_ENCAPSULATED_UNION */
0x49, /* 73 */
/* 52 */ NdrFcShort( 0x18 ), /* 24 */
/* 54 */ NdrFcShort( 0xa ), /* 10 */
/* 56 */ NdrFcLong( 0x8 ), /* 8 */
/* 60 */ NdrFcShort( 0x5a ), /* Offset= 90 (150) */
/* 62 */ NdrFcLong( 0xd ), /* 13 */
/* 66 */ NdrFcShort( 0x90 ), /* Offset= 144 (210) */
/* 68 */ NdrFcLong( 0x9 ), /* 9 */
/* 72 */ NdrFcShort( 0xc2 ), /* Offset= 194 (266) */
/* 74 */ NdrFcLong( 0xc ), /* 12 */
/* 78 */ NdrFcShort( 0x2c0 ), /* Offset= 704 (782) */
/* 80 */ NdrFcLong( 0x24 ), /* 36 */
/* 84 */ NdrFcShort( 0x2ea ), /* Offset= 746 (830) */
/* 86 */ NdrFcLong( 0x800d ), /* 32781 */
/* 90 */ NdrFcShort( 0x306 ), /* Offset= 774 (864) */
/* 92 */ NdrFcLong( 0x10 ), /* 16 */
/* 96 */ NdrFcShort( 0x320 ), /* Offset= 800 (896) */
/* 98 */ NdrFcLong( 0x2 ), /* 2 */
/* 102 */ NdrFcShort( 0x33a ), /* Offset= 826 (928) */
/* 104 */ NdrFcLong( 0x3 ), /* 3 */
/* 108 */ NdrFcShort( 0x354 ), /* Offset= 852 (960) */
/* 110 */ NdrFcLong( 0x14 ), /* 20 */
/* 114 */ NdrFcShort( 0x36e ), /* Offset= 878 (992) */
/* 116 */ NdrFcShort( 0xffff ), /* Offset= -1 (115) */
/* 118 */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 120 */ NdrFcShort( 0x4 ), /* 4 */
/* 122 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 124 */ NdrFcShort( 0x0 ), /* 0 */
/* 126 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 128 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 130 */
0x48, /*
FC_VARIABLE_REPEAT */
0x49, /*
FC_FIXED_OFFSET */
/* 132 */ NdrFcShort( 0x4 ), /* 4 */
/* 134 */ NdrFcShort( 0x0 ), /* 0 */
/* 136 */ NdrFcShort( 0x1 ), /* 1 */
/* 138 */ NdrFcShort( 0x0 ), /* 0 */
/* 140 */ NdrFcShort( 0x0 ), /* 0 */
/* 142 */ 0x13, 0x0, /* FC_OP */
/* 144 */ NdrFcShort( 0xff82 ), /* Offset= -126 (18) */
/* 146 */
0x5b, /* FC_END */
0x8, /* FC_LONG */
/* 148 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 150 */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 152 */ NdrFcShort( 0x8 ), /* 8 */
/* 154 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 156 */
0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 158 */ NdrFcShort( 0x4 ), /* 4 */
/* 160 */ NdrFcShort( 0x4 ), /* 4 */
/* 162 */ 0x11, 0x0, /* FC_RP */
/* 164 */ NdrFcShort( 0xffd2 ), /* Offset= -46 (118) */
/* 166 */
0x5b, /* FC_END */
0x8, /* FC_LONG */
/* 168 */ 0x8, /* FC_LONG */
0x5b, /* FC_END */
/* 170 */
0x2f, /* FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 172 */ NdrFcLong( 0x0 ), /* 0 */
/* 176 */ NdrFcShort( 0x0 ), /* 0 */
/* 178 */ NdrFcShort( 0x0 ), /* 0 */
/* 180 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 182 */ 0x0, /* 0 */
0x0, /* 0 */
/* 184 */ 0x0, /* 0 */
0x0, /* 0 */
/* 186 */ 0x0, /* 0 */
0x46, /* 70 */
/* 188 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /* 3 */
/* 190 */ NdrFcShort( 0x0 ), /* 0 */
/* 192 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 194 */ NdrFcShort( 0x0 ), /* 0 */
/* 196 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 198 */ NdrFcLong( 0xfffffff ), /* -1 */
/* 202 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 204 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 206 */ NdrFcShort( 0xffdc ), /* Offset= -36 (170) */
/* 208 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 210 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 212 */ NdrFcShort( 0x8 ), /* 8 */
/* 214 */ NdrFcShort( 0x0 ), /* 0 */
/* 216 */ NdrFcShort( 0x6 ), /* Offset= 6 (222) */
/* 218 */ 0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 220 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 222 */
0x11, 0x0, /* FC_RP */
/* 224 */ NdrFcShort( 0xffdc ), /* Offset= -36 (188) */
/* 226 */
0x2f, /* FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 228 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 232 */ NdrFcShort( 0x0 ), /* 0 */
/* 234 */ NdrFcShort( 0x0 ), /* 0 */
/* 236 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 238 */ 0x0, /* 0 */

```

```

0x0, /* 0 */
/* 240 */ 0x0, /* 0 */
/* 242 */ 0x0, /* 0 */
/* 244 */ 0x46, /* 70 */
FC_BOGUS_ARRAY */
0x21, /*
/* 246 */ NdrFcShort( 0x0 ), /* 0 */
/* 248 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 250 */ NdrFcShort( 0x0 ), /* 0 */
/* 252 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 254 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 258 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 260 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 262 */ NdrFcShort( 0xffdc ), /* Offset= -36 (226) */
/* 264 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 266 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 268 */ NdrFcShort( 0x8 ), /* 8 */
/* 270 */ NdrFcShort( 0x0 ), /* 0 */
/* 272 */ NdrFcShort( 0x6 ), /* Offset= 6 (278) */
/* 274 */ 0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 276 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 278 */
0x11,0x0, /* FC_RP */
/* 280 */ NdrFcShort( 0xffdc ), /* Offset= -36 (244) */
/* 282 */
0x2b, /*
FC_NON_ENCAPSULATED_UNION */
0x9, /* FC_ULONG */
/* 284 */ 0x7, /* Corr desc: FC_USHORT */
0x0, /* */
/* 286 */ NdrFcShort( 0xffff8 ), /* -8 */
/* 288 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 290 */ NdrFcShort( 0x2 ), /* Offset= 2 (292) */
/* 292 */ NdrFcShort( 0x10 ), /* 16 */
/* 294 */ NdrFcShort( 0x2f ), /* 47 */
/* 296 */ NdrFcLong( 0x14 ), /* 20 */
/* 300 */ NdrFcShort( 0x800b ), /* Simple arm type: FC_HYPER */
/* 302 */ NdrFcLong( 0x3 ), /* 3 */
/* 306 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 308 */ NdrFcLong( 0x11 ), /* 17 */
/* 312 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 314 */ NdrFcLong( 0x2 ), /* 2 */
/* 318 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 320 */ NdrFcLong( 0x4 ), /* 4 */
/* 324 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 326 */ NdrFcLong( 0x5 ), /* 5 */
/* 330 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/*
/* 332 */ NdrFcLong( 0xb ), /* 11 */
/* 336 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 338 */ NdrFcLong( 0xa ), /* 10 */
/* 342 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 344 */ NdrFcLong( 0x6 ), /* 6 */
/* 348 */ NdrFcShort( 0xe8 ), /* Offset= 232 (580) */
/* 350 */ NdrFcLong( 0x7 ), /* 7 */
/* 354 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/*
/* 356 */ NdrFcLong( 0x8 ), /* 8 */
/* 360 */ NdrFcShort( 0xe2 ), /* Offset= 226 (586) */
/* 362 */ NdrFcLong( 0xd ), /* 13 */
/* 366 */ NdrFcShort( 0xff3c ), /* Offset= -196 (170) */
/* 368 */ NdrFcLong( 0x9 ), /* 9 */
/* 372 */ NdrFcShort( 0xff6e ), /* Offset= -146 (226) */
/* 374 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 378 */ NdrFcShort( 0xd4 ), /* Offset= 212 (590) */
/* 380 */ NdrFcLong( 0x24 ), /* 36 */
/* 384 */ NdrFcShort( 0xd6 ), /* Offset= 214 (598) */
/* 386 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 390 */ NdrFcShort( 0xd0 ), /* Offset= 208 (598) */
/* 392 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 396 */ NdrFcShort( 0x100 ), /* Offset= 256 (652) */
/* 398 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 402 */ NdrFcShort( 0xfe ), /* Offset= 254 (656) */
/* 404 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 408 */ NdrFcShort( 0xfc ), /* Offset= 252 (660) */
/* 410 */ NdrFcLong( 0x4014 ), /* 16404 */
/* 414 */ NdrFcShort( 0xfa ), /* Offset= 250 (664) */
/* 416 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 420 */ NdrFcShort( 0xf8 ), /* Offset= 248 (668) */
/* 422 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 426 */ NdrFcShort( 0xf6 ), /* Offset= 246 (672) */
/* 428 */ NdrFcLong( 0x400b ), /* 16395 */
/* 432 */ NdrFcShort( 0xe0 ), /* Offset= 224 (656) */
/* 434 */ NdrFcLong( 0x400a ), /* 16394 */
/* 438 */ NdrFcShort( 0xde ), /* Offset= 222 (660) */
/* 440 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 444 */ NdrFcShort( 0xe8 ), /* Offset= 232 (676) */
/* 446 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 450 */ NdrFcShort( 0xde ), /* Offset= 222 (672) */
/* 452 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 456 */ NdrFcShort( 0xe0 ), /* Offset= 224 (680) */
/* 458 */ NdrFcLong( 0x400d ), /* 16397 */
/* 462 */ NdrFcShort( 0xde ), /* Offset= 222 (684) */
/* 464 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 468 */ NdrFcShort( 0xdc ), /* Offset= 220 (688) */
/* 470 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 474 */ NdrFcShort( 0xda ), /* Offset= 218 (692) */
/* 476 */ NdrFcLong( 0x400c ), /* 16396 */
/* 480 */ NdrFcShort( 0xe0 ), /* Offset= 224 (704) */
/* 482 */ NdrFcLong( 0x10 ), /* 16 */
/* 486 */ NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 488 */ NdrFcLong( 0x12 ), /* 18 */
/* 492 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 494 */ NdrFcLong( 0x13 ), /* 19 */
/* 498 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 500 */ NdrFcLong( 0x15 ), /* 21 */
/* 504 */ NdrFcShort( 0x800b ), /* Simple arm type: FC_HYPER */
/* 506 */ NdrFcLong( 0x16 ), /* 22 */
/* 510 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 512 */ NdrFcLong( 0x17 ), /* 23 */
/* 516 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 518 */ NdrFcLong( 0xe ), /* 14 */
/* 522 */ NdrFcShort( 0xbe ), /* Offset= 190 (712) */
/* 524 */ NdrFcLong( 0x400e ), /* 16398 */
/* 528 */ NdrFcShort( 0xc2 ), /* Offset= 194 (722) */
/* 530 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 534 */ NdrFcShort( 0xc0 ), /* Offset= 192 (726) */
/* 536 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 540 */ NdrFcShort( 0x74 ), /* Offset= 116 (656) */
/* 542 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 546 */ NdrFcShort( 0x72 ), /* Offset= 114 (660) */
/* 548 */ NdrFcLong( 0x4015 ), /* 16405 */
/* 552 */ NdrFcShort( 0x70 ), /* Offset= 112 (664) */
/* 554 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 558 */ NdrFcShort( 0x66 ), /* Offset= 102 (660) */

```

```

/* 560 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 564 */ NdrFcShort( 0x60 ), /* Offset= 96 (660) */
/* 566 */ NdrFcLong( 0x0 ), /* 0 */
/* 570 */ NdrFcShort( 0x0 ), /* Offset= 0 (570) */
/* 572 */ NdrFcLong( 0x1 ), /* 1 */
/* 576 */ NdrFcShort( 0x0 ), /* Offset= 0 (576) */
/* 578 */ NdrFcShort( 0xffff ), /* Offset= -1 (577) */
/* 580 */
                                0x15, /* FC_STRUCT */
                                0x7, /* 7 */
/* 582 */ NdrFcShort( 0x8 ), /* 8 */
/* 584 */ 0xb, /* FC_HYPER */
                                0x5b, /* FC_END */
/* 586 */
                                0x13, 0x0, /* FC_OP */
/* 588 */ NdrFcShort( 0xfdc6 ), /* Offset= -570 (18) */
/* 590 */
                                0x13, 0x10, /* FC_OP
[pointer_deref] */
/* 592 */ NdrFcShort( 0x2 ), /* Offset= 2 (594) */
/* 594 */
                                0x13, 0x0, /* FC_OP */
/* 596 */ NdrFcShort( 0x1b8 ), /* Offset= 440 (1036) */
/* 598 */
                                0x13, 0x0, /* FC_OP */
/* 600 */ NdrFcShort( 0x20 ), /* Offset= 32 (632) */
/* 602 */
                                0x2f, /* FC_IP */
                                0x5a, /*
FC_CONSTANT_IID */
/* 604 */ NdrFcLong( 0x2f ), /* 47 */
/* 608 */ NdrFcShort( 0x0 ), /* 0 */
/* 610 */ NdrFcShort( 0x0 ), /* 0 */
/* 612 */ 0xc0, /* 192 */
                                0x0, /* 0 */
/* 614 */ 0x0, /* 0 */
                                0x0, /* 0 */
/* 616 */ 0x0, /* 0 */
                                0x0, /* 0 */
/* 618 */ 0x0, /* 0 */
                                0x46, /* 70 */
/* 620 */
                                0x1b, /* FC_CARRAY */
                                0x0, /* 0 */
/* 622 */ NdrFcShort( 0x1 ), /* 1 */
/* 624 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /*
/* 626 */ NdrFcShort( 0x4 ), /* 4 */
/* 628 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 630 */ 0x1, /* FC_BYTE */
                                0x5b, /* FC_END */
/* 632 */
                                0x1a, /*
FC_BOGUS_STRUCT */
                                0x3, /* 3 */
/* 634 */ NdrFcShort( 0x10 ), /* 16 */
/* 636 */ NdrFcShort( 0x0 ), /* 0 */
/* 638 */ NdrFcShort( 0xa ), /* Offset= 10 (648) */
/* 640 */ 0x8, /* FC_LONG */
                                0x8, /* FC_LONG */
/* 642 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
                                0x0, /* 0 */
/* 644 */ NdrFcShort( 0xffd6 ), /* Offset= -42 (602) */
/* 646 */ 0x36, /* FC_POINTER */
                                0x5b, /* FC_END */
/* 648 */
                                0x13, 0x0, /* FC_OP */
/* 650 */ NdrFcShort( 0xffe2 ), /* Offset= -30 (620) */
/* 652 */
                                0x13, 0x8, /* FC_OP [simple_pointer] */
/* 654 */ 0x1, /* FC_BYTE */
                                0x5c, /* FC_PAD */
/* 656 */
                                0x13, 0x8, /* FC_OP [simple_pointer] */
/* 658 */ 0x6, /* FC_SHORT */
                                0x5c, /* FC_PAD */
/* 660 */
                                0x13, 0x8, /* FC_OP [simple_pointer] */
/* 662 */ 0x8, /* FC_LONG */
                                0x5c, /* FC_PAD */
/* 664 */
                                0x13, 0x8, /* FC_OP [simple_pointer] */
/* 666 */ 0xb, /* FC_HYPER */
                                0x5c, /* FC_PAD */
/* 668 */
                                0x13, 0x8, /* FC_OP [simple_pointer] */
/* 670 */ 0xa, /* FC_FLOAT */
                                0x5c, /* FC_PAD */
/* 672 */
                                0x13, 0x8, /* FC_OP [simple_pointer] */
/* 674 */ 0xc, /* FC_DOUBLE */
                                0x5c, /* FC_PAD */
/* 676 */
                                0x13, 0x0, /* FC_OP */
/* 678 */ NdrFcShort( 0xff9e ), /* Offset= -98 (580) */
/* 680 */
                                0x13, 0x10, /* FC_OP
[pointer_deref] */
/* 682 */ NdrFcShort( 0xffa0 ), /* Offset= -96 (586) */
/* 684 */
                                0x13, 0x10, /* FC_OP
[pointer_deref] */
/* 686 */ NdrFcShort( 0xfdfc ), /* Offset= -516 (170) */
/* 688 */
                                0x13, 0x10, /* FC_OP
[pointer_deref] */
/* 690 */ NdrFcShort( 0xfe30 ), /* Offset= -464 (226) */
/* 692 */
                                0x13, 0x10, /* FC_OP
[pointer_deref] */
/* 694 */ NdrFcShort( 0x2 ), /* Offset= 2 (696) */
/* 696 */
                                0x13, 0x10, /* FC_OP
[pointer_deref] */
/* 698 */ NdrFcShort( 0x2 ), /* Offset= 2 (700) */
/* 700 */
                                0x13, 0x0, /* FC_OP */
/* 702 */ NdrFcShort( 0x14e ), /* Offset= 334 (1036) */
/* 704 */
                                0x13, 0x10, /* FC_OP
[pointer_deref] */
/* 706 */ NdrFcShort( 0x2 ), /* Offset= 2 (708) */
/* 708 */
                                0x13, 0x0, /* FC_OP */
/* 710 */ NdrFcShort( 0x14 ), /* Offset= 20 (730) */
/* 712 */
                                0x15, /* FC_STRUCT */
                                0x7, /* 7 */
/* 714 */ NdrFcShort( 0x10 ), /* 16 */
/* 716 */ 0x6, /* FC_SHORT */
                                0x1, /* FC_BYTE */
/* 718 */ 0x1, /* FC_BYTE */
                                0x8, /* FC_LONG */
/* 720 */ 0xb, /* FC_HYPER */
                                0x5b, /* FC_END */
/* 722 */

```

```

0x13, 0x0, /* FC_OP */
/* 724 */ NdrFcShort( 0xffff4 ), /* Offset= -12 (712) */
/* 726 */
0x13, 0x8, /* FC_OP [simple_pointer] */
/* 728 */ 0x2, /* FC_CHAR */
0x5c, /* FC_PAD */
/* 730 */
0x1a, /*
FC_BOGUS_STRUCT */
0x7, /* 7 */
/* 732 */ NdrFcShort( 0x20 ), /* 32 */
/* 734 */ NdrFcShort( 0x0 ), /* 0 */
/* 736 */ NdrFcShort( 0x0 ), /* Offset= 0 (736) */
/* 738 */ 0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 740 */ 0x6, /* FC_SHORT */
0x6, /* FC_SHORT */
/* 742 */ 0x6, /* FC_SHORT */
0x6, /* FC_SHORT */
/* 744 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 746 */ NdrFcShort( 0xfe30 ), /* Offset= -464 (282) */
/* 748 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 750 */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 752 */ NdrFcShort( 0x4 ), /* 4 */
/* 754 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 756 */ NdrFcShort( 0x0 ), /* 0 */
/* 758 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 760 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 762 */
0x48, /*
FC_VARIABLE_REPEAT */
0x49, /*
FC_FIXED_OFFSET */
/* 764 */ NdrFcShort( 0x4 ), /* 4 */
/* 766 */ NdrFcShort( 0x0 ), /* 0 */
/* 768 */ NdrFcShort( 0x1 ), /* 1 */
/* 770 */ NdrFcShort( 0x0 ), /* 0 */
/* 772 */ NdrFcShort( 0x0 ), /* 0 */
/* 774 */ 0x13, 0x0, /* FC_OP */
/* 776 */ NdrFcShort( 0xffd2 ), /* Offset= -46 (730) */
/* 778 */
0x5b, /* FC_END */
0x8, /* FC_LONG */
/* 780 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 782 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 784 */ NdrFcShort( 0x8 ), /* 8 */
/* 786 */ NdrFcShort( 0x0 ), /* 0 */
/* 788 */ NdrFcShort( 0x6 ), /* Offset= 6 (794) */
/* 790 */ 0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 792 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 794 */
0x11, 0x0, /* FC_RP */
/* 796 */ NdrFcShort( 0xffd2 ), /* Offset= -46 (750) */
/* 798 */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 800 */ NdrFcShort( 0x4 ), /* 4 */
/* 802 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 804 */ NdrFcShort( 0x0 ), /* 0 */
/* 806 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 808 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 810 */
0x48, /*
FC_VARIABLE_REPEAT */
0x49, /*
FC_FIXED_OFFSET */
/* 812 */ NdrFcShort( 0x4 ), /* 4 */
/* 814 */ NdrFcShort( 0x0 ), /* 0 */
/* 816 */ NdrFcShort( 0x1 ), /* 1 */
/* 818 */ NdrFcShort( 0x0 ), /* 0 */
/* 820 */ NdrFcShort( 0x0 ), /* 0 */
/* 822 */ 0x13, 0x0, /* FC_OP */
/* 824 */ NdrFcShort( 0xff40 ), /* Offset= -192 (632) */
/* 826 */
0x5b, /* FC_END */
0x8, /* FC_LONG */
/* 828 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 830 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 832 */ NdrFcShort( 0x8 ), /* 8 */
/* 834 */ NdrFcShort( 0x0 ), /* 0 */
/* 836 */ NdrFcShort( 0x6 ), /* Offset= 6 (842) */
/* 838 */ 0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 840 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 842 */
0x11, 0x0, /* FC_RP */
/* 844 */ NdrFcShort( 0xffd2 ), /* Offset= -46 (798) */
/* 846 */
0x1d, /* FC_SMFARRAY */
0x0, /* 0 */
/* 848 */ NdrFcShort( 0x8 ), /* 8 */
/* 850 */ 0x1, /* FC_BYTE */
0x5b, /* FC_END */
/* 852 */
0x15, /* FC_STRUCT */
0x3, /* 3 */
/* 854 */ NdrFcShort( 0x10 ), /* 16 */
/* 856 */ 0x8, /* FC_LONG */
0x6, /* FC_SHORT */
/* 858 */ 0x6, /* FC_SHORT */
0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 860 */ 0x0, /* 0 */
NdrFcShort( 0xffff1 ), /* Offset= -15 (846) */
0x5b, /* FC_END */
/* 864 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 866 */ NdrFcShort( 0x18 ), /* 24 */
/* 868 */ NdrFcShort( 0x0 ), /* 0 */
/* 870 */ NdrFcShort( 0xa ), /* Offset= 10 (880) */
/* 872 */ 0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 874 */ 0x4c, /* FC_EMBEDDED_COMPLEX */

```

```

0x0, /* 0 */
/* 876 */ NdrFcShort( 0xffe8 ), /* Offset= -24 (852) */
/* 878 */ 0x5c, /* FC_PAD */
/* 880 */ 0x5b, /* FC_END */
/* 882 */ NdrFcShort( 0xfd4a ), /* FC_RP */
/* 884 */ /* Offset= -694 (188) */
0x1b, /* FC_CARRAY */
0x0, /* 0 */
/* 886 */ NdrFcShort( 0x1 ), /* 1 */
/* 888 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 890 */ NdrFcShort( 0x0 ), /* 0 */
/* 892 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 894 */ 0x1, /* FC_BYTE */
/* 896 */ 0x5b, /* FC_END */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 898 */ NdrFcShort( 0x8 ), /* 8 */
/* 900 */ 0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 902 */ 0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 904 */ NdrFcShort( 0x4 ), /* 4 */
/* 906 */ NdrFcShort( 0x4 ), /* 4 */
/* 908 */ 0x13, 0x0, /* FC_OP */
/* 910 */ NdrFcShort( 0xffe6 ), /* Offset= -26 (884) */
/* 912 */ 0x5b, /* FC_END */
0x8, /* FC_LONG */
/* 914 */ 0x8, /* FC_LONG */
/* 916 */ 0x5b, /* FC_END */
0x1b, /* FC_CARRAY */
0x1, /* 1 */
/* 918 */ NdrFcShort( 0x2 ), /* 2 */
/* 920 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 922 */ NdrFcShort( 0x0 ), /* 0 */
/* 924 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 926 */ 0x6, /* FC_SHORT */
/* 928 */ 0x5b, /* FC_END */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 930 */ NdrFcShort( 0x8 ), /* 8 */
/* 932 */ 0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 934 */ 0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 936 */ NdrFcShort( 0x4 ), /* 4 */
/* 938 */ NdrFcShort( 0x4 ), /* 4 */
/* 940 */ 0x13, 0x0, /* FC_OP */
/* 942 */ NdrFcShort( 0xffe6 ), /* Offset= -26 (916) */
/* 944 */ 0x5b, /* FC_END */
0x8, /* FC_LONG */
/* 946 */ 0x8, /* FC_LONG */
/* 948 */ 0x5b, /* FC_END */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 950 */ NdrFcShort( 0x4 ), /* 4 */
/* 952 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 954 */ NdrFcShort( 0x0 ), /* 0 */
/* 956 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 958 */ 0x8, /* FC_LONG */
/* 960 */ 0x5b, /* FC_END */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 962 */ NdrFcShort( 0x8 ), /* 8 */
/* 964 */ 0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 966 */ 0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 968 */ NdrFcShort( 0x4 ), /* 4 */
/* 970 */ NdrFcShort( 0x4 ), /* 4 */
/* 972 */ 0x13, 0x0, /* FC_OP */
/* 974 */ NdrFcShort( 0xffe6 ), /* Offset= -26 (948) */
/* 976 */ 0x5b, /* FC_END */
0x8, /* FC_LONG */
/* 978 */ 0x8, /* FC_LONG */
/* 980 */ 0x5b, /* FC_END */
0x1b, /* FC_CARRAY */
0x7, /* 7 */
/* 982 */ NdrFcShort( 0x8 ), /* 8 */
/* 984 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 986 */ NdrFcShort( 0x0 ), /* 0 */
/* 988 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 990 */ 0xb, /* FC_HYPER */
/* 992 */ 0x5b, /* FC_END */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 994 */ NdrFcShort( 0x8 ), /* 8 */
/* 996 */ 0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 998 */ 0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 1000 */ NdrFcShort( 0x4 ), /* 4 */
/* 1002 */ NdrFcShort( 0x4 ), /* 4 */
/* 1004 */ 0x13, 0x0, /* FC_OP */
/* 1006 */ NdrFcShort( 0xffe6 ), /* Offset= -26 (980) */
/* 1008 */ 0x5b, /* FC_END */
0x8, /* FC_LONG */
/* 1010 */ 0x8, /* FC_LONG */
/* 1012 */ 0x5b, /* FC_END */
0x15, /* FC_STRUCT */
0x3, /* 3 */
/* 1014 */ NdrFcShort( 0x8 ), /* 8 */
/* 1016 */ 0x8, /* FC_LONG */
/* 1018 */ 0x8, /* FC_LONG */
/* 1020 */ 0x5b, /* FC_PAD */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 1022 */ NdrFcShort( 0x8 ), /* 8 */
/* 1024 */ 0x7, /* Corr desc: FC_USHORT */

```

```

0x0, /* */
/* 1026 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 1028 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 1030 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 1032 */ NdrFcShort( 0xffec ), /* Offset= -20 (1012) */
/* 1034 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 1036 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 1038 */ NdrFcShort( 0x28 ), /* 40 */
/* 1040 */ NdrFcShort( 0xffec ), /* Offset= -20 (1020) */
/* 1042 */ NdrFcShort( 0x0 ), /* Offset= 0 (1042) */
/* 1044 */ 0x6, /* FC_SHORT */
0x6, /* FC_SHORT */
/* 1046 */ 0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 1048 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 1050 */ NdrFcShort( 0xfc18 ), /* Offset= -1000 (50) */
/* 1052 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 1054 */ 0xb4, /* FC_USER_MARSHAL */
0x83, /* 131 */
/* 1056 */ NdrFcShort( 0x1 ), /* 1 */
/* 1058 */ NdrFcShort( 0x4 ), /* 4 */
/* 1060 */ NdrFcShort( 0x0 ), /* 0 */
/* 1062 */ NdrFcShort( 0xfc04 ), /* Offset= -1020 (42) */
/* 1064 */
0x11, 0x8, /* FC_RP [simple_pointer] */
/* 1066 */ 0x8, /* FC_LONG */
0x5c, /* FC_PAD */
/* 1068 */
0x11, 0x14, /* FC_RP
[allocated_on_stack] [pointer_deref] */
/* 1070 */ NdrFcShort( 0x2 ), /* Offset= 2 (1072) */
/* 1072 */
0x13, 0x0, /* FC_OP */
/* 1074 */ NdrFcShort( 0x2 ), /* Offset= 2 (1076) */
/* 1076 */
0x1b, /* FC_CARRAY */
0x0, /* 0 */
/* 1078 */ NdrFcShort( 0x1 ), /* 1 */
/* 1080 */ 0x28, /* Corr desc: parameter, FC_LONG */
0x54, /*
FC_DEREFERENCE */
/* 1082 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 1084 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 1086 */ 0x2, /* FC_CHAR */
0x5b, /* FC_END */
0x0
}
};
static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
{
{
BSTR_UserSize
,BSTR_UserMarshal
,BSTR_UserUnmarshal
,BSTR_UserFree
},
{
LPSAFEARRAY_UserSize
,LPSAFEARRAY_UserMarshal

```

```

,LPSAFEARRAY_UserUnmarshal
,LPSAFEARRAY_UserFree
}
};
/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */
/* Object interface: IDispatch, ver. 0.0,
GUID={0x00020400,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */
/* Object interface: IComponentRegistrar, ver. 0.0,
GUID={0xa817e7a2,0x43fa,0x11d0,{0x9e,0x44,0x00,0xaa,0x00,0xb6,0x77,0x0a}} */
#pragma code_seg("orpc")
static const unsigned short IComponentRegistrar_FormatStringOffsetTable[] =
{
(unsigned short) -1,
(unsigned short) -1,
(unsigned short) -1,
(unsigned short) -1,
0,
36,
66,
96,
138,
174
};
static const MIDL_STUBLESS_PROXY_INFO
IComponentRegistrar_ProxyInfo =
{
&Object_StubDesc,
_MIDL_ProcFormatString.Format,
&IComponentRegistrar_FormatStringOffsetTable[-3],
0,
0,
0
};
static const MIDL_SERVER_INFO IComponentRegistrar_ServerInfo =
{
&Object_StubDesc,
0,
_MIDL_ProcFormatString.Format,
&IComponentRegistrar_FormatStringOffsetTable[-3],
0,
0,
0,
0,
0
};
CINTERFACE_PROXY_VTABLE(13) _IComponentRegistrarProxyVtbl =
{
&IComponentRegistrar_ProxyInfo,
&IID_IComponentRegistrar,
IUnknown_QueryInterface_Proxy,
IUnknown_AddRef_Proxy,
IUnknown_Release_Proxy ,
0 /* (void *) (INT_PTR) -1 /* IDispatch::GetTypeInfoCount */ ,
0 /* (void *) (INT_PTR) -1 /* IDispatch::TypeInfo */ ,
0 /* (void *) (INT_PTR) -1 /* IDispatch::GetIDsOfNames */ ,
0 /* IDispatch_Invoke_Proxy */ ,
(void *) (INT_PTR) -1 /* IComponentRegistrar::Attach */ ,
(void *) (INT_PTR) -1 /* IComponentRegistrar::RegisterAll */ ,
(void *) (INT_PTR) -1 /* IComponentRegistrar::UnregisterAll */ ,

```

```

(void *) (INT_PTR) -1 /* IComponentRegistrar::GetComponents */,
(void *) (INT_PTR) -1 /* IComponentRegistrar::RegisterComponent */,
(void *) (INT_PTR) -1 /* IComponentRegistrar::UnregisterComponent */
};

static const PRPC_STUB_FUNCTION IComponentRegistrar_table[] =
{
    STUB_FORWARDING_FUNCTION,
    STUB_FORWARDING_FUNCTION,
    STUB_FORWARDING_FUNCTION,
    STUB_FORWARDING_FUNCTION,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2
};

CInterfaceStubVtbl _IComponentRegistrarStubVtbl =
{
    &IID_IComponentRegistrar,
    &IComponentRegistrar_ServerInfo,
    13,
    &IComponentRegistrar_table[-3],
    CStdStubBuffer_DELEGATING_METHODS
};

/* Object interface: Itpcc_com, ver. 0.0,

GUID={0x5B4FA473,0x2E68,0x4D79,{0xA6,0x26,0xF3,0x8B,0x30,0xB8,0x
19,0x6E}} */
#pragma code_seg(".orpc")
static const unsigned short Itpcc_com_FormatStringOffsetTable[] =
{
    210,
    252,
    294,
    336,
    378,
    36
};

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

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

CINTERFACE_PROXY_VTABLE(9) _Itpcc_comProxyVtbl =
{
    &Itpcc_com_ProxyInfo,
    &IID_Itpcc_com,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy,
    (void *) (INT_PTR) -1 /* Itpcc_com::doStockLevel */,
    (void *) (INT_PTR) -1 /* Itpcc_com::doNewOrder */,
    (void *) (INT_PTR) -1 /* Itpcc_com::doPayment */,
    (void *) (INT_PTR) -1 /* Itpcc_com::doOrderStatus */,
    (void *) (INT_PTR) -1 /* Itpcc_com::doDBInfo */,
    (void *) (INT_PTR) -1 /* Itpcc_com::doSetComplete */
};

const CInterfaceStubVtbl _Itpcc_comStubVtbl =
{
    &IID_Itpcc_com,
    &Itpcc_com_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,
    0x6000169, /* MIDL Version 6.0.361 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* cs routines */
    0, /* proxy/server info */
    0 /* Reserved5 */
};

const CInterfaceProxyVtbl * _tpccCom_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &_Itpcc_comProxyVtbl,
    ( CInterfaceProxyVtbl *) &_IComponentRegistrarProxyVtbl,
    0
};

const CInterfaceStubVtbl * _tpccCom_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &_Itpcc_comStubVtbl,
    ( CInterfaceStubVtbl *) &_IComponentRegistrarStubVtbl,
    0
};

PCInterfaceName const _tpccCom_InterfaceNamesList[] =
{
    "Itpcc_com",
    "IComponentRegistrar",
    0
};

const IID * _tpccCom_BaseIIDList[] =
{
    0,
    &IID_IDispatch,
    0
};

#define _tpccCom_CHECK_IID(n) IID_GENERIC_CHECK_IID(
_tpccCom, piID, n)
int __stdcall _tpccCom_IID_Lookup( const IID * piID, int * piIndex )
{

```

```

IID_BS_LOOKUP_SETUP
IID_BS_LOOKUP_INITIAL_TEST( _tpccCom, 2, 1 )
IID_BS_LOOKUP_RETURN_RESULT( _tpccCom, 2, *pIndex )
}
const ExtendedProxyFileInfo tpccCom_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) & _tpccCom_ProxyVtblList,

    (PCInterfaceStubVtblList *) & _tpccCom_StubVtblList,
    (const PCInterfaceName *) & _tpccCom_InterfaceNamesList,
    (const IID **) & _tpccCom_BaseIIDList,
    & _tpccCom_IID_Lookup,
    2,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
};
#endif _MSC_VER >= 1200
#pragma warning(pop)
#endif
#endif /* !defined(_M_IA64) && !defined(_M_AMD64)*/

```

TpccDB2Glue/stdafx.h

```

// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//
#pragma once

#define WIN32_LEAN_AND_MEAN // Exclude rarely-used
stuff from Windows headers
// Windows Header Files:
#include <windows.h>
// TODO: reference additional headers your program requires here

```

tpccDB2Glue/tpccDB2glue.h

```

// The following ifdef block is the standard way of creating macros which make
exporting
// from a DLL simpler. All files within this DLL are compiled with the
TPCCDB2GLUE_EXPORTS
// symbol defined on the command line. this symbol should not be defined on
any project
// that uses this DLL. This way any other project whose source files include this
file see
// TPCCDB2GLUE_API functions as being imported from a DLL, whereas this
DLL sees symbols
// defined with this macro as being exported.
#ifdef TPCCDB2GLUE_EXPORTS
#define TPCCDB2GLUE_API __declspec(dllexport)
#else
#define TPCCDB2GLUE_API __declspec(dllimport)
#endif
#ifndef SPGENERAL
#define SPGENERAL
#endif
#include <db2tpcc.h>
#include <tpcc.h>
// Error/Debug log file defines
ofstream debugStream;

```

```

ofstream errorStream;
CRITICAL_SECTION debugMutex;
CRITICAL_SECTION errorMutex;
//define TIMING 1
FILE *respTimes;
struct txn
{
    short txnType;
    struct _timeb startime;
    struct _timeb endtime;
    short padding;
};
// Registry Values
#define DB_USER_NAME "dbUserName"
#define DB_USER_PASSWORD "dbPassword"
#define DB_NAME "dbName"
char userName[16] = {NULL};
char userPassword[16] = {NULL};
HKEY registryKey;
DWORD regType;
char value[MAX_STRING_LEN];
DWORD regValueSize = MAX_STRING_LEN;
// DB2 Glue Function Prototypes
extern "C" TPCCDB2GLUE_API int connect_db(char *dbName, void **ctx);
extern "C" TPCCDB2GLUE_API int getContext(void **ctx);
extern "C" TPCCDB2GLUE_API int detachContext(void *ctx);
extern "C" TPCCDB2GLUE_API int attachContext(void *ctx);
extern "C" TPCCDB2GLUE_API int disconnect_db(void *ctx);
extern "C" TPCCDB2GLUE_API int do_nord(nord_wrapper *nord, void *ctx);
extern "C" TPCCDB2GLUE_API int do_pymt(pymt_wrapper *pymt, void *ctx);
extern "C" TPCCDB2GLUE_API int do_ord(ords_wrapper *ords, void *ctx);
extern "C" TPCCDB2GLUE_API int do_dlv(dlv_wrapper *dlv, void *ctx);
extern "C" TPCCDB2GLUE_API int do_stok(stok_wrapper *stok, void *ctx);

```

tpccDB2Glue/stdafx.cpp

```

// stdafx.cpp : source file that includes just the standard includes
// tpccDB2glue.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information
#include "stdafx.h"
// TODO: reference any additional headers you need in STDAFX.H
// and not in this file

```

tpccDB2Glue/tpccDB2glue.cpp

```

// tpccDB2glue.cpp : Defines the entry point for the DLL application.
//
#include "stdafx.h"
#include "tpccDB2glue.h"
BOOL WINAPI DllMain( HANDLE hModule,
                    DWORD ul_reason_for_call,
                    LPVOID lpReserved
                    )
{
    switch (ul_reason_for_call)
    {
        case DLL_PROCESS_ATTACH:
            if(debugFlag)
            {

```



```

        InitializeCriticalSection(&debugMutex);

        debugStream.rdbuf(
)->open("C:\\inetpub\\wwwroot\\tpcc\\debug_gluecode.txt",ios_base::in |
ios_base::out | ios_base::app);
        if(!debugStream.rdbuf( ) ->is_open())
            return FALSE;
    }

    DEBUGMSG("Entered dllMain of tpccDB2glue.dll" <<
endl);
    InitializeCriticalSection(&errorMutex);
    errorStream.rdbuf(
)->open("C:\\inetpub\\wwwroot\\tpcc\\error_gluecode.txt",ios_base::in |
ios_base::out | ios_base::app);
    if(!errorStream.rdbuf( )->is_open())
        return FALSE;

#ifdef TIMING
respTimes=fopen("c:\\inetpub\\wwwroot\\tpcc\\respTimes","wb");
    if(!respTimes)
    {
        ERRORMSG("Unable to open response time
file c:\\inetpub\\wwwroot\\tpcc\\respTimes"<<endl);
        return FALSE;
    }
    ERRORMSG("Response time file created:"<<endl);
#endif

    DEBUGMSG("Opening registry sub key "<<
REGISTRY_SUB_KEY << endl);
    //open up registry key

if(RegOpenKeyEx(HKEY_LOCAL_MACHINE,REGISTRY_SUB_KEY,0,K
EY_READ,&registryKey) == ERROR_SUCCESS)
    {
        DEBUGMSG("Registry key open"<<endl);
        //get the null db user name
        regValueSize = sizeof(value);
        if
        (RegQueryValueEx(registryKey,DB_USER_NAME,0,&regType,(BYT
E *) &value,&regValueSize)== ERROR_SUCCESS )
            strcpy(userName,value);
        else
            return
ERR_INVALID_USERNAME;
        DEBUGMSG("DB user name:"<< userName
<< endl);
        regValueSize = sizeof(value);
        if
        (RegQueryValueEx(registryKey,DB_USER_PASSWORD,0,&regType,(BYT
E *) &value,&regValueSize)== ERROR_SUCCESS )
            strcpy(userPassword,value);
        else
            return
ERR_INVALID_PASSWORD;
        DEBUGMSG("DB user
password:"<<userPassword << endl);
    }
    else
    {
        return ERR_INVALID_REGISTRY_KEY;
        DEBUGMSG("Unable to open registry
key"<< REGISTRY_SUB_KEY << endl);
    }
    break;

```

```

        case DLL_THREAD_ATTACH:
            break;
        case DLL_THREAD_DETACH:
            break;
        case DLL_PROCESS_DETACH:
            #ifdef TIMING
                ERRORMSG("dll_process_detach called,
closing timing file"<<endl);
                fclose(respTimes);
            #endif
            break;
    }
    return TRUE;
}
/*
*****
** Name          :          attachContext
** Description   :
**              :          Function calls db2 api
to attach thread to
**              :          a specific context per
thread basis.
** Parameters    :
**              :          void*
stored context
** Returns      :
**              :          int - return code
** Comments     :
**
*****
*/
extern "C" int attachContext(void *ctx)
{
    int rc;
    if ( (rc = attach_context(ctx)) != OK)
        return ERR_ATTACHING_CONTEXT;

    return OK;
}
/*
*****
** Name          :          detachContext
** Description   :
**              :          Function calls db2 api
to detach thread from context
** Parameters    :
**              :          void*
stored context
** Returns      :
**              :          int - return code
** Comments     :
**
*****
*/
extern "C" int detachContext(void *ctx)
{
    int rc;
    if ( (rc = detach_context(ctx)) != OK)
    {
        ERRORMSG("error detaching context from db,
rc:"<<rc<<endl);
        return ERR_DETACHING_CONTEXT;
    }
    return OK;
}
/*
*****
** Name          :          connect_db

```

```

** Description      :
**
** Function calls db2 api
** Parameters      :
**
** char*
**
** dbName
**
** uninitialized context
** Returns        :
**
** int - return code
** Comments       :
**
** To connect to db, first
** connection must be
** established. Next,
** context for that connect
** detach from the
** context just created.
**

```

```

*****
*/
extern "C" TPCCDB2GLUE_API int connect_db(char *dbName,void **ctx)
{
    DEBUGMSG("Entered db2glue do_connect using dbName:"<<
dbName << endl << "Calling connect_to_TM_auth() with username:"<<
userName << " password:" <<userPassword << endl);
    int rc = connect_to_TM_auth(dbName,userName,userPassword);
    if(rc != OK)
    {
        DEBUGMSG("Object do_connect failed,
rc:"<<rc<<endl);
        ERRORMSG("Object do_connect failed,
rc:"<<rc<<endl);
        return rc;
    }
    DEBUGMSG("calling get_context"<<endl);
    if ( (rc = get_context(ctx)) != OK)
    {
        DEBUGMSG("Object get_context() failed, rc:"<< rc
<<endl);
        ERRORMSG("Object get_context() failed, rc:"<< rc
<<endl);
        return ERR_SAVING_CONTEXT;
    }
    DEBUGMSG("Object get_context successful, context:"<<
DEBUGADDRESS(*ctx)<<" saved"<<endl);
    DEBUGMSG("Object calling detach_context() w/
ctx:"<<DEBUGADDRESS(*ctx)<<endl);
    if( (rc = detach_context(*ctx)) != OK)
    {
        DEBUGMSG("Object failed detach_context w/
ctx:"<<DEBUGADDRESS(*ctx)<<" rc:" << rc << endl);
        ERRORMSG("Object failed detach_context w/
ctx:"<<DEBUGADDRESS(*ctx)<<" rc:" << rc << endl);
        return ERR_DETACHING_CONTEXT;
    }
    DEBUGMSG("Object detach_context successful,
context:"<<DEBUGADDRESS(*ctx)<<" , connection complete"<<endl);
    return OK;
}
/*

```

```

*****
** Name           : disconnect_db
** Description    :
**
** Function calls db2 api
** Parameters    :

```

```

**
** void*
** stored context
** Returns      :
**
** int - return code
** Comments     :
**
** To disconnect from db,
** first must attach to
** thread's context. Next,
** disconnect from db

```

```

*****
*/
extern "C" TPCCDB2GLUE_API int disconnect_db(void *ctx)
{
    DEBUGMSG("Entered do_disconnect, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        return ERR_ATTACHING_CONTEXT;
    }
    DEBUGMSG("context established. preparing to call db2" << endl);
    rc = disconnect_from_TM();
    if(rc != OK)
    {
        DEBUGMSG("disconnect failed, rc:"<<rc<<endl);
        ERRORMSG("disconnect failed, rc:"<<rc<<endl);
        return rc;
    }
    return OK;
}
/*

```

```

*****
** Name           : do_nord
** Description    :
**
** Function calls db2 api
** Parameters    :
**
** nord_wrapper*
**
** new order txn structs wrapper
**
** void*
** stored context
** Returns      :
**
** int - return code
** Comments     :
**
** Attach to thread's
** context, call nord sql function
** then detach from
** context.

```

```

*****
*/
extern "C" TPCCDB2GLUE_API int do_nord(nord_wrapper *nord,void *ctx)
{
    DEBUGMSG("Entered do_nord, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("nord failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("nord failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        return ERR_ATTACHING_CONTEXT;
    }
}

```

```

        DEBUGMSG("attached to context:" <<
DEBUGADDRESS(ctx)<<" , preparing to call db2" << endl);
#ifdef TIMING
    struct txn timeSample;
    _ftime(&timeSample.startTime);
#endif

    //call new order txn
    neword_sql(&nord->in_nord,&nord->out_nord);
#ifdef TIMING
    _ftime(&timeSample.endTime);
    timeSample.txnType=1;
    EnterCriticalSection(&errorMutex);
    rc = fwrite(&timeSample,sizeof(struct txn),1,respTimes);

    LeaveCriticalSection(&errorMutex);
#endif
    DEBUGMSG("return from neword_sql(), s_transtatus:" <<
nord->out_nord.s_transtatus << endl);
    rc = detachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("nord failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("nord failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        return ERR_DETACHING_CONTEXT;
    }
    return OK;
}
/*
*****
** Name          :          do_pymt
** Description   :
**              Function calls db2 api
to execute pymt txn
** Parameters    :
**              paym_wrapper*
payment txn structs wrapper
**              void*
stored context
** Returns      :
**              int - return code
** Comments     :
**              Attach to thread's
context, call nord sql function
**              then detach from
context.

*****

*/
extern "C" TPCCDB2GLUE_API int do_pymt(paym_wrapper *pymt,void *ctx)
{
    DEBUGMSG("Entered do_pymt, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("pymt failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("pymt failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        return ERR_ATTACHING_CONTEXT;
    }
    DEBUGMSG("attached to context:"<< DEBUGADDRESS(ctx) <<"
preparing to call db2" << endl);
#ifdef TIMING

```

```

    struct txn timeSample;
    _ftime(&timeSample.startTime);
#endif

    //call pymt txn
    payment_sql(&pymt->in_paym,&pymt->out_paym);
#ifdef TIMING
    _ftime(&timeSample.endTime);
    timeSample.txnType=2;
    EnterCriticalSection(&errorMutex);
    if( (fwrite(&timeSample,sizeof(struct txn),1,respTimes)) != 1 )
    {
        ERRORMSG("Unable to write to binary file,
pymt"<<endl);
    }
    LeaveCriticalSection(&errorMutex);
#endif
    DEBUGMSG("return from payment_sql(), s_transtatus:" <<
pymt->out_paym.s_transtatus << endl);

    rc = detachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("pymt failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<endl);
        DEBUGMSG("pymt failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);

        return ERR_DETACHING_CONTEXT;
    }
    DEBUGMSG("pymt detach_context successful. pymt txn
complete."<<endl);
    return OK;
}
/*
*****
** Name          :          do_ord
** Description   :
**              Function calls db2 api
to execute ord
** Parameters    :
**              ord_wrapper*
order status txn structs wrapper
**              void*
stored context
** Returns      :
**              int - return code
** Comments     :
**              Attach to thread's
context, call nord sql function
**              then detach from
context.

*****

*/
extern "C" TPCCDB2GLUE_API int do_ord(ord_wrapper *ord,void *ctx)
{
    DEBUGMSG("Entered do_ord, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("ord failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("ord failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        return ERR_ATTACHING_CONTEXT;
    }
}

```

```

        DEBUGMSG("attached to context:"<<DEBUGADDRESS(ctx)<<","
preparing to call db2" << endl);
        DEBUGMSG("calling ordstat_sql()" <<endl);
#ifdef TIMING
        struct txn timeSample;
        _ftime(&timeSample.startTime);
#endif
        ordstat_sql(&ords->in_ords,&ords->out_ords);
#ifdef TIMING
        _ftime(&timeSample.endTime);
        timeSample.txnType=3;
        EnterCriticalSection(&errorMutex);
        if( (fwrite(&timeSample,sizeof(struct txn),1,respTimes)) != 1 )
        {
                ERRORMSG("Unable to write to binary file,
ords"<<endl);
        }
        LeaveCriticalSection(&errorMutex);
#endif
        DEBUGMSG("return from ordstat_sql(), s_transtatus:" <<
ords->out_ords.s_transtatus << endl);
        rc = detachContext(ctx);
        if(rc != OK)
        {
                ERRORMSG("ords failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
                DEBUGMSG("ords failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
                return ERR_DETACHING_CONTEXT;
        }
        DEBUGMSG("ords detach_context successful. pymt txn
complete."<<endl);
        return OK;
}
/*
*****
** Name          :          do_dlvvy
** Description   :
**              Function calls db2 api
to execute ords txn
** Parameters   :
**              dlvvy_wrapper*
dlvy txn structs wrapper
**              void*
stored context
** Returns     :
**              int - return code
** Comments    :
**              Attach to thread's
context, call nord sql function
**              then detach from
context.
*****
*/
extern "C" TPCCDB2GLUE_API int do_dlvvy(dlvvy_wrapper *dlvy,void *ctx)
{
        DEBUGMSG("Entered do_dlvvy, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
        int rc = attachContext(ctx);
        if(rc != OK)
        {
                ERRORMSG("dlvy failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
                DEBUGMSG("dlvy failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
                return ERR_ATTACHING_CONTEXT;
        }
}

```

```

        DEBUGMSG("attached to context:"<<DEBUGADDRESS(ctx)<<","
preparing to call db2" << endl);
        DEBUGMSG("calling delivery_sql" << endl);
#ifdef TIMING
        struct txn timeSample;
        _ftime(&timeSample.startTime);
#endif
        //call dlvy txn
        delivery_sql(&dlvy->in_dlvvy,&dlvy->out_dlvvy);
#ifdef TIMING
        _ftime(&timeSample.endTime);
        timeSample.txnType=3;
        EnterCriticalSection(&errorMutex);
        if( (fwrite(&timeSample,sizeof(struct txn),1,respTimes)) != 1 )
        {
                ERRORMSG("Unable to write to binary file,
dlvy"<<endl);
        }
        LeaveCriticalSection(&errorMutex);
#endif
        DEBUGMSG("return from delivery_sql(), s_transtatus:" <<
dlvy->out_dlvvy.s_transtatus << endl);
        rc = detachContext(ctx);
        if(rc != OK)
        {
                ERRORMSG("dlvy failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
                DEBUGMSG("dlvy failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
                return ERR_DETACHING_CONTEXT;
        }
        DEBUGMSG("dlvy detach_context successful. dlvy txn
complete."<<endl);
        return OK;
}
/*
*****
** Name          :          do_stok
** Description   :
**              Function calls db2 api
to execute stok txn
** Parameters   :
**              stok_wrapper*
stock txn structs wrapper
**              void*
stored context
** Returns     :
**              int - return code
** Comments    :
**              Attach to thread's
context, call nord sql function
**              then detach from
context.
*****
*/
extern "C" TPCCDB2GLUE_API int do_stok(stok_wrapper *stok,void *ctx)
{
        DEBUGMSG("Entered do_stok, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
        int rc = attachContext(ctx);
        if(rc != OK)
        {
                ERRORMSG("stok failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
                DEBUGMSG("stok failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
                return ERR_ATTACHING_CONTEXT;
        }
}

```

```

        DEBUGMSG("attaching to context:"<<DEBUGADDRESS(ctx)<<","
preparing to call db2" << endl);
        DEBUGMSG("calling stocklev_sql()" <<endl);
#ifdef TIMING
        struct txn timeSample;
        _ftime(&timeSample.startTime);
#endif
        //call stock level txn
        stocklev_sql(&stok->in_stok, &stok->out_stok);
#ifdef TIMING
        _ftime(&timeSample.endTime);
        timeSample.txnType=5;
        EnterCriticalSection(&errorMutex);
        if( (fwrite(&timeSample,sizeof(struct txn),1,respTimes)) != 1 )
        {
                ERRORMSG("Unable to write to binary file,
stok"<<endl);
        }
        LeaveCriticalSection(&errorMutex);
#endif
        DEBUGMSG("return from stocklev_sql(), s_transtatus:" <<
stok->out_stok.s_transtatus << endl);
        DEBUGMSG("calling detach_context"<<endl);
        rc = detachContext(ctx);
        if(rc != OK)
        {
                ERRORMSG("stok failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
                DEBUGMSG("stok failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
                return ERR_DETACHING_CONTEXT;
        }
        DEBUGMSG("detach_context successful. stok txn
complete."<<endl);
        return OK;
}
??
??
??
??

TPC Benchmark™ C Full Disclosure Report - IBM eServer p5 595 Model
9119-595 Page 121 of 704

```

NullDB.cpp

```

// nullDB.cpp : Defines the entry point for the DLL application.
//

```

```

#include "stdafx.h"
#include "nullDB.h"
#include "..\tpcc\api\tpcc.h"

```

```

BOOL APIENTRY DllMain( HANDLE hModule,
        DWORD ul_reason_for_call,
        LPVOID lpReserved
        )
{
        switch (ul_reason_for_call)
        {
        case DLL_PROCESS_ATTACH:
        case DLL_THREAD_ATTACH:
        case DLL_THREAD_DETACH:
        case DLL_PROCESS_DETACH:

```

```

        break;
        }
        return TRUE;
}
// This is an example of an exported variable
NULLDB_API int dataSet = 0;
extern "C" NULLDB_API int connect_db(char *dbName,void **ctx)
{
        return OK;
}
extern "C" NULLDB_API int disconnect_db(void *ctx)
{
        return OK;
}
extern "C" NULLDB_API int do_nord(struct nord_wrapper *nord,void *ctx)
{
        nord->out_nord.s_transtatus = 0;

        if (dataSet == 0)
        {
                strcpy(nord->out_nord.s_C_LAST,"NOYOLA");
                strcpy(nord->out_nord.s_C_CREDIT,"GC");
                nord->out_nord.s_W_TAX = 1694;
                nord->out_nord.s_D_TAX = 967;
                nord->out_nord.s_C_DISCOUNT = 1024;
                nord->out_nord.s_O_ID = 3013;
                nord->out_nord.s_O_OL_CNT = 4;
                nord->out_nord.s_total_amount = 32345;
                nord->out_nord.s_O_ENTRY_D_time = 1234567890;

                strcpy(nord->out_nord.item[0].s_I_NAME,"98 Toyota
Supra Turbo");
                nord->in_nord.in_item[0].s_OL_I_ID = 1;
                nord->in_nord.in_item[0].s_OL_QUANTITY = 1;
                nord->in_nord.in_item[0].s_OL_SUPPLY_W_ID = 1;
                nord->out_nord.item[0].s_I_PRICE = 42000;
                nord->out_nord.item[0].s_OL_AMOUNT = 554000;
                nord->out_nord.item[0].s_S_QUANTITY = 31;
                nord->out_nord.item[0].s_brand_generic = 'G';

                strcpy(nord->out_nord.item[1].s_I_NAME,"HKS Turbo
Timer");
                nord->in_nord.in_item[1].s_OL_I_ID = 1;
                nord->in_nord.in_item[1].s_OL_QUANTITY = 1;
                nord->in_nord.in_item[1].s_OL_SUPPLY_W_ID = 1;
                nord->out_nord.item[1].s_I_PRICE = 4500;
                nord->out_nord.item[1].s_OL_AMOUNT = 438100;
                nord->out_nord.item[1].s_S_QUANTITY = 57;
                nord->out_nord.item[1].s_brand_generic = 'G';

                strcpy(nord->out_nord.item[2].s_I_NAME,"TRD GEN2
Exhaust");
                nord->in_nord.in_item[2].s_OL_I_ID = 1;
                nord->in_nord.in_item[2].s_OL_QUANTITY = 1;
                nord->in_nord.in_item[2].s_OL_SUPPLY_W_ID = 1;
                nord->out_nord.item[2].s_I_PRICE = 6734;
                nord->out_nord.item[2].s_OL_AMOUNT = 47173;
                nord->out_nord.item[2].s_S_QUANTITY = 42;
                nord->out_nord.item[2].s_brand_generic = 'G';

                strcpy(nord->out_nord.item[3].s_I_NAME,"BLITZ
DUAL-SOLENOID");
                nord->in_nord.in_item[3].s_OL_I_ID = 1;

```

```

nord->in_nord.in_item[3].s_OL_QUANTITY = 1;
nord->in_nord.in_item[3].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[3].s_I_PRICE = 35000;
nord->out_nord.item[3].s_OL_AMOUNT = 12096;
nord->out_nord.item[3].s_S_QUANTITY = 84;
nord->out_nord.item[3].s_brand_generic = 'G';

}
else
{
    dataSet = 1;

    strcpy(nord->out_nord.s_C_LAST,"SIMPSON");
    strcpy(nord->out_nord.s_C_CREDIT,"GC");
    nord->out_nord.s_W_TAX = 913;
    nord->out_nord.s_D_TAX = 1519;
    nord->out_nord.s_C_DISCOUNT = 958;
    nord->out_nord.s_O_ID = 1410;
    nord->out_nord.s_O_OL_CNT = 9;
    nord->out_nord.s_total_amount = 12345;
    nord->out_nord.s_O_ENTRY_D_time = 1234567890;

    strcpy(nord->out_nord.item[0].s_I_NAME,"97 Toyota

Supra NA");

    nord->in_nord.in_item[0].s_OL_I_ID = 1;
    nord->in_nord.in_item[0].s_OL_QUANTITY = 1;
    nord->in_nord.in_item[0].s_OL_SUPPLY_W_ID = 1;
    nord->out_nord.item[0].s_I_PRICE = 30000;
    nord->out_nord.item[0].s_OL_AMOUNT = 769600;
    nord->out_nord.item[0].s_S_QUANTITY = 97;
    nord->out_nord.item[0].s_brand_generic = 'G';

    strcpy(nord->out_nord.item[1].s_I_NAME,"98 Turbo

Stereo");

    nord->in_nord.in_item[1].s_OL_I_ID = 1;
    nord->in_nord.in_item[1].s_OL_QUANTITY = 1;
    nord->in_nord.in_item[1].s_OL_SUPPLY_W_ID = 1;
    nord->out_nord.item[1].s_I_PRICE = 10001;
    nord->out_nord.item[1].s_OL_AMOUNT = 192999;
    nord->out_nord.item[1].s_S_QUANTITY = 51;
    nord->out_nord.item[1].s_brand_generic = 'G';

    strcpy(nord->out_nord.item[2].s_I_NAME,"XERD

Exhaust Header");

    nord->in_nord.in_item[2].s_OL_I_ID = 1;
    nord->in_nord.in_item[2].s_OL_QUANTITY = 1;
    nord->in_nord.in_item[2].s_OL_SUPPLY_W_ID = 1;
    nord->out_nord.item[2].s_I_PRICE = 4000;
    nord->out_nord.item[2].s_OL_AMOUNT = 41670;
    nord->out_nord.item[2].s_S_QUANTITY = 14;
    nord->out_nord.item[2].s_brand_generic = 'G';

    strcpy(nord->out_nord.item[3].s_I_NAME,"LEXOL

Conditioner");

    nord->in_nord.in_item[3].s_OL_I_ID = 1;
    nord->in_nord.in_item[3].s_OL_QUANTITY = 1;
    nord->in_nord.in_item[3].s_OL_SUPPLY_W_ID = 1;
    nord->out_nord.item[3].s_I_PRICE = 1400;
    nord->out_nord.item[3].s_OL_AMOUNT = 17213;
    nord->out_nord.item[3].s_S_QUANTITY = 90;
    nord->out_nord.item[3].s_brand_generic = 'G';

    strcpy(nord->out_nord.item[4].s_I_NAME,"TRD Sticker

1");

    nord->in_nord.in_item[4].s_OL_I_ID = 1;
    nord->in_nord.in_item[4].s_OL_QUANTITY = 1;
    nord->in_nord.in_item[4].s_OL_SUPPLY_W_ID = 1;
    nord->out_nord.item[4].s_I_PRICE = 1400;
    nord->out_nord.item[4].s_OL_AMOUNT = 27232;

    nord->out_nord.item[4].s_S_QUANTITY = 75;
    nord->out_nord.item[4].s_brand_generic = 'G';

    strcpy(nord->out_nord.item[5].s_I_NAME,"TRD Sticker

2");

    nord->in_nord.in_item[5].s_OL_I_ID = 1;
    nord->in_nord.in_item[5].s_OL_QUANTITY = 1;
    nord->in_nord.in_item[5].s_OL_SUPPLY_W_ID = 1;
    nord->out_nord.item[5].s_I_PRICE = 4400;
    nord->out_nord.item[5].s_OL_AMOUNT = 35808;
    nord->out_nord.item[5].s_S_QUANTITY = 22;
    nord->out_nord.item[5].s_brand_generic = 'G';

    strcpy(nord->out_nord.item[6].s_I_NAME,"TRD Sticker

3");

    nord->in_nord.in_item[6].s_OL_I_ID = 1;
    nord->in_nord.in_item[6].s_OL_QUANTITY = 1;
    nord->in_nord.in_item[6].s_OL_SUPPLY_W_ID = 1;
    nord->out_nord.item[6].s_I_PRICE = 5500;
    nord->out_nord.item[6].s_OL_AMOUNT = 44392;
    nord->out_nord.item[6].s_S_QUANTITY = 21;
    nord->out_nord.item[6].s_brand_generic = 'G';

    strcpy(nord->out_nord.item[7].s_I_NAME,"TRD Sticker

4");

    nord->in_nord.in_item[7].s_OL_I_ID = 1;
    nord->in_nord.in_item[7].s_OL_QUANTITY = 1;
    nord->in_nord.in_item[7].s_OL_SUPPLY_W_ID = 1;
    nord->out_nord.item[7].s_I_PRICE = 8300;
    nord->out_nord.item[7].s_OL_AMOUNT = 83410;
    nord->out_nord.item[7].s_S_QUANTITY = 35;
    nord->out_nord.item[7].s_brand_generic = 'G';

    strcpy(nord->out_nord.item[8].s_I_NAME,"98 Toyota

OEM Bra");

    nord->in_nord.in_item[8].s_OL_I_ID = 1;
    nord->in_nord.in_item[8].s_OL_QUANTITY = 1;
    nord->in_nord.in_item[8].s_OL_SUPPLY_W_ID = 1;
    nord->out_nord.item[8].s_I_PRICE = 10000;
    nord->out_nord.item[8].s_OL_AMOUNT = 43160;
    nord->out_nord.item[8].s_S_QUANTITY = 73;
    nord->out_nord.item[8].s_brand_generic = 'G';

    dataSet = 0;
}
return OK;
}
extern "C" NULLDB_API int do_pymt(struct paym_wrapper *pymt,void *ctx)
{
    pymt->out_paym.s_transtatus = 0;

    if (dataSet == 0)
    {
        pymt->out_paym.s_C_CREDIT_LIM = 5000000;
        pymt->out_paym.s_C_DISCOUNT = 1024;
        pymt->out_paym.s_C_BALANCE = 17815;
        pymt->out_paym.s_C_ID = 89;
        pymt->out_paym.s_H_DATE_time = 1234567890;

        strcpy(pymt->out_paym.s_W_STREET_1,"11501 Burnet

Rd");

        strcpy(pymt->out_paym.s_W_STREET_2,"BLD 905");
        strcpy(pymt->out_paym.s_W_CITY,"Austin");
        strcpy(pymt->out_paym.s_W_STATE,"TX");
        strcpy(pymt->out_paym.s_W_ZIP,"78758");
    }
}

```



```

strcpy(ords->out_ords.s_C_MIDDLE,"J");
strcpy(ords->out_ords.s_C_LAST,"Forman");
ords->out_ords.s_O_ENTRY_D_time = 1234567890;

ords->out_ords.item[0].s_OL_AMOUNT = 12000;
ords->out_ords.item[0].s_OL_I_ID = 54602;
ords->out_ords.item[0].s_OL_SUPPLY_W_ID = 10;
ords->out_ords.item[0].s_OL_QUANTITY = 5;
ords->out_ords.item[0].s_OL_DELIVERY_D_time =
1234567890;

ords->out_ords.item[1].s_OL_AMOUNT = 2300;
ords->out_ords.item[1].s_OL_I_ID = 18860;
ords->out_ords.item[1].s_OL_SUPPLY_W_ID = 10;
ords->out_ords.item[1].s_OL_QUANTITY = 5;
ords->out_ords.item[1].s_OL_DELIVERY_D_time =
1234567890;

ords->out_ords.item[2].s_OL_AMOUNT = 56009;
ords->out_ords.item[2].s_OL_I_ID = 90488;
ords->out_ords.item[2].s_OL_SUPPLY_W_ID = 10;
ords->out_ords.item[2].s_OL_QUANTITY = 5;
ords->out_ords.item[2].s_OL_DELIVERY_D_time =
1234567890;

ords->out_ords.item[3].s_OL_AMOUNT = 98000;
ords->out_ords.item[3].s_OL_I_ID = 22741;
ords->out_ords.item[3].s_OL_SUPPLY_W_ID = 10;
ords->out_ords.item[3].s_OL_QUANTITY = 5;
ords->out_ords.item[3].s_OL_DELIVERY_D_time =
1234567890;

ords->out_ords.item[4].s_OL_AMOUNT = 25000;
ords->out_ords.item[4].s_OL_I_ID = 92952;
ords->out_ords.item[4].s_OL_SUPPLY_W_ID = 10;
ords->out_ords.item[4].s_OL_QUANTITY = 5;
ords->out_ords.item[4].s_OL_DELIVERY_D_time =
1234567890;

    dataSet = 0;
}
return OK;
}
extern "C" NULLDB_API int do_dlvly(struct dlvly_wrapper *dlvly,void *ctx)
{
    dlvly->out_dlvly.s_transtatus = 0;

    if (dataSet == 0)
    {
        dataSet = 1;

        for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
            dlvly->out_dlvly.s_O_ID[districtIndex]= 2055;
    }
    else
    {
        for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
            dlvly->out_dlvly.s_O_ID[districtIndex]= 2056;

        dataSet = 0;
    }
    return OK;
}

```

```

extern "C" NULLDB_API int do_stok(struct stok_wrapper *stok,void *ctx)
{
    stok->out_stok.s_transtatus = 0;

    if (dataSet == 0)
    {
        stok->out_stok.s_low_stock = 100;

        dataSet = 1;
    }
    else
    {
        stok->out_stok.s_low_stock = 40;

        dataSet = 0;
    }
    return OK;
}

```

NullDB.h

```

// The following ifdef block is the standard way of creating macros which make
exporting
// from a DLL simpler. All files within this DLL are compiled with the
NULLDB_EXPORTS
// symbol defined on the command line. this symbol should not be defined on
any project
// that uses this DLL. This way any other project whose source files include this
file see
// NULLDB_API functions as being imported from a DLL, whereas this DLL
sees symbols
// defined with this macro as being exported.
#ifdef NULLDB_EXPORTS
#define NULLDB_API __declspec(dllexport)
#else
#define NULLDB_API __declspec(dllimport)
#endif

extern NULLDB_API int dataSet;

extern "C" NULLDB_API int do_nord(struct nord_wrapper *nord,void *ctx);
extern "C" NULLDB_API int do_pymt(struct pymt_wrapper *pymt,void *ctx);
extern "C" NULLDB_API int do_ords(struct ords_wrapper *ords,void *ctx);
extern "C" NULLDB_API int do_dlvly(struct dlvly_wrapper *dlvly,void *ctx);
extern "C" NULLDB_API int do_stok(struct stok_wrapper *stok,void *ctx);

extern "C" NULLDB_API int connect_db(char *dbName,void **ctx);
extern "C" NULLDB_API int disconnect_db(void *ctx);

```

Stdafx.cpp

```

// stdafx.cpp : source file that includes just the standard includes
// tpccsapi.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"

// TODO: reference any additional headers you need in STDAFX.H
// and not in this file

```

Stdafx.h


```

// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//

#pragma once

#define WIN32_LEAN_AND_MEAN           // Exclude rarely-used
stuff from Windows headers

#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS // some
CString constructors will be explicit

// turns off ATL's hiding of some common and often safely ignored warning
messages
#define _ATL_ALL_WARNINGS

// critical error descriptions will only be shown to the user
// in debug builds. they will always be logged to the event log
#ifdef _DEBUG
#define ATL_CRITICAL_ISAPI_ERROR_LOGONLY
#endif

#ifdef _WIN32_WINNT
#define _WIN32_WINNT 0x0403
#endif

// TODO: this disables support for registering COM objects
// exported by this project since the project contains no
// COM objects or typelib. If you wish to export COM objects
// from this project, add a typelib and remove this line
#define _ATL_NO_COM_SUPPORT

```

```

#include "resource.h"
#include <atlsrvres.h>
#include <atlisapi.h>
#include <atlstencil.h>

```

// TODO: reference additional headers your program requires here

Stdafx.cpp

```

// stdafx.cpp : source file that includes just the standard includes
// tpccComClient.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information

```

```

#include "stdafx.h"

```

// TODO: reference any additional headers you need in STDAFX.H
// and not in this file

Stdafx.h

```

// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//

```

```

#pragma once
#include <iostream>
#include <tchar.h>

```

// TODO: reference additional headers your program requires here

TpccComClient.cpp

```

// tpccComClient.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"

#include "..\tpccCom\tpccCom.h"
#include "..\tpccCom\tpccCom_i.c"
#include <tpcc.h>

struct txn_buffer
{
    char    *dataBuffer;
    int     size;
};

int _tmain(int argc, _TCHAR* argv[])
{
    HRESULT hres;
    Itpcc_com * pTxn;

    hres = CoInitialize(NULL);
    if (FAILED(hres))
    {
        printf("Error : CoInitialize() failed
rc:%d\n",GetLastError());
        fflush(stdout);
        return 0;
    }

    hres =
CoCreateInstance(CLSID_tpcc_com,NULL,CLSCTX_SERVER,IID_Itpcc_co
m,(void **)&pTxn);
    if (FAILED(hres))
    {
        printf("Error : CoCreateInstance() failed rc:%d
hres:%X\n",GetLastError(),hres);
        fflush(stdout);
        return 0;
    }

    //int size = sizeof(in_stocklev_struct);
    //int size2 = sizeof(out_stocklev_struct);

    //define txn buffer to store txn structure in
    struct txn_buffer    comBuffer;
    comBuffer.dataBuffer = (char *)
CoTaskMemAlloc(sizeof(STOCKLEVELDATA));
    if (!(comBuffer.dataBuffer))
    {
        printf(comBuffer.dataBuffer,"CoTaskMemAlloc failed,
rc:%d\n",GetLastError());
        return(TRUE);
    };
    comBuffer.size =
sizeof(STOCKLEVELDATA);

    struct STOCKLEVELDATA *pStock;
    pStock = (STOCKLEVELDATA*)comBuffer.dataBuffer;
    ZeroMemory(pStock,comBuffer.size);

    //initialize fields
    pStock->in_s_W_ID = 10; pStock->in_s_D_ID = 1;
    pStock->in_s_threshold = 2; pStock->out_s_transtatus = -1;

    int dataLen = comBuffer.size;

```

```

    try{
        hres = pTxn->doStockLevel(&dataLen,(unsigned
char*)&comBuffer.dataBuffer);
    }
    catch(...)
    {
        printf("Error : StockLevel() com caused exeception failed
rc:%d\n",GetLastError());
        fflush(stdout);
        return 0;
    }
    if (FAILED(hres))
    {
        printf("Error : StockLevel() com call failed
rc:%d\n",GetLastError());
        return 0;
    }

    pStock = (STOCKLEVELDATA *)comBuffer.dataBuffer;

    printf("Stock Level txn complete.
s_transtatus:%d\n",pStock->out_s_transtatus);

    return 0;
    return 0;
}

```

HtmlPhraser.cpp

```

////////////////////////////////////
// htmlPhraser.cpp
////////////////////////////////////
// Class implmentation of htmlPhraser.
// This class will take a query string and break it into a series
// of consuitant parts
////////////////////////////////////

#include "htmlPhraser.h"

////////////////////////////////////
// htmlPhraser::htmlPhraser
////////////////////////////////////
// Title      : Constructor
// Parameters  : char * query string
// Return Value : None
// Comments   :
////////////////////////////////////

htmlPhraser::htmlPhraser(char *queryString)
{
    // initialize query values
    iCustomerIdFlag = iCarrierNumFlag = iStockThresholdFlag = false;

    // this initializes the query list to NULL's. This means that
    // characters being added are overwriting null characters and
    // therefore the string will be null terminated implicitly.

    memset(iQueryValues,NULL,(MAX_FIELD_NUM *
MAX_FIELD_LEN));

    // controls
    char          queryChar          = NULL;

    int           queryIndex          = -1;
    int           valueIndex          = -1;

```

```

// process each character of query string
while(*queryString)
{
    // check for special case characters
    if(queryChar)
    {
        // a percentage sign would indicate a token
        if(*queryString != '%')
        {
            // a plus sign represents a space
            if(*queryString == '+')
            {
                queryChar = ' ';
                *queryString++;
            }
            else queryChar = *queryString++;
        }
        else queryChar =
convertQueryToken(&queryString);
    }
    else queryChar = '&';

    // handle query reference (&)
    if(queryChar == '&')
    {
        // reset value index
        valueIndex = -1;

        // do we have a numeric query reference
        if(*queryString >= '0' && *queryString <=
'9')
        {
            // numeric query id
            queryIndex =
(((*queryString - '0') *
10) + (*(queryString + 1) - '0'));

            // walk past the two command
            characters
            queryString += 2;

            // validate query value
            if(queryIndex >
MAX_QUERY_ID)
            queryIndex = -1;
        }
        else queryIndex = -1;

        // finished processing for query reference
        continue;
    }

    // we have a query reference but need to wait until we see
    // before accepting value
    if(valueIndex == -1)
    {
        // we are waiting for '='
        if(queryChar == '=')
        {
            valueIndex = 0;

            // set query string flags
            switch(queryIndex)
            {
                case C_ID:

```

```

                                iCustomerIdFlag = true;
break;
                                case CARRIER_NUM:
                                iCarrierNumFlag =
true; break;
                                case STK_THRESHOLD:
                                iStockThresholdFlag =
true; break;
                                default: break;
                                }
                                // finishes looging for '='
                                continue;
                                }
                                // add each character to the query value
                                if(queryIndex > -1 && valueIndex > -1)
                                {
                                    // we are processing a query value
                                    if(valueIndex < MAX_FIELD_LEN)
                                    {
                                        // we have not exceeded max line
len
                                iQueryValues[queryIndex][valueIndex++] = queryChar;
                                    }
                                    continue;
                                }
                                }
                                return;
                                }
////////////////////////////////////
// htmlPhraser::getCommandId
////////////////////////////////////
// Title   : Returns the page command
// Parameters   : None
// Return Value   : int - page command
// Comments   :
////////////////////////////////////

int htmlPhraser::getCommandId()
{
    // return command numeric code
    switch(*iQueryValues[COMMAND_ID])
    {
        case NEW_ORDER_CODE:
            if(iCustomerIdFlag)
                return
COMMAND_NEW_ORDER_RESULTS;
            else return COMMAND_NEW_ORDER;
        case PAYMENT_CODE:
            if(iCustomerIdFlag)
                return COMMAND_PAYMENT_RESULTS;
            else return COMMAND_PAYMENT;
        case ORDER_STATUS_CODE:
            if(iCustomerIdFlag)
                return
COMMAND_ORDER_STATUS_RESULTS;
            else return COMMAND_ORDER_STATUS;
        case DELIVERY_CODE:
            if(iCarrierNumFlag)
                return COMMAND_DELIVERY_RESULTS;
            else return COMMAND_DELIVERY;
        case STOCK_CODE:
            if(iStockThresholdFlag)
                return COMMAND_STOCK_RESULTS;
            else return COMMAND_STOCK;
        case MENU_CODE:
            return COMMAND_LOGIN_RESULTS;
        case EXIT_CODE:
            return COMMAND_EXIT;
        default:
            return COMMAND_LOGIN;
    };
}

// should not get here
return COMMAND_LOGIN;
}

////////////////////////////////////
// htmlPhraser::validate
////////////////////////////////////
// Title   : validate url parameter list for all txn types
// Parameters   : int - txn type
// Return Value   : int - error code
// Comments   :
////////////////////////////////////

int validate(int txnType)
{
    return 0;
}

////////////////////////////////////
// htmlPhraser::convertQueryToken
////////////////////////////////////
// Title   : Returns the page command
// Parameters   : None
// Return Value   : int - page command
// Comments   :
////////////////////////////////////

char htmlPhraser::convertQueryToken(char **queryString)
{
    char queryChar = NULL;

    // skip over %
    (*queryString)++;

    // look at first character
    switch(**queryString)
    {
        case '2':
            {
                // what follows?
                (*queryString)++;

                switch(**queryString)
                {
                    case '1':
                        queryChar = '!';
                        break;
                    case '3':
                        queryChar = '#';
                        break;
                    case '4':
                        queryChar = '$';
                        break;
                    case '5':
                        queryChar = '%';
                        break;
                }
            }
    }
}

```

```

        case '6':
            queryChar = '&';
            break;
        case '8':
            queryChar = '(';
            break;
        case '9':
            queryChar = ')';
            break;
        case 'B':
            queryChar = '+';
            break;
        case 'C':
            queryChar = ',';
            break;
        case 'F':
            queryChar = '/';
            break;
        case '!':
            queryChar = ' ';
            break;
    }
}
break;
case '3':
{
    // what follows?
    (*queryString)++;

    switch(**queryString)
    {
        case 'A':
            queryChar = '!';
            break;
        case 'B':
            queryChar = ',';
            break;
        case 'D':
            queryChar = '=';
            break;
        case 'F':
            queryChar = '?';
            break;
        case '!':
            queryChar = ' ';
            break;
    }
}
break;
case '4':
{
    // what follows?
    (*queryString)++;

    switch(**queryString)
    {
        case '0':
            queryChar = '@';
            break;
        case '!':
            queryChar = ' ';
            break;
    }
}
break;

```

```

        case '5':
        {
            // what follows?
            (*queryString)++;

            switch(**queryString)
            {
                case 'B':
                    queryChar = '[';
                    break;
                case 'D':
                    queryChar = ']';
                    break;
                case 'E':
                    queryChar = '^';
                    break;
                case '!':
                    queryChar = ' ';
                    break;
            }
        }
        break;
        case '7':
        {
            // what follows?
            (*queryString)++;

            switch(**queryString)
            {
                case 'B':
                    queryChar = '{';
                    break;
                case 'C':
                    queryChar = '|';
                    break;
                case 'D':
                    queryChar = '}';
                    break;
                case 'E':
                    queryChar = '~';
                    break;
                case '!':
                    queryChar = ' ';
                    break;
            }
        }
        break;
        case '+':
            queryChar = '+';
            break;
    }

    // advance pointer and return
    (*queryString)++; return queryChar;
}

```

```

////////////////////////////////////

```

HtmlPhraser.h

```

////////////////////////////////////
// htmlPhraser.h
////////////////////////////////////
// Class to decode a html query string

```

```

////////////////////////////////////
#pragma once
#include <memory.h>

////////////////////////////////////
// Definitions
////////////////////////////////////

#define NULL 0

#define COMMAND_ID 0
#define TERM_ID 1
#define W_ID 2
#define D_ID 3
#define C_ID 4
#define C_NAME 5

#define C_W_ID 6
#define C_D_ID 7
#define AMT_PAID 8
#define STK_THRESHOLD 9
#define CARRIER_NUM 10

#define ITEM_LIST_START 11
#define ITEM_LIST_FINISH 55

#define MAX_QUERY_ID 55
#define MAX_FIELD_LEN 256
#define MAX_FIELD_NUM 56

////////////////////////////////////
// Command Codes
////////////////////////////////////

#define NEW_ORDER_CODE 'n'
#define PAYMENT_CODE 'p'
#define ORDER_STATUS_CODE 'o'
#define DELIVERY_CODE 'd'
#define STOCK_CODE 's'
#define EXIT_CODE 'e'
#define MENU_CODE 'm'

#define COMMAND_LOGIN 0
#define COMMAND_NEW_ORDER 1
#define COMMAND_PAYMENT 2
#define COMMAND_ORDER_STATUS 3
#define COMMAND_DELIVERY 4
#define COMMAND_STOCK 5
#define COMMAND_EXIT 6

#define COMMAND_LOGIN_RESULTS 7
#define COMMAND_NEW_ORDER_RESULTS 8
#define COMMAND_PAYMENT_RESULTS 9

#define COMMAND_ORDER_STATUS_RESULTS 10
#define COMMAND_DELIVERY_RESULTS 11
#define COMMAND_STOCK_RESULTS 12

////////////////////////////////////
// Class htmlPhraser
////////////////////////////////////

class htmlPhraser
{
    // Constructors / Destructor
public:
    htmlPhraser(char *queryString);
    ~htmlPhraser()
    {return;}

    // getters
public:
    int getCommandId();
    int validate(int txnType);

    char * get_TERM_ID()
    {return iQueryValues[TERM_ID];}
    char * get_W_ID()
    {return iQueryValues[W_ID];}
    char * get_D_ID()
    {return iQueryValues[D_ID];}
    char * get_C_ID()
    {return iQueryValues[C_ID];}
    char * get_C_NAME()
    {return iQueryValues[C_NAME];}
    char * get_C_W_ID()
    {return iQueryValues[C_W_ID];}
    char * get_C_D_ID()
    {return iQueryValues[C_D_ID];}
    char * get_AMT_PAID()
    {return iQueryValues[AMT_PAID];}
    char * get_STK_THRESHOLD()
    {return iQueryValues[STK_THRESHOLD];}
    char * get_CARRIER_NUM()
    {return iQueryValues[CARRIER_NUM];}

    char * get_ITEM_SUPP_W(int item) {return
iQueryValues[(ITEM_LIST_START + 0) + (item * 3)];}
    char * get_ITEM_ITEM_NUM(int item)
    {return iQueryValues[(ITEM_LIST_START + 1) + (item * 3)];}
    char * get_ITEM_QTY(int item)
    {return iQueryValues[(ITEM_LIST_START + 2) + (item * 3)];}

    // Class Functions
private:
    char convertQueryToken(char **queryString);

    // Class Attributes
private:
    int iCustomerIdFlag;
    int iCarrierNumFlag;
    int iStockThresholdFlag;

    char
iQueryValues[MAX_FIELD_NUM][MAX_FIELD_LEN];
};

////////////////////////////////////

```

Resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by tpccsapi.rc
//
#define IDS_PROJNAME 100
```

```
// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifdef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 201
#define _APS_NEXT_COMMAND_VALUE 32768
#define _APS_NEXT_CONTROL_VALUE 201
#define _APS_NEXT_SYMED_VALUE 101
#endif
#endif
```

StdAfx.cpp

```
// stdafx.cpp : source file that includes just the standard includes
// tpccsapi.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information
```

```
#include "stdafx.h"
```

```
// TODO: reference any additional headers you need in STDAFX.H
// and not in this file
```

StdAfx.h

```
// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//
```

```
#pragma once
```

```
#define WIN32_LEAN_AND_MEAN // Exclude rarely-used stuff from Windows headers
```

```
#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS // some CString constructors will be explicit
```

```
// turns off ATL's hiding of some common and often safely ignored warning messages
#define _ATL_ALL_WARNINGS
```

```
// critical error descriptions will only be shown to the user
// in debug builds. they will always be logged to the event log
#ifdef _DEBUG
#define ATL_CRITICAL_ISAPI_ERROR_LOGONLY
#endif
```

```
#ifndef WIN32_WINNT
#define WIN32_WINNT 0x0403
#endif
```

```
// TODO: this disables support for registering COM objects
// exported by this project since the project contains no
```

```
// COM objects or typelib. If you wish to export COM objects
// from this project, add a typelib and remove this line
#define _ATL_NO_COM_SUPPORT
```

```
#include "resource.h"
#include <atlsrvres.h>
#include <atlisapi.h>
#include <atlstencil.h>
```

```
// TODO: reference additional headers your program requires here
```

Tpcc.h

```
// Common defines and structures use internally by client code
// Not to be confused with structures actually passed in transactions
//
```

```
// standard includes
```

```
#ifndef _COMMON_TPCC
#define _COMMON_TPCC
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/timeb.h>
#include <time.h>
```

```
#include <db2tpcc.h>
#include <iostream>
#include <fstream>
#include <process.h>
#include <ios>
```

```
////////////////////////////////////
// Defines
////////////////////////////////////
```

```
#define OK
0
#define INVALID_STATUS -1
#define ERR -1
#define INVALID_COM_STATUS -2
```

```
#define TXN_MAX_COMMANDS 55
#define MAX_TRANSACTIONS 14
#define MAX_CMD_LENGTH 100
#define INPUT_ITEMS 3
#define MAX_INT_BUFFER 15
#define NORD_ITEMS 15
#define ITEM_START 11
#define ITEM_END 55
#define MAX_ITEMS 15
```

```
#define MAX_STRING_LEN 256
#define MAX_HTML_PAGE_LEN 4096
#define MAX_HTML_HEADER_LEN 512
```

```
#define DELIVERY_THREADS_NUM 100
```

```
#define DISTRICTS_PER_WAREHOUSE 10
////////////////////////////////////
```

```

// Transaction Codes
///////////////////////////////////////////////////////////////////
#define          ITEM01_SUPP_W          "11"
#define          ITEM01_ITEM_NUM
"12"
#define          ITEM01_OTY
1 "13"

#define          TXN_LOGIN
0
#define          TXN_NEW_ORDER          1
#define          TXN_PAYMENT
2
#define          TXN_ORDER_STATUS      3
#define          TXN_DELIVERY          4
#define          TXN_STOCK
5
#define          TXN_EXIT
6
#define          TXN_LOGIN_RESULTS      7
#define          TXN_NEW_ORDER_RESULTS  8
#define          TXN_PAYMENT_RESULTS    9
#define          TXN_ORDER_STATUS_RESULTS 10
#define          TXN_DELIVERY_RESULTS   11
#define          TXN_STOCK_RESULTS
12

#define          CMD_NORD
"nord"
#define          CMD_PYMT
"pymt"
#define          CMD_ORDS
"ords"
#define          CMD_DLVY
"dlvy"
#define          CMD_STOK
"stok"
#define          CMD_EXIT
"exit"
#define          CMD_MENU
"menu"

#define          APP_NAME
"tpcc.html"
#define          HEADER
"Content-Type:text/html\r\nContent-Length: %d\r\nConnection:
Keep-Alive\r\n\r\n"

/////////////////////////////////////////////////////////////////
// URL Commands
/////////////////////////////////////////////////////////////////

#define          CMD_TXN_ID
"00"
#define          CMD_TERM_ID
"01"
#define          CMD_W_ID
"02"
#define          CMD_D_ID
"03"
#define          CMD_C_ID
"04"
#define          CMD_C_NAME
"05"
#define          CMD_C_W_ID
"06"
#define          CMD_C_D_ID
"07"
#define          CMD_AMT_PAID          "08"
#define          CMD_STK_THRESHOLD     "09"
#define          CMD_CARRIER_NUM
"10"

#define          CHAR_FILL              ''
#define          NUMERIC_FILL           ''
#define          NEGITIVE_SYMBOL        '-'
#define          MONEY_SYMBOL           '$'
#define          DECIMAL_SYMBOL         '.'
#define          ZERO_SYMBOL            '0'
#define          ZIP_DELIMITER          '-'
#define          PHONE_DELIMITER        '-'
#define          DATE_DELIMITER         '-'
#define          TIME_DELIMITER         ':'

#define          DEFAULT_MONEY64_LEN    15
#define          DEFAULT_MONEY32_LEN    9
#define          DEFAULT_MONEY16_LEN    9

#define          DEFAULT_NUMERIC64_LEN  15
#define          DEFAULT_NUMERIC32_LEN  9
#define          DEFAULT_NUMERIC16_LEN  9

#define          DEFAULT_DECIMAL64_LEN  5
#define          DEFAULT_DECIMAL32_LEN  5
#define          DEFAULT_DECIMAL16_LEN  5

#define          DEFAULT_DATETIME_LEN   19
#define          DEFAULT_DATE_LEN       11
#define          DEFAULT_TIME_LEN       8

#define          DEFAULT_STRING_LEN     25
#define          DEFAULT_ZIP_LEN        17
#define          DEFAULT_PHONE_LEN      18

/////////////////////////////////////////////////////////////////
// String Field Lengths
/////////////////////////////////////////////////////////////////

#define          NAME_LEN                24
#define          LAST_NAME_LEN          16
#define          FIRST_NAME_LEN         16
#define          INITIALS_LEN           2

#define          CREDIT_LEN              2

#define          STREET_LEN              20
#define          CITY_LEN                20
#define          STATE_LEN               2
#define          ZIP_LEN                 9

#define          PHONE_LEN               16
#define          DATA_LEN               200

#define          ITEM_LIST               15
#define          ORDER_LIST              10

/////////////////////////////////////////////////////////////////
// Type definitions
/////////////////////////////////////////////////////////////////

typedef __int8          INT8b;
typedef __int16         INT16b;

```

| | | | | |
|-------------------------------------|----------|----------------------|--|----------------------|
| typedef __int32 | | INT32b; | #define ERR_INVALID_ITEM_NUM | "ERROR: Invalid Item |
| typedef __int64 | | INT64b; | Number. Try Again." | |
| typedef unsigned __int8 | | UINT8b; | #define ERR_INVALID_ITEM_OTY | "ERROR: Invalid Item |
| typedef unsigned __int16 | UINT16b; | | Qty. Try Again." | |
| typedef unsigned __int32 | UINT32b; | | #define ERR_MISSING_C_ID_OR_CLAST | "ERROR: Must Enter |
| typedef unsigned __int64 | UINT64b; | | Customer Id or Customer Last Name. Try Again." | |
| | | | #define ERR_INVALID_PAYMENT_AMOUNT | "ERROR: Invalid |
| | | | Payment Amount. Try Again." | |
| typedef INT16b | | sqlint16; | #define ERR_INVALID_CARRIER | "ERROR: |
| typedef INT32b | | sqlint32; | Invalid Carrier Number. Try Again." | |
| typedef INT64b | | sqlint64; | #define ERR_INVALID_THRESHOLD | "ERROR: Invalid |
| | | | Threshold. Try Again." | |
| typedef INT16b | | int16_t; | #define ERR_INVALID_C_D_ID | "ERROR: |
| typedef INT32b | | int32_t; | Invalid Customer District Id. Try Again." | |
| typedef INT64b | | int64_t; | #define ERR_INVALID_C_W_ID | "ERROR: |
| | | | Invalid Customer Warehouse Id. Try Again." | |
| typedef char | | BYTE8b; | #define ERR_TERMINAL_FULL | "ERROR: |
| typedef double | | DOUBLE; | Terminal can not support user. Terminal full." | |
| typedef unsigned long | NATURAL; | | #define ERR_C_ID_OR_CLAST_ONLY | "ERROR: Either |
| | | | customer id or customer last name can be specified." | |
| //// | | | | |
| // Date and time values | | | #define ERR_UNABLE_TO_OPEN_REG | |
| //// | | | -50 | |
| #define SECONDS_IN_DAY | 86400 | | #define ERR_DLVY_THREAD_FAILED | |
| #define SECONDS_IN_HOUR | 3600 | | -51 | |
| #define SECONDS_IN_MINUTE | 60 | | #define ERR_DLVY_SEMAPHORE_INIT_FAILED | -52 |
| #define GMT_OFFSET | | 5 | #define ERR_DLVY_EVENT_INIT_FAILED | -53 |
| | | | #define ERR_DLVY_QUEUE_EATING_TAIL | -54 |
| #define DAYS_IN_YEAR | 365 | | | |
| #define YEARS_IN_LEAP | 4 | | #define ERR_INVALID_USERNAME | -70 |
| #define START_YEAR | | 1970 | #define ERR_INVALID_PASSWORD | -71 |
| #define MONTHS_IN_YEAR | 12 | | #define ERR_INVALID_DB_NAME | -72 |
| //// | | | #define ERR_INVALID_REGISTRY_KEY | -73 |
| // Error codes | | | #define ERR_DB2_DLL_NOT_LOADED | -74 |
| //// | | | #define ERR_ORACLE_DLL_NOT_LOADED | -75 |
| #define ERR_INVALID_TXN_TYPE | | -1 | #define ERR_CONNECT_ADDRESS_NOT_FOUND | -76 |
| | | | #define ERR_NORD_ADDRESS_NOT_FOUND | -77 |
| #define ERR_MISSING_W_ID | | -2 | #define ERR_PYMT_ADDRESS_NOT_FOUND | -78 |
| #define ERR_NON_NUMERIC_W_ID | | -3 | #define ERR_ORDS_ADDRESS_NOT_FOUND | -79 |
| #define ERR_MISSING_D_ID | | -4 | #define ERR_DLVY_ADDRESS_NOT_FOUND | -80 |
| #define ERR_NON_NUMERIC_D_ID | | -5 | #define ERR_STOK_ADDRESS_NOT_FOUND | -81 |
| #define ERR_MISSING_C_ID | | -6 | #define ERR_NULL_DLL_NOT_LOADED | -82 |
| #define ERR_NON_NUMERIC_C_ID | | -7 | #define ERR_UNKNOWN_DB | -83 |
| | | | #define ERR_DISCONNECT_ADDRESS_NOT_FOUND | -84 |
| #define ERR_MISSING_SUPP_W | | -8 | | |
| #define ERR_NON_NUMERIC_SUPP_W | | -9 | #define ERR_SAVING_CONTEXT | |
| #define ERR_MISSING_ITEM_NUM | | -10 | -90 | |
| #define ERR_NON_NUMERIC_ITEM_NUM | | -11 | #define ERR_DETACHING_CONTEXT | -91 |
| #define ERR_MISSING_ITEM_OTY | | -12 | #define ERR_ATTACHING_CONTEXT | -92 |
| #define ERR_NON_NUMERIC_ITEM_QTY | | -13 | #define ERR_HANDLE_IN_USE | -93 |
| | | | | |
| #define ERR_MISSING_CLAST_NAME | | -14 | | |
| #define ERR_NON_NUMERIC_CUST_W_ID | | -15 | #define ERR_CONNECT_TO_TM_FAILED | -99 |
| #define ERR_NON_NUMERIC_CUST_D_ID | | -16 | #define ERR_DLVY_LOG_OPEN_FAILED | -100 |
| #define ERR_MISSING_AMOUNT_PAID | | -17 | #define ERR_DLVY_QUEUE_FULL | |
| #define ERR_NON_NUMERIC_AMOUNT_PAID | | -18 | -101 | |
| | | | | |
| #define ERR_INVALID_D_ID | | "ERROR: | //// | |
| Invalid District ID. Try Again." | | | // Registry Definitions | |
| #define ERR_INVALID_W_ID | | "ERROR: | //// | |
| Invalid Warehouse ID. Try Again." | | | #define | REGISTRY_SUB_KEY |
| #define ERR_INVALID_C_ID | | "ERROR: Invalid | | "SOFTWARE\TPCC" |
| Customer ID. Try Again." | | | #define | DELIVERY_THREADS |
| #define ERR_INVALID_SUPPLY_W_ID | | "ERROR: Invalid Item | | "dlvyThreads" |
| Supply Warehouse. Try Again." | | | | |


```

#define DELIVERY_QUEUE_LEN
"dlvyQueueLen"
#define DELIVERY_LOG_PATH
"dlvyLogPath"
#define ERROR_LOG_FILE
"errorLogFile"
#define HTML_TRACE_LOG_FILE
"htmlTraceLogFile"
#define DB_NAME
"dbName"
#define NULL_DB
>nullDB"
#define COM_NULL_DB
"comnullDB"
#define CLIENT_NULL_DB
"clientNullDB"

#define NUM_USERS
"numUsers"
#define DB_TYPE
"dbType"

#define TXN_MONITOR
"txn_server"
#define COMM_POOL
"comm_pool"
#define HTML_TRACE
"htmlTrace"
#define ISAPI_TRACE
"isapi_trace"

#define DEFAULT_DLVY_THREADS 1
#define DEFAULT_DLVY_QUEUE_LEN
10
#define DEFAULT_DLVY_LOG_PATH
"c:\inetpub\wwwroot\tpcc\dlvy"
#define DEFAULT_ERROR_LOG_FILE
"c:\inetpub\wwwroot\tpcc\errorLog.txt"
#define DEFAULT_HTML_TRACE_LOG_FILE
"c:\inetpub\wwwroot\tpcc\htmlTrace.txt"
#define DEFAULT_NUM_USERS
10000

#define DEFAULT_DB_NAME
"tpcc"

// Structure defines
// Structure defines

struct nord_wrapper {
    struct in_neword_struct in_nord;
    struct out_neword_struct out_nord;
};

struct paym_wrapper {
    struct in_payment_struct in_paym;
    struct out_payment_struct out_paym;
};

struct ords_wrapper {
    struct in_ordstat_struct in_ords;
    struct out_ordstat_struct out_ords;
};

struct dlvy_wrapper {
    struct in_delivery_struct in_dlvy;
    struct out_delivery_struct out_dlvy;
};

```

```

};

struct stok_wrapper {
    struct in_stocklev_struct in_stok;
    struct out_stocklev_struct out_stok;
};

typedef struct
{
    int year;
    int month;
    int day;

    int hour;
    int minute;
    int second;
} datetime;

struct NEWORDERDATA
{
    struct in_items_struct {
        int s_OL_I_ID;
        int s_OL_SUPPLY_W_ID;
        short s_OL_QUANTITY;
    } in_item[15];

    long long in_s_O_ENTRY_D_time; /* init by SUT */
    int in_s_C_ID;
    int in_s_W_ID;
    short in_s_D_ID;
    short in_s_O_OL_CNT; /* init by SUT */
    short in_s_all_local;
    short in_duplicate_items;

    struct out_items_struct {
        double s_I_PRICE;
        double s_OL_AMOUNT;
        short s_S_QUANTITY;
        char s_I_NAME[25];
        char s_brand_generic;
    } out_item[15];

    long long out_s_O_ENTRY_D_time;
    double out_s_W_TAX;
    double out_s_D_TAX;
    double out_s_C_DISCOUNT;
    double out_s_total_amount;
    int out_s_O_ID;
    short out_s_O_OL_CNT;
    short out_s_transstat;
    short out_deadlocks;
    char out_s_C_LAST[17];
    char out_s_C_CREDIT[3];
};

struct PAYMENTDATA
{
    long long in_s_H_DATE_time;
    double in_s_H_AMOUNT;
    int in_s_W_ID;
    int in_s_C_W_ID;
    int in_s_C_ID;
    short in_s_C_D_ID;
    short in_s_D_ID;
    char in_s_C_LAST[17];

    long long out_s_H_DATE_time;
};

```

```

long long out_s_C_SINCE_time;
double out_s_C_CREDIT_LIM;
double out_s_C_BALANCE;
double out_s_C_DISCOUNT;
int out_s_C_ID;
short out_s_transtatus;
short out_deadlocks;
char out_s_W_STREET_1[21];
char out_s_W_STREET_2[21];
char out_s_W_CITY[21];
char out_s_W_STATE[3];
char out_s_W_ZIP[10];
char out_s_D_STREET_1[21];
char out_s_D_STREET_2[21];
char out_s_D_CITY[21];
char out_s_D_STATE[3];
char out_s_D_ZIP[10];
char out_s_C_FIRST[17];
char out_s_C_MIDDLE[3];
char out_s_C_LAST[17];
char out_s_C_STREET_1[21];
char out_s_C_STREET_2[21];
char out_s_C_CITY[21];
char out_s_C_STATE[3];
char out_s_C_ZIP[10];
char out_s_C_PHONE[17];
char out_s_C_CREDIT[3];
char out_s_C_DATA[20];
};

```

```

struct ORDERSTATUSDATA

```

```

{
    int in_s_C_ID;
    int in_s_W_ID;
    short in_s_D_ID;
    char in_s_C_LAST[17];

    double out_s_C_BALANCE;
    long long out_s_O_ENTRY_D_time;
    int out_s_C_ID;
    int out_s_O_ID;
    short out_s_O_CARRIER_ID;
    short out_s_ol_cnt;
    struct out_oitems_struct {
        long long s_OL_DELIVERY_D_time;
        double s_OL_AMOUNT;
        int s_OL_I_ID;
        int s_OL_SUPPLY_W_ID;
        short s_OL_QUANTITY;
    } out_item[15];
    short out_s_transtatus;
    short out_deadlocks;
    char out_s_C_FIRST[17];
    char out_s_C_MIDDLE[3];
    char out_s_C_LAST[17];
};

```

```

struct DELIVERYDATA

```

```

{
    long long in_s_O_DELIVERY_D_time;
    int in_s_W_ID;
    short in_s_O_CARRIER_ID;
    int out_s_O_ID[10];
    short out_s_transtatus;
    short outdeadlocks;
};

```

```

struct STOCKLEVELDATA

```

```

{
    int in_s_threshold;
    int in_s_W_ID;
    short in_s_D_ID;

    int out_s_low_stock;
    short out_s_transtatus;
    short out_deadlocks;
};

// MISCELLANEOUS HELPER FUNCTIONS
inline void appendText(char **string,char *text);
inline void appendText(char **string,char *text,int length,int justify);
inline void appendChar(char **string,char byte);
inline void DEBUGMSG(FILE * debugFile, char * message);
inline void appendSpaces(char **string,int spaces);

inline void calcOutDateTme(const INT64b value,datetime *timestamp);
inline int copyOutPhone(char *buffer,char *value,int len);
inline bool copyInMoney64(const char * value,INT64 *number);
inline int copyInMoney(const char *value);
inline void copyOutMoney64(char *buffer,INT64b value,unsigned int len);
inline int copyOutDateTme(char *buffer,INT64b value);
inline int copyOutDate(char *buffer,INT64b value);
inline int copyOutTime(char *buffer,INT64b value);
inline int copyOutDecimal64(char *buffer,INT64b value,unsigned int len);

```

```

inline UINT16b changeOrder16(UINT16b value);
inline UINT32b changeOrder32(UINT32b value);
inline UINT64b changeOrder64(UINT64b value);

```

```

inline INT16b changeOrder16(INT16b value);
inline INT32b changeOrder32(INT32b value);
inline INT64b changeOrder64(INT64b value);

```

```

//
// Name      : appendText
// Description :
//           Append text to string
// Parameters :
//           char ** - string point to append to
//           char * - text to append
// Returns   :
//           None
// Comments  :
//

```

```

inline void appendText(char **string,char *text)

```

```

{
    while(*text)
    {
        *(*string)++ = *text++;
    }

    **string='\0';
    return;
}

```

```

//
// Name      : appendText
// Description :
//           Append text to string
// Parameters :
//           char ** - string point to append to

```



```

// fixed days in each month (FEB 29 is special case)
static int daysInMonth[12] =
{31,28,31,30,31,30,31,31,30,31,30,31};

// mask out EPOC seconds
int dateValue = ((int) (value & 0xffffffff)) +
(SECONDS_IN_DAY -
(GMT_OFFSET * SECONDS_IN_HOUR));

int offset = (int) (value >> 32);

// break out the seconds
int hms = dateValue % SECONDS_IN_DAY;
int days = dateValue / SECONDS_IN_DAY;

int years = (days - 1) / DAYS_IN_YEAR;
int leaps = years / YEARS_IN_LEAP;

int daysUsed = (years * DAYS_IN_YEAR) + leaps;

// adjust the number of days to account for calculated years
days = days - daysUsed;

// set the starting year, month, and day
timestamp->day = 1;
timestamp->month = 1;
timestamp->year = START_YEAR + years;

// is the current year a leap year
int leap = !(timestamp->year % YEARS_IN_LEAP);

// apply remaining days based on days in months
int daysInCurrentMonth;

while(days)
{
// get days in current month
daysInCurrentMonth =
daysInMonth[timestamp->month - 1];
if(timestamp->month == 2 && leap)
daysInCurrentMonth =
daysInCurrentMonth + 1;

// days > days in current month
if(days > daysInCurrentMonth)
{
// increment month
timestamp->month += 1;
days = days -
daysInCurrentMonth;

// month exceeds months in year
if(timestamp->month >
MONTHS_IN_YEAR)
{
// increment year and
timestamp->year += 1;

reset month
timestamp->month = 1;

year
// are we now on a leap
leap =
!(timestamp->year % YEARS_IN_LEAP);
}
else
{

```

```

// set day of month to remaining
days
timestamp->day = days; days = 0;
}
}

// set time values to remaining seconds
timestamp->hour = hms / SECONDS_IN_HOUR;
hms = hms % SECONDS_IN_HOUR;

timestamp->minute = hms / SECONDS_IN_MINUTE;
timestamp->second = hms % SECONDS_IN_MINUTE;
return;
}

//
// copyOutZip
//
// Title : Copy zip data out of class array
// Parameters : char * - buffer to copy zip string into
//
// Return Value : int - Length of copy
// Comments :
//
inline int copyOutZip(char *buffer,char *value,int len = DEFAULT_ZIP_LEN)
{
int index = 0;
int bufferPos = 0;

// add each digit of zip number to buffer inserting delimiter at 5
while(value[index] && bufferPos < len)
{
if(index == 5)
buffer[bufferPos++] = ZIP_DELIMITER;

buffer[bufferPos++] = value[index++];
}

// space fill to the required length
while(bufferPos < len)
buffer[bufferPos++] = CHAR_FILL;

buffer[bufferPos] = NULL;
return len;
}

//
// copyOutPhone
//
// Title : Copy phone data out of class array
// Parameters : char * - buffer to copy phone string into
//
// Return Value : int - Length of copy
// Comments :
//
inline int copyOutPhone(char *buffer,char *value,int len =
DEFAULT_PHONE_LEN)
{
int index = 0;
int bufferPos = 0;

// add each digit of phone number to buffer inserting delimiter before
6, 9, and 12
while(value[index] && index < len)
{
switch(index)

```

```

    {
    case 6:
    case 9:
    case 12:
        // insert delimiter
        buffer[bufferPos++] = PHONE_DELIMITER;
    default:
        // add phone digit to buffer
        buffer[bufferPos++] = value[index++];
    }
}

// space fill to the required length
while(bufferPos < len)
    buffer[bufferPos++] = CHAR_FILL;

buffer[bufferPos] = '\0';

return len;
}

//
// copyInMoney64
//
// Title      : Copy money data into class array
// Parameters : const char * - value string
// Return Value : INT64b integer value
// Comments   :
//
inline bool copyInMoney64(const char * value,INT64b *number)
{
    //INT64b  number          = 0;
    int      index          = 0;
    int      decimal        = 0;
    int      decimals       = 0;
    int      digitsAfterDec = 0;

    bool     negativeFlag   = false;

    // convert each digit to a numeric portion
    while(value[index])
    {
        // handle $ . - All the rest assumed numeric
        switch(value[index])
        {
            case MONEY_SYMBOL:
                // ignore $ sign
                break;
            case NEGATIVE_SYMBOL:
                // set negative flag
                negativeFlag = true;
                break;
            case DECIMAL_SYMBOL:
                // set decimal
                decimal=1;
                decimals++;
                if(decimals > 1)
                    //more than 1 decimal point found
                    return false;
                break;
            default:
                // adjust decimal places
                decimal = decimal * 10;

                // add digit to running total
                if(value[index] >= '0' && value[index] <= '9')
                {
                    if(decimal)
                        if(++digitsAfterDec >
                            return false;

                    *number = (*number * 10) +
                        (value[index] - '0');
                }
                else
                {
                    //non-numeric field inserted
                    return false;
                }
                index++;
            }
        }

        // apply decimal where decimal not found
        if(decimal < 100)
        {
            if(decimal)
            {
                *number *= (100 / decimal);
            }
            else
            {
                *number *= 100;
            }
        }

        // make negative
        if(negativeFlag)
            *number = *number * (-1);

        return true;
    }
}

//
// copyInMoney
//
// Title      : Convert char string money field to double
// Parameters : const char * - value string
// Return Value : double integer value
// Comments   :
//
inline int copyInMoney(const char *value)
{
    char buf[20];
    int i,j,decimalFound,digitsAfterDecimal=0;

    int decimal=0;

    //walk past $ if present in char string
    if(*value == '$')
        *value++;

    int len=(int)strlen(value);
    for (i=0;i<len;i++)
    {
        if(value[i] == '.')
        {
            decimalFound++;
            if(decimalFound > 1)
                return -1;
        }
    }
}

```

```

    }
    if(value[i] == '-')

    if (value[i] != '.')
    {
        if(decimal)
        {
            if(digitsAfterDecimal<2)
                digitsAfterDecimal++;
            else
                return -1;
        }
        buf[j++] = value[i];
    }
}
int amount = atoi(buf);

return amount;
}

//
// copyOutMoney64
//
// Title           : Copy money data out of class array
// Parameters      : char * - buffer to copy string 64 bit money into
//                  INT64b - value
//                  unsigned len - max number of
//                  bytes to copy
// Return Value    : int - Length of copy
// Comments       :
//
inline void copyOutMoney64(char *buffer,INT64b value,unsigned int len =
DEFAULT_MONEY64_LEN)
{
    unsigned int    index          = len;
    int             places         = 0;
    bool            negativeFlag   = false;
    bool            moneyFlag      = true;

    // NULL terminate string
    buffer[index] = NULL;

    // check length > 0
    if(!index) return len;

    // handle negative value
    if(value < 0)
    {
        negativeFlag = true;
        value = value * (-1);
    }

    // break off each digit from value, fill if needed
    do
    {
        if(value)
        {
            // get next digit and add to buffer
            buffer[--index] = (char) (value % 10 + '0');
            value /= 10; places++;

            if(places == 2 && index)
                DECIMAL_SYMBOL;
        }
        else
        {
            // add zeros to first place before decimal point
            if(places < 2 || places == 3)
            {
                buffer[--index] =
                ZERO_SYMBOL;
            }
            else
            {
                // add the decimal point
                if(places == 2)
                {
                    buffer[--index] =
                    DECIMAL_SYMBOL;
                }
                else
                {
                    // add the negative
                    if(negativeFlag)
                    {
                        negativeFlag
                        = false;
                        buffer[--index] = NEGATIVE_SYMBOL;
                    }
                    else
                    {
                        // add the
                        money indicator
                    }
                }
            }
        }
    } while(index);

    // need to trace place for decimal point and
    zero fill
    places++;

    //return len;
}

//
// copyOutDateTime
//
// Title           : Copy date & time data out of class array
// Parameters      : char * - buffer to copy date & time string into
//                  INT64b - value
// Return Value    : int - Length of copy

```

```

// Comments          : Fixed length
//
inline int copyOutDateTime(char *buffer,INT64b value)
{
    datetime timestamp;

    // break value into    time/date components
    calcOutDateTime(value,&timestamp);

    // put month into buffer
    *buffer++ =      (char) ((timestamp.month / 10) + '0');
    *buffer++ =      (char) ((timestamp.month % 10) + '0');
    *buffer++ =      DATE_DELIMITER;

    // put day into      buffer
    *buffer++ =      (char) ((timestamp.day / 10) + '0');
    *buffer++ =      (char) ((timestamp.day % 10) + '0');
    *buffer++ =      DATE_DELIMITER;

    // put year  into buffer
    int      year = timestamp.year;
    *buffer++ =      (char) ((year /      1000) +  '0');
year = year% 1000;
    *buffer++ =      (char) ((year /      100) + '0'); year = year
% 100;
    *buffer++ =      (char) ((year /      10)  + '0');
    *buffer++ =      (char) ((year %      10)  + '0');
    *buffer++ =      CHAR_FILL;

    // put hour  into buffer
    *buffer++ =      (char) ((timestamp.hour  / 10) +
'0');
    *buffer++ =      (char) ((timestamp.hour  % 10) +
'0');
    *buffer++ =      TIME_DELIMITER;

    // put minute into buffer
    *buffer++ =      (char) ((timestamp.minute / 10)  +
'0');
    *buffer++ =      (char) ((timestamp.minute % 10)  +
'0');
    *buffer++ =      TIME_DELIMITER;

    // put second into buffer
    *buffer++ =      (char) ((timestamp.second / 10)  +
'0');
    *buffer++ =      (char) ((timestamp.second % 10)  +
'0');

    *buffer  = NULL; return DEFAULT_DATETIME_LEN;
}
//
// copyOutTime
//
// Title          : Copy date data out of class array
// Parameters     : char * - buffer to copy date string into
//                INT64b - value
// Return Value   : int - Length of copy
// Comments      : Fixed length TBD
//
inline int copyOutTime(char *buffer,INT64b value)
{
    datetime timestamp;

    // break value into time/date components
    calcOutDateTime(value,&timestamp);

    // put hour into buffer
    *buffer++ = (char) ((timestamp.hour / 10) + '0');
    *buffer++ = (char) ((timestamp.hour % 10) + '0');
    *buffer++ = TIME_DELIMITER;

    // put minute into buffer
    *buffer++ = (char) ((timestamp.minute / 10) + '0');
    *buffer++ = (char) ((timestamp.minute % 10) + '0');
    *buffer++ = TIME_DELIMITER;

    // put second into buffer
    *buffer++ = (char) ((timestamp.second / 10) + '0');
    *buffer++ = (char) ((timestamp.second % 10) + '0');

    *buffer = NULL; return DEFAULT_TIME_LEN;
}
//
// copyOutDecimal64
//
// Title          : Copy decimal data out of class array
// Parameters     : char * - buffer to copy string 64 bit money into
//                INT64b - value
//                unsigned len - max number of
//                bytes to copy
// Return Value   : int - Length of copy
// Comments      :

```



```

#ifdef _DEBUG
    #define DEBUGMSG(TEXT)
\
EnterCriticalSection(&debugMutex);
\
    debugStream << debugFileName(__FILE__)
    _TIMESTAMP_ << "|" << _LINE_ << "|"
    << "|" << GetCurrentThreadId() << "|"
\
debugStream.flush();
\
LeaveCriticalSection(&debugMutex);

    #define DEBUGSTRING(TEXT,LENGTH)
        debugVarString(TEXT,LENGTH)
\

#else
    #define DEBUGMSG(TEXT);
    #define DEBUGSTRING(TEXT,LENGTH);

#endif
#endif /* _COMMON_TPCC */

TpccIsapi.cpp

/*
*****
** Project      : AIX
** Component    : Performance/TPC-C Benchmark
** Name        : tpccIsapi.cpp
** Title       : TPCC html processing
*****
** Copyright (c) 2003 IBM Corporation
** All rights reserved
*****
** History      :
**      Developed at IBM Austin by the AIX RS/6000
**      performance group.
**
** Comments    :
**
*****
*/

#include "stdafx.h"

#include "..\tpccCom\tpccCom.h"
#include "..\tpccCom\tpccCom_i.c"
#include <tpccIsapi.hpp>

// For custom assert and trace handling with WebDbg.exe
[ module(name="tpccIsapi", type="dll") ];
[ emitidl(restricted) ];

#define _WIN32_DCOM

////////////////////////////////////
// Globals
////////////////////////////////////

int          maxSize;
//max struct size of all txn(s)
int          numUsers;
//number of users that client will service.
int          dlvyQueueLen;
//static length of dlvy queue
int          dlvyThreads;
//number of dlvy threads to create
int          dlvyBufferFreeSlots;           //length of dlvy txn
queue
int          dlvyBufferSlotIndex;          //index into next
available slot in dlvy txn queue
int          dlvyBufferThreadIndex;        //thread
index into dlvy txn queue
int          nullDB;
//null db on client(bypass com call).

int          trace;

static DWORD          threadLSIndex;
//isapi thread local storage index
CRITICAL_SECTION     isapiLock;
//isapi lock
CRITICAL_SECTION     errorLock;
//error log file lock.
CRITICAL_SECTION     termLock;
//terminal array lock.
CRITICAL_SECTION     dlvyQueueLock;
//dlvy queue critical section lock
HANDLE                dlvyThreadDone =
INVALID_HANDLE_VALUE; //dlvy thread exit event
HANDLE                dlvyThreadSemaphore
= INVALID_HANDLE_VALUE; //dlvy thread wrk to do semaphore
int
dlvyThreadID = 0;

struct DLVYQUEUEDATA *dlvyQueue;
//dlvy queue
HANDLE                *dlvyThreadHandles;
//ptr to array of thread handles

TERM_ENTRY            *termArray;
//array of terminal entries to store each users info.
int                  termNextFree;
//next available slot in terminal array

FILE                 *htmlDebug          =          NULL;
//html debug file
FILE                 *errorLog           =          NULL;
//error file
FILE                 *htmlTrace          =          NULL;

ofstream debugStream;
ofstream errorStream;
CRITICAL_SECTION debugMutex;
CRITICAL_SECTION errorMutex;

```

```

char    dlvyLogPath[128]    = {NULL};
char    errorLogFile[128]  = {NULL};
char    htmlTraceLogFile[128] = {NULL};
char    dbName[64]        = {NULL};
char    dbType[16]        = {NULL};

```

```

typedef INT (*CONNECT_PTR)(char *dbName,void **connectHandle);
typedef INT (*DISCONNECT_PTR)(void *connectHandle);
typedef INT (*DLVY_FUNC_PTR)(dlvy_wrapper *dlvy,void
*connectHandle);
typedef INT (*NORD_FUNC_PTR)(nord_wrapper *nord,void
*connectHandle);
typedef INT (*PYMT_FUNC_PTR)(paym_wrapper *pymt,void
*connectHandle);
typedef INT (*ORDS_FUNC_PTR)(ords_wrapper *ords,void *connectHandle);
typedef INT (*STOK_FUNC_PTR)(stok_wrapper *stok,void *connectHandle);

```

```

HINSTANCE          dbInstance;
CONNECT_PTR        db_connect;
DISCONNECT_PTR    db_disconnect;
DLVY_FUNC_PTR      dlvyCall;

```

```

////////////////////////////////////
// Page functions arrays
////////////////////////////////////

```

```

typedef int (*pageFuncPtr) (htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle);

```

```

pageFuncPtr htmlPageFunctions[MAX_TRANSACTIONS] =
{
    {doLoginForm},
    {doNewOrderForm},
    {doPaymentForm},
    {doOrderStatusForm},
    {doDeliveryForm},
    {doStockForm},
    {doExit},
    {doLoginResults},
    {doNewOrderResults},
    {doPaymentResults},
    {doOrderStatusResults},
    {doDeliveryResults},
    {doStockResults}
};

```

```

extern "C" DWORD WINAPI
HttpExtensionProc(LPEXTENSION_CONTROL_BLOCK lpECB)
{
    struct TXN_HANDLE *txnHandle = NULL;

    txnHandle = (TXN_HANDLE *) TlsGetValue(threadLSIndex);

    if(txnHandle == NULL)
    {
        int rc = initTxnHandle(&txnHandle);
        if (rc != OK)
        {
            char response[256]; char htmlHeader[256];
            sprintf(response,"ERROR: Init txnHandle
function failed.\n");

            size_t htmlPageLen = strlen(response);

            //add content length and keep alive header
            sprintf(htmlHeader,HEADER,htmlPageLen);

```

```

lpECB->ServerSupportFunction(lpECB->ConnID,HSE_REQ_SEND_RESPO
NSE_HEADER,"200 OK",NULL,(DWORD*)htmlHeader);

lpECB->WriteClient(lpECB->ConnID,response,(LPDWORD)&htmlPageLen,0
);

```

```

return
HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}

txnHandle = (TXN_HANDLE *)
TlsGetValue(threadLSIndex);
if (txnHandle == NULL)
{
    char response[256]; char htmlHeader[256];
    sprintf(response,"ERROR: Unable to retrieve
txnHandle from TLS.\n");

    size_t htmlPageLen = strlen(response);

    //add content length and keep alive header
    sprintf(htmlHeader,HEADER,htmlPageLen);

```

```

lpECB->ServerSupportFunction(lpECB->ConnID,HSE_REQ_SEND_RESPO
NSE_HEADER,"200 OK",NULL,(DWORD*)htmlHeader);

```

```

lpECB->WriteClient(lpECB->ConnID,response,(LPDWORD)&htmlPageLen,0
);

```

```

return
HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}

try
{
    txnHandle->urlString =
(char*)lpECB->lpszQueryString;

    DEBUGMSG("calling doHtml() w/ query string:" <<
txnHandle->urlString << endl);
    doHtml(txnHandle);

    size_t htmlPageLen;
    htmlPageLen = strlen(txnHandle->htmlPage);
    if(htmlPageLen >= 4096)
    {
        ERRORMSG("WARNING: HTML PAGE IS
>= 4096!, page size:"<<htmlPageLen<<endl);
    }
    //add content length and keep alive header
    sprintf(txnHandle->htmlHeader,HEADER,htmlPageLen);
    size_t headerLen = strlen(txnHandle->htmlHeader);
    if(headerLen >= 256)
    {
        ERRORMSG("WARNING: HTML
HEADER IS >= 256!, header size:"<<headerLen<<endl);
    }

    //write response to user

```

```

lpECB->ServerSupportFunction(lpECB->ConnID,HSE_REQ_SEND_RESPO
NSE_HEADER,"200 OK",NULL,(DWORD*)txnHandle->htmlHeader);

```

```

lpECB->WriteClient(lpECB->ConnID,txnHandle->htmlPage,(LPDWORD)&ht
mlPageLen,0);

```

```

        DEBUGMSG("HTML
PAGE--><<endl<<txnHandle->htmlHeader<<txnHandle->htmlPage<<endl);
    }
    catch (...)
    {
        char response[256];
        ZeroMemory(response,256);
        char *ptr = response;

        appendText(&ptr,"<HTML><BODY> Error : Unhandled
Exception </BODY></HTML>");
        DWORD cbResponse = sizeof(response)-1 ;

        //write response to user

lpECB->ServerSupportFunction(lpECB->ConnID,HSE_REQ_SEND_RESPO
NSE_HEADER,"200 OK",NULL,(DWORD*)response);

lpECB->WriteClient(lpECB->ConnID,response,&cbResponse,0);
    }

    return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}

extern "C" BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO*
pVer)
{
    // Create the extension version string, and copy string to
HSE_VERSION_INFO structure.
    pVer->dwExtensionVersion =
MAKELONG(HSE_VERSION_MINOR, HSE_VERSION_MAJOR);

    // Copy description string into HSE_VERSION_INFO structure.
strcpy(pVer->lpszExtensionDesc, "TPCC ISAPI Extension");

    // Initialize isapi critical section
InitializeCriticalSection(&isapiLock);

    // Initialize error log critical section
InitializeCriticalSection(&errorLogLock);

    // Initialize terminal critical section
InitializeCriticalSection(&termLock);

    // Initialize debug/error critical sections
if(debugFlag)
    InitializeCriticalSection(&debugMutex);
InitializeCriticalSection(&errorMutex);

    // Read registry values
if(readRegistryValues() != OK)
    return(FALSE);

    // Initialize terminal array
termArray = (TERM_ENTRY*)
calloc(numUsers,sizeof(TERM_ENTRY));
termNextFree = 1;

    //open up error/debug streams
errorStream.rdbuf( )->open(errorLogFile,ios::out);
if(debugFlag)
    debugStream.rdbuf( )->open(htmlTraceLogFile,ios::out);

    ERRORMSG("Error log file open."<<endl);

    DEBUGMSG("Loading library for dlvy txn."<<endl);
    int rc = getDBInstance();

```

```

    if (rc != OK)
    {
        ERRORMSG("Error, unable to load database dll,
rc:"<<rc);
        DEBUGMSG("Error, unable to load database dll,
rc:"<<rc);

        return FALSE;
    }
    DEBUGMSG("Library loaded for dlvy txn."<<endl);

    DEBUGMSG("Calling initDlvy." <<endl);

    if(initDlvy() != OK)
        return (FALSE);

    DEBUGMSG("Initializing TLS." << endl);

    // Initialize thread local storage index
threadLSIndex = TlsAlloc();
if (threadLSIndex == TLS_NULL)
{
    ERRORMSG("Isapi error: unable to initialize thread
local storage(TLS), rc:" << GetLastError()<<endl);
    return(FALSE);
}

    DEBUGMSG("sizeof out_neword_struct: "<<sizeof(struct
out_neword_struct)<<endl);
    DEBUGMSG("sizeof in_neword_struct: "<<sizeof(struct
in_neword_struct)<<endl);
    DEBUGMSG("sizeof out_payment_struct: "<<sizeof(struct
out_payment_struct)<<endl);
    DEBUGMSG("sizeof in_payment_struct: "<<sizeof(struct
in_payment_struct)<<endl);
    DEBUGMSG("sizeof out_ordstat_struct: "<<sizeof(struct
out_ordstat_struct)<<endl);
    DEBUGMSG("sizeof in_ordstat_struct: "<<sizeof(struct
in_ordstat_struct)<<endl);
    DEBUGMSG("sizeof out_delivery_struct: "<<sizeof(struct
out_delivery_struct)<<endl);
    DEBUGMSG("sizeof in_delivery_struct: "<<sizeof(struct
in_delivery_struct)<<endl);
    DEBUGMSG("sizeof out_stocklev_struct: "<<sizeof(struct
out_stocklev_struct)<<endl);
    DEBUGMSG("sizeof in_stocklev_struct: "<<sizeof(struct
in_stocklev_struct)<<endl);

    //compute the max struct size for com data construct
maxDataSize = max(maxDataSize,sizeof(nord_wrapper));
maxDataSize = max(maxDataSize,sizeof(paym_wrapper));
maxDataSize = max(maxDataSize,sizeof(ords_wrapper));
maxDataSize = max(maxDataSize,sizeof(dlvy_wrapper));
maxDataSize = max(maxDataSize,sizeof(stok_wrapper));
maxDataSize += 10;

    DEBUGMSG("max data struct size:"<<maxDataSize <<endl);

    return true;
}

extern "C" BOOL WINAPI TerminateExtension(DWORD dwFlags)
{
    return true;
}

/*

```

```

*****
** Name          :          initTxnHandle
** Description   :
**              :          Isapi thread initializes
its own com interface
**              :          structure.
** Parameters    :
**              :          TXN_HANDLE**
isapi txn handle
** Returns      :
**              :          int - return code
** Comments     :
**
*****
*/
int initTxnHandle(TXN_HANDLE **txnHandle)
{
    DEBUGMSG("Inside init txn handle, getting isapiLock." << endl);
    EnterCriticalSection(&isapiLock);

    HRESULT hres = NULL;
    try
    {
        DEBUGMSG("Got ispaiLock, initializing txnHandle:
"<<DEBUGADDRESS(*txnHandle)<< endl);
        *txnHandle = (TXN_HANDLE *)
        calloc(1,sizeof(TXN_HANDLE));
        if (*txnHandle == NULL)
        {
            ERRORMSG("Unable to allocated
TXN_HANDLE, rc:"<<GetLastError()<<endl);
            return ERR;
        };

        (*txnHandle)->comInterface.comHandle = NULL;
        DEBUGMSG("Initializing txnHandle com data buffer to
"<<maxDataSize<<"bytes"<<endl);
        (*txnHandle)->comInterface.txnBuffer = (char *)
        CoTaskMemAlloc(maxDataSize);
        if (!((*txnHandle)->comInterface.txnBuffer))
        {
            ERRORMSG("CoTaskMemAlloc() failed of
size "<<maxDataSize<<"; rc: "<<hres<<endl);
            return(ERR);
        };
        DEBUGMSG("txnHandle com data buffer initialized to "
<< maxDataSize << "bytes" <<endl);

        DEBUGMSG("Calling CoInitialize with txnHandle:
"<<DEBUGADDRESS(*txnHandle)<<endl);
        hres =
        CoInitializeEx(NULL,COINIT_MULTITHREADED);
        if (FAILED(hres))
        {
            ERRORMSG("CoInitializeEx() failed, rc :
"<<hres<<endl);
            return(ERR);
        };

        struct _timeb
        startTime;
        struct _timeb
        endTime;

        DEBUGMSG("Calling CoCreateInstance with
txnHandle:"<<DEBUGADDRESS(*txnHandle)<< endl);
        _ftime(&startTime);

```

```

        hres =
        CoCreateInstance(CLSID_tpcc_com,NULL,CLSCTX_SERVER,IID_Itpcc_co
m,(void **)&(*txnHandle)->comInterface.comHandle);
        if (FAILED(hres))
        {
            _ftime(&endTime);
            //store error code in txnHandle
            ERRORMSG("CoCreateInstance() failed,
code:"<<HRESULT_CODE(hres)<<"
facility:"<<HRESULT_FACILITY(hres)<<
" hres:"<<hres<< " time
waiting:"<<
            (((endTime.time -
startTime.time)*1000)+
            (endTime.millitm -
startTime.millitm))/1000.0)<<endl);

            DEBUGMSG("CoCreateInstance() failed,
code:"<<HRESULT_CODE(hres)<<"
facility:"<<HRESULT_FACILITY(hres)<<
" hres:"<<hres<< " time
waiting:"<<
            (((endTime.time -
startTime.time)*1000)+
            (endTime.millitm -
startTime.millitm))/1000.0)<<endl);

            return(ERR);
        };

        _ftime(&endTime);
        DEBUGMSG("CoCreateInstance successful.txnHande
com initialized, time waiting for object to be activated." <<
            (((endTime.time - startTime.time)*1000)+
            (endTime.millitm -
startTime.millitm))/1000.0)<<endl);

        //call set complete to return object to pool.
        (*txnHandle)->comInterface.comHandle->doSetComplete();

        //set the com buffers size
        DEBUGMSG("Setting txnHandle: " <<
        DEBUGADDRESS(*txnHandle) << "com buffer size to " << maxDataSize<<
        endl)
        (*txnHandle)->comInterface.size = maxDataSize;

        DEBUGMSG("txnHandle:
"<<DEBUGADDRESS(*txnHandle) <<"set to " << maxDataSize << endl);

        TlsSetValue(threadLSIndex,*txnHandle);

        DEBUGMSG("txnHandle:
"<<DEBUGADDRESS(*txnHandle) << "stored in TLS" << endl);

        ZeroMemory((*txnHandle)->htmlPage,MAX_HTML_PAGE_LEN);

        ZeroMemory((*txnHandle)->htmlHeader,MAX_HTML_HEADER_LEN);

        LeaveCriticalSection(&isapiLock);
        return(OK);
    }
    catch(...)
    {
        DEBUGMSG("Unhandled exception in initTxnHandle,
unlocking isapi lock" <<endl);

```

```

        ERRORMSG("Unhandled exception in initTxnHandle,
unlocking isapi lock" <<endl);
        LeaveCriticalSection(&isapiLock);
    };

    return ERR;
}

/*
*****
** Name          :          getDBInstance
** Description    :
**               load db specific lib
based on dbType registry
**               value.
** Parameters     :
**
** Returns       :
**               int - return code
** Comments      :
**               This function only
exists for the dlvy threads
**               Dlvvy threads hold
direct connections to the database
**               and therefore need to
know what db interface to talk to.
*****
*/
int getDBInstance()
{
    if(nullDB)
    {
        dbInstance =
LoadLibrary("c:\inetpub\wwwroot\tpcc\nullDB.dll");
        if(dbInstance == NULL)
        {
            return ERR_NULL_DLL_NOT_LOADED;
        }
    }
    else if( (strcmp(dbType,"DB2") == 0) )
    {
        dbInstance =
LoadLibrary("c:\inetpub\wwwroot\tpcc\tpccDB2glue.dll");
        if(dbInstance == NULL)
        {
            return ERR_DB2_DLL_NOT_LOADED;
        }
    }
    else if( (strcmp(dbType,"ORACLE") == 0) )
    {
        return ERR_ORACLE_DLL_NOT_LOADED;
    }
    else
    {
        return ERR_UNKNOWN_DB;
    }

    db_connect =
(CONNECT_PTR)GetProcAddress(dbInstance,"connect_db");
    if(db_connect == NULL)
    {
        return ERR_CONNECT_ADDRESS_NOT_FOUND;
    }
    dlvyCall =
(DLVY_FUNC_PTR)GetProcAddress(dbInstance,"do_dlvy");
    if(dlvyCall == NULL)
    {
        return ERR_DLVY_ADDRESS_NOT_FOUND;
    }
}

```

```

    }
    return OK;
}

/*
*****
** Name          :          initDlvy
** Description    :
**               initialize dlvy
threads/dlvy queueu
** Parameters     :
**
** Returns       :
**               int - return code
** Comments      :
**
*****
*/
int initDlvy()
{
    // Initialize critical section
    InitializeCriticalSection(&dlvyQueueLock);

    //create dlvy queue
    dlvyQueue = (DLVYQUEUEEDATA *)
calloc(dlvyQueueLen,sizeof(DLVYQUEUEEDATA));

    dlvyThreadDone = CreateEvent(NULL,
TRUE, //manual reset
FALSE, //initially not signalled.
NULL);
    if(dlvyThreadDone == NULL)
    {
        DEBUGMSG("Error: dlvyThreadDone handled init
failed, GetLastError:<<GetLastError()<<endl);

        ERRORMSG("Error : dlvyThreadDone handled init
failed, GetLastError:<<GetLastError()<<endl);

        return ERR_DLVY_EVENT_INIT_FAILED;
    }

    //create dlvy semaphore
    dlvyThreadSemaphore =
CreateSemaphore(NULL,0,dlvyQueueLen,NULL);
    if(dlvyThreadSemaphore == NULL)
    {
        DEBUGMSG("Error: dlvyThreadSemaphore semaphore
init failed, GetLastError:<<GetLastError()<<endl);
        ERRORMSG("Error: dlvyThreadSemaphore semaphore
init failed, GetLastError:<<GetLastError()<<endl);
        return ERR_DLVY_SEMAPHORE_INIT_FAILED;
    }

    //set number of free slots available in queue
    dlvyBufferFreeSlots = dlvyQueueLen;

    //index into next available slot in dlvy txn queue
    dlvyBufferSlotIndex = 0;

    //thread index into dlvy txn queue
    dlvyBufferThreadIndex = 0;
}

```

```

        dlvyThreadHandles = new HANDLE[dlvyThreads];
        //create threads
        for(int threadCount = 0;threadCount < dlvyThreads;threadCount++)
        {
            dlvyThreadHandles[threadCount] =
(HANDLE)_beginthread(dlvyThreadEntry,0,NULL);
            if(dlvyThreadHandles[threadCount] ==
INVALID_HANDLE_VALUE)
                return ERR_DLVY_THREAD_FAILED;
        }

        return OK;
    }

/*
*****
** Name          :          readRegistryValues
** Description   :
**              initialize isapi global
variables from registry
** Parameters    :
**              :
** Returns       :
**              int - return code
** Comments      :
**              :
*****
*/
int readRegistryValues()
{
    HKEY    registryKey;
    char    value[MAX_STRING_LEN];
    DWORD   regType;
    DWORD   regValue;
    DWORD   regValueSize = MAX_STRING_LEN;

    //open up registry key

    if(RegOpenKeyEx(HKEY_LOCAL_MACHINE,REGISTRY_SUB_KEY,0,KEY_READ,&registryKey) != ERROR_SUCCESS)
        return ERR_UNABLE_TO_OPEN_REG;

    //get null db flag
    regValueSize = sizeof(regValue);
    if(RegQueryValueEx(registryKey,NULL_DB,0,&regType,(BYTE
*)&regValue,&regValueSize) == ERROR_SUCCESS)
        nullDB = regValue;

    else
        nullDB = 0;

    //get num dlvy threads
    regValueSize = sizeof(regValue);

    if(RegQueryValueEx(registryKey,DELIVERY_THREADS,0,&regType,(BYTE
E *)&regValue,&regValueSize) == ERROR_SUCCESS)
        dlvyThreads = regValue;

    else
        dlvyThreads =
DEFAULT_DLVY_THREADS;

    //get dlvy queue len
    regValueSize = sizeof(regValue);

    if(RegQueryValueEx(registryKey,DELIVERY_QUEUE_LEN,0,&regType,(B
YTE *)&regValue,&regValueSize) == ERROR_SUCCESS)
        dlvyQueueLen = regValue;

```

```

        else
            dlvyQueueLen =
DEFAULT_DLVY_QUEUE_LEN;

        //get the htmlTrace flag
        regValueSize = sizeof(regValue);

    if(RegQueryValueEx(registryKey,HTML_TRACE,0,&regType,(BYTE
*)&regValue,&regValueSize) == ERROR_SUCCESS)
        trace = regValue;

    else
        trace = 0;

    //get the client null db flag
    regValueSize = sizeof(regValue);
    if(RegQueryValueEx(registryKey,NULL_DB,0,&regType,(BYTE
*)&regValue,&regValueSize) == ERROR_SUCCESS)
        nullDB = regValue;

    else
        nullDB = 0;

    //get the num of users
    regValueSize = sizeof(regValue);

    if(RegQueryValueEx(registryKey,NUM_USERS,0,&regType,(BYTE
*)&regValue,&regValueSize) == ERROR_SUCCESS)
        numUsers = regValue;

    else
        numUsers = DEFAULT_NUM_USERS;

    //get dlvy log file path
    regValueSize = sizeof(value);
    if
(RegQueryValueEx(registryKey,DELIVERY_LOG_PATH,0,&regType,(BYT
E *)&value,&regValueSize)== ERROR_SUCCESS )
        strcpy(dlvyLogPath,value);

    else
        strcpy(dlvyLogPath,DEFAULT_DLVY_LOG_PATH);

    //get global error log file path/name
    regValueSize = sizeof(value);
    if
(RegQueryValueEx(registryKey,ERROR_LOG_FILE,0,&regType,(BYTE *)
&value,&regValueSize)== ERROR_SUCCESS )
        strcpy(errorLogFile,value);

    else
        strcpy(errorLogFile,DEFAULT_ERROR_LOG_FILE);

    //get global error log file path/name
    regValueSize = sizeof(value);
    if
(RegQueryValueEx(registryKey,HTML_TRACE_LOG_FILE,0,&regType,(B
YTE *)&value,&regValueSize)== ERROR_SUCCESS )
        strcpy(htmlTraceLogFile,value);

    else
        strcpy(htmlTraceLogFile,DEFAULT_HTML_TRACE_LOG_FILE);

    //get db name
    regValueSize = sizeof(value);
    if (RegQueryValueEx(registryKey,DB_NAME,0,&regType,(BYTE
*)&value,&regValueSize)== ERROR_SUCCESS )
        strcpy(dbName,value);

    else
        strcpy(dbName,DEFAULT_DB_NAME);

    //get db type
    regValueSize = sizeof(value);

```

```

        if (RegQueryValueEx(registryKey,DB_TYPE,0,&regType,(BYTE
*) &value,&regValueSize)== ERROR_SUCCESS )
            strcpy(dbType,value);

        RegCloseKey(registryKey);

        return OK;
    }

/*
*****
** Name          : doLoginForm
** Description   :
**              HTML Login page entry point
** Parameters    :
**              htmlPhraser*    command
**              TXN_HANDLE*    txn handle
** Returns      :
**              int - return code
** Comments     :
**
*****
*/

int doLoginForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    DEBUGMSG("Entering doLoginForm()."<<endl);
    char *html=txnHandle->htmlPage;

    DEBUGMSG("Creating html login page"<<endl);
    //begin html page
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Client
Home Page</TITLE></HEAD>"

    "ACTION=""
    APP_NAME
    ""
    METHOD=""GET""
    "<H2>Please
    "<INPUT
    TYPE=""hidden"" NAME=""
    CMD_TXN_ID
    "" VALUE=""
    CMD_MENU
    "">"
    "<H3>Warehouse
    <INPUT NAME=""
    CMD_W_ID
    "" SIZE=6""
    " District <INPUT
    NAME=""
    CMD_D_ID
    "" SIZE=2></H3>"
    "<INPUT
    TYPE=""submit"" VALUE=""Submit"">"
    "</FORM>");

    html+=sprintf(html,"dlvy Queue Length:%d <BR> num dlvy threads:%d <BR>
dlvy queue free slots:%d <BR> isapi queue index:%d <BR> thread queue
index:%d <BR> </BODY></HTML>\n",
                dlvyQueueLen,
                dlvyThreads,
                dlvyBufferFreeSlots,
                dlvyBufferSlotIndex,

```

```

        dlvyBufferThreadIndex);
    DEBUGMSG("Html login page done"<<endl);

    return OK;
}

/*
*****
** Name          : doLoginResults
** Description   :
**              HTML Login results page entry
** Parameters    :
**              htmlPhraser*    command
**              TXN_HANDLE*    txn handle
** Returns      :
**              int - return code
** Comments     :
**
*****
*/

int doLoginResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    char *html=txnHandle->htmlPage;

    //validate parameters
    if( (txnHandle->w_id = atoi(commandBlock->get_W_ID())) == 0 )
    {
        doLoginErrorPage(html,ERR_INVALID_W_ID);
        return OK;
    }
    if( (txnHandle->d_id = atoi(commandBlock->get_D_ID())) == 0 )
    {
        doLoginErrorPage(html,ERR_INVALID_D_ID);
        return OK;
    }

    //store user into terminal array,
    //function will ERR if the terminal array is full
    if( assignTerminal(txnHandle) != OK)
    {
        doLoginErrorPage(html,ERR_TERMINAL_FULL);
        return OK;
    };

    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Main
Menu</TITLE></HEAD>\r\n"

    "<BODY><FORM
    ACTION=""
    APP_NAME
    ""
    METHOD=""GET"">\r\n"
    "<H3>Please Select
    Transaction.</H3>\r\n");
    html+=appendButtons(html);
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"</FORM></BODY></HTML>");

    return OK;
}

/*
*****
** Name          : doLoginErrorPage

```

```

** Description      :
**                  HTML Login page entry point
** Parameters      :
**                  char *          html page
buffer
**                  char *          error
message
** Returns         :
**                  int - return code
** Comments        :
**
*****
*/

int doLoginPage(char *htmlPage,char *errorMessage)
{
    char *html=htmlPage;

    //begin html page
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Client
Home Page</TITLE></HEAD>"

ACTION="\ ""
APP_NAME
""

METHOD="\GET">);
appendText(&html,"<H2>Please Login.</H2>"

TYPE="hidden" NAME=""

CMD_TXN_ID
"" VALUE=""
CMD_MENU
">"
"<H3>Warehouse

<INPUT NAME=""

CMD_W_ID
"" SIZE=6>"
" District <INPUT

NAME=""

CMD_D_ID
"" SIZE=2></H3>"
"<INPUT

TYPE="submit" VALUE="Submit">"

"</FORM>");
appendText(&html,errorMessage);
appendText(&html,"<BODY></HTML>");

return OK;
}

/*
*****
** Name           : doNewOrderForm
** Description    :
**                  HTML neworder page entry point
** Parameters     :
**                  htmlPhraser*    command
block
**                  TXN_HANDLE*     txn handle
struct
** Returns       :
**                  int - return code
** Comments      :
**
*****
*/

```

```

int doNewOrderForm(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
    char *html=txnHandle->htmlPage;

    appendText(&html,"<HTML><HEAD><TITLE>TPC-C New
Order</TITLE></HEAD>\r\n"

" <BODY><FORM

ACTION=""

APP_NAME
""

METHOD="\GET">\r\n"

"<CENTER><H3>Please Fill In New Order Form.</H3></CENTER>\r\n"
//check if not needed

"Submit Transaction

CMD_TXN_ID
"" VALUE=""
CMD_NORD
"">);

//append the hidden
html+=appendHiddenFields(html,txnHandle);

//int buffer for warehouse
char buffer[15];
appendText(&html," <PRE>"

//
" 1 2 3
4 5 6 7 8 9\r\n"

//
"123456789012345678901234567890123456789012345678901234567890123
456789012345678901234567890\r\n"

"Warehouse: ");
appendText(&html,itoa(txnHandle->w_id,buffer,10),7,1);
appendText(&html,"District: <INPUT NAME=""

CMD_D_ID
"" SIZE=1>

Date:<BR>"
"Customer <INPUT NAME=""

CMD_C_ID
"" SIZE=6> Name:

Credit: %Disc.:<BR>"

"Order Number:

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

//
" 1 2 3
4 5 6 7 8 9\r\n"

//
"123456789012345678901234567890123456789012345678901234567890123
456789012345678901234567890\r\n"

" Supp_W Item_Num
Item_Name Qty Stock B/G Price Amount <BR> ");

//append the 15 items commands
html+=appendItems(html,NORD_ITEMS,ITEM_START);

//seal up html page
appendText(&html,"</PRE></BODY></HTML>");

return OK;
}

/*
*****
** Name           : doNewOrderResults
** Description    :
**                  HTML neworder page entry point
** Parameters     :

```



```

**                               htmlPhraser*   command
block
**                               TXN_HANDLE*   txn handle
struct
** Returns                       :
**                               int - return code
** Comments                       :
**
*****
*/

int doNewOrderResults(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
    DEBUGMSG("Entered doNewOrderResults" << endl);

    char *html=txnHandle->htmlPage;
    struct nord_wrapper *nord = NULL;

    DEBUGMSG("Casting COM txnBuffer to nord struct" <<endl);
    nord = (nord_wrapper*)txnHandle->comInterface.txnBuffer;
    ZeroMemory(nord,maxDataSize);
    DEBUGMSG("COM txnBuffer initialized, validating input
parameters" << endl);

    //set warehouse,district and customer id from command block
    nord->in_nord.s_W_ID = txnHandle->w_id;
    DEBUGMSG("nord w_id:" << nord->in_nord.s_W_ID << endl);

    if( (nord->in_nord.s_D_ID = atoi(commandBlock->get_D_ID()) )
== 0)
    {
        doNewOrderErrorPage(html,ERR_INVALID_D_ID,commandBlock,txnHandle
);
        return OK;
    }
    DEBUGMSG("nord d_id:" << nord->in_nord.s_D_ID << endl);

    if((nord->in_nord.s_C_ID = atoi(commandBlock->get_C_ID()) ) ==
0)
    {
        doNewOrderErrorPage(html,ERR_INVALID_C_ID,commandBlock,txnHandle
);
        return OK;
    }
    DEBUGMSG("nord c_id:" << nord->in_nord.s_C_ID << endl);

    int itemCmd                       = ITEM_START;
    short itemComplete = 0;
    char field[256] = {NULL};

    for (int itemIndex=0;itemIndex<NORD_ITEMS;itemIndex++)
    {
        //supply warehouse
        if( *(commandBlock->get_ITEM_SUPP_W(itemIndex))
)
            if (
(nord->in_nord.in_item[nord->in_nord.s_O_OL_CNT].s_OL_SUPPLY_W_ID
= atoi(commandBlock->get_ITEM_SUPP_W(itemIndex))) == 0)
            {
                doNewOrderErrorPage(html,ERR_INVALID_SUPPLY_W_ID,commandBlock
,txnHandle);
                return OK;
            }
    }
}

```

```

else
    itemComplete++;

//item number
if(
*(commandBlock->get_ITEM_ITEM_NUM(itemIndex))
)
{
    if(itemComplete==1)
    {
        if (
(nord->in_nord.in_item[nord->in_nord.s_O_OL_CNT].s_OL_I_ID =
atoi(commandBlock->get_ITEM_ITEM_NUM(itemIndex))) == 0)
        {
            doNewOrderErrorPage(html,ERR_INVALID_ITEM_NUM,commandBlock,txn
Handle);
            return OK;
        }
        else
            itemComplete++;
    }
    //missing previous value of item supp
warehouse, flag error
else
{
    doNewOrderErrorPage(html,ERR_INVALID_SUPPLY_W_ID,commandBlock
,txnHandle);
    return OK;
}
}
else if( (itemComplete==1) ) //nothing in the command
block, check to see if the previous item value is present
{
    doNewOrderErrorPage(html,ERR_INVALID_ITEM_NUM,commandBlock,txn
Handle);
    return OK;
}

//item qty
if(*(commandBlock->get_ITEM_QTY(itemIndex)))
{
    if(itemComplete==2)
    {
        if(
(nord->in_nord.in_item[nord->in_nord.s_O_OL_CNT].s_OL_QUANTITY =
atoi(commandBlock->get_ITEM_QTY(itemIndex))) == 0)
        {
            doNewOrderErrorPage(html,ERR_INVALID_ITEM_OTY,commandBlock,txn
Handle);
            return OK;
        }
        else
            itemComplete++;
    }
    //missing previous value of item number
else if (itemComplete ==1)
    {
        doNewOrderErrorPage(html,ERR_INVALID_ITEM_NUM,commandBlock,txn
Handle);
        return OK;
    }
    //missing 1st value of supp warehouse
else
    {

```

```

doNewOrderErrorPage(html,ERR_INVALID_SUPPLY_W_ID,commandBlock
,txnHandle);
        return OK;
    }
}
else if(itemComplete==2) //nothing in
the command block, check to see if the previous item values are present
{
doNewOrderErrorPage(html,ERR_INVALID_ITEM_NUM,commandBlock,txn
Handle);
        return OK;
    }
    DEBUGMSG("nord item:" <<
nord->in_nord.s_O_OL_CNT << "SUPPLY_W_ID:" <<
nord->in_nord.in_item[nord->in_nord.s_O_OL_CNT].s_OL_SUPPLY_W_ID
<<
        " OL_I_ID:" <<
nord->in_nord.in_item[nord->in_nord.s_O_OL_CNT].s_OL_I_ID << "
OL_QUANTITY:" <<
nord->in_nord.in_item[nord->in_nord.s_O_OL_CNT].s_OL_QUANTITY
<<endl);

    if(itemComplete == 3)
        nord->in_nord.s_O_OL_CNT++;

    itemComplete=0;
}

DEBUGMSG("complete nord
items:<<nord->in_nord.s_O_OL_CNT<<" initializing remainig unused items "
<<NORD_ITEMS - nord->in_nord.s_O_OL_CNT << " to 0" <<endl);
for(int
itemIndex=nord->in_nord.s_O_OL_CNT;itemIndex<NORD_ITEMS;itemIndex
++)
{
nord->in_nord.in_item[itemIndex].s_OL_SUPPLY_W_ID=0;
nord->in_nord.in_item[itemIndex].s_OL_I_ID = 0;
nord->in_nord.in_item[itemIndex].s_OL_QUANTITY
=0;
}

DEBUGMSG("nord creating new order results html title page"
<<endl);

appendText(&html,"<HTML><HEAD><TITLE>TPC-C New Order
Results</TITLE></HEAD>\r\n"
        "<BODY><FORM
ACTION=""
        APP_NAME
""
METHOD=""GET"">\r\n");
//append menu buttons
html+=appendButtons(html);
html+=appendHiddenFields(html,txnHandle);

appendText(&html,"</FORM><CENTER><H3>New Order</H3>
<BR></CENTER>"
        "<PRE>"
//
//
"123456789012345678901234567890123456789012345678901234567890123
456789012345678901234567890\r\n
        """);

```

```

//assume failure
nord->out_nord.s_transtatus = -1;

DEBUGMSG("nord executing COM interface function" << endl);
HRESULThres;
try
{
    hres =
txnHandle->comInterface.comHandle->doNewOrder(&txnHandle->comInterfa
ce.size,(UCHAR*)&txnHandle->comInterface.txnBuffer);
}
catch(...)
{
    html+=sprintf(html,"ERROR: nord com call caused
exeception to occur.</PRE></BODY></HTML>");
    ERRORMSG("ERROR : nord com call cause exeception
to occur,"<<endl);
    return OK;
}

if(FAILED(hres))
{
    ERRORMSG("ERROR : nord com call failed, rc:" <<
hex << hres);
    DEBUGMSG("ERROR : nord com call failed, rc:" <<
hex << hres);
    return OK;
}

//com call successful, return object back to pool.
hres = txnHandle->comInterface.comHandle->doSetComplete();
if(FAILED(hres))
{
    ERRORMSG("ERROR : nord setcomplete call failed,
rc:" << hex << hres);
    DEBUGMSG("ERROR : nord setcomplete call failed,
rc:" << hex << hres);
}

nord = (nord_wrapper *)txnHandle->comInterface.txnBuffer;
if(FAILED(hres))
{
    html+=sprintf(html,"ERROR: nord com doSetComplete
failed, rc:%ld</PRE></BODY></HTML>",hres);
    ERRORMSG("ERROR : nord com doSetComplete
failed, rc:"<<DEBUGADDRESS(hres)<<endl);
    return OK;
}

DEBUGMSG("nord COM interface function successful,
s_transtatus:" << nord->out_nord.s_transtatus << endl);

int rc = nord->out_nord.s_transtatus;

char buffer[10];
appendText(&html,"Warehouse: ");
appendText(&html,ittoa(nord->in_nord.s_W_ID,buffer,10),6,1);

appendText(&html,"District: ");
appendText(&html,ittoa(nord->in_nord.s_D_ID,buffer,10),26,1);

appendText(&html,"Date: ");
if(rc == OK)
{
    char dateTimeBuffer[50];
copyOutDateTime(dateTimeBuffer,nord->out_nord.s_O_ENTRY_D_time);
appendText(&html,dateTimeBuffer);
}

```

```

    }
    appendText(&html," <BR>"
              "Customer: ");
    appendText(&html,itoa(nord->in_nord.s_C_ID,buffer,10),8,1);
    appendText(&html,"Name: ");

appendText(&html,nord->out_nord.s_C_LAST,LAST_NAME_LEN+3,1);

    appendText(&html,"Credit: ");
    appendText(&html,nord->out_nord.s_C_CREDIT,5,1);

    appendText(&html,"%Disc.: ");
    if(rc == OK)
    {

html+=sprintf(html,"%2.2lf",nord->out_nord.s_C_DISCOUNT/100.0);
    }
    appendText(&html," <BR>"
              "Order Number: ");
    if(rc != INVALID_STATUS)

appendText(&html,itoa(nord->out_nord.s_O_ID,buffer,10),10,1);

    appendText(&html,"Number of Lines: ");

    if(rc != INVALID_STATUS)

appendText(&html,itoa(nord->out_nord.s_O_OL_CNT,buffer,10),10,1);

    appendText(&html,"W_Tax: ");
    if(rc == OK)
    {

html+=sprintf(html,"%5.2lf",nord->out_nord.s_W_TAX/100.0);
    }

    appendText(&html," D_Tax: ");
    if(rc == OK)
    {

html+=sprintf(html,"%5.2lf",nord->out_nord.s_D_TAX/100.0);
    }
    appendText(&html," <BR> <BR>"
              " 1 2 3 4 5 6
// 7 8 9\r\n"
//
"1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890\r\n"
" Supp_W Item_Id Item_Name
Qty Stock B/G Price Amount <BR> ");

    //display items
    if(rc == OK)
    {
        //display valid items
        for(int itemCount=0;itemCount <
nord->out_nord.s_O_OL_CNT;itemCount++)
        {

appendText(&html,itoa(nord->in_nord.in_item[itemCount].s_OL_SUPPLY_W
_ID,buffer,10),8,1);

appendText(&html,itoa(nord->in_nord.in_item[itemCount].s_OL_I_ID,buffer,1
0),10,1);

appendText(&html,nord->out_nord.item[itemCount].s_I_NAME,DEFAULT_S
TRING_LEN+1,1);

```

```

appendText(&html,itoa(nord->in_nord.in_item[itemCount].s_OL_QUANTITY,
buffer,10),5,1);

appendText(&html,itoa(nord->out_nord.item[itemCount].s_S_QUANTITY,buf
fer,10),7,1);

        html+=sprintf(html,"%c $%-7.2lf $%-7.2lf
<BR> ",nord->out_nord.item[itemCount].s_brand_generic,

nord->out_nord.item[itemCount].s_I_PRICE/100.0,

nord->out_nord.item[itemCount].s_OL_AMOUNT/100.0);
    }
    //display blank line for remaining empty items in the
order
        for(int lineBreaks=0;lineBreaks <
(NORD_ITEMS-nord->out_nord.s_O_OL_CNT);lineBreaks++)
            appendText(&html," <BR>");
    }
    else
        appendText(&html," <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>");

        appendText(&html," \r\n <BR> ");

        html+=displayStatus(html,rc);
        if(rc == OK)
            html+=sprintf(html," Total:
$%.2lf",nord->out_nord.s_total_amount/100.0);
        else
            appendText(&html," Total: <BR>");

        appendText(&html,"</PRE></BODY> </HTML>");

        DEBUGMSG("nord html page complete. returning to calling
function" << endl);

        return OK;
    }
}
/*
*****
** Name : doNewOrderErrorPage
** Description :
** Parameters : HTML neworder page entry point
** page char * html result
** error char *
message
** htmlPhraser* command block
** TXN_HANDLE* txn handle
struct
** Returns : int - return code
** Comments :
**
*****
*/
int doNewOrderErrorPage(char *htmlPage,char *message,htmlPhraser
*commandBlock,TXN_HANDLE *txnHandle)
{
    char *html=htmlPage;

```

```

appendText(&html,"<HTML><HEAD><TITLE>TPC-C New
Order</TITLE></HEAD></BODY><FORM
ACTION=""
APP_NAME
"\"
METHOD=""GET"">\r\n"
"<CENTER><H3>Please Fill In New Order Form.</H3></CENTER>\r\n"
"Submit Transaction
<INPUT TYPE=""submit"" NAME=""
CMD_TXN_ID
"\" VALUE=""
CMD_NORD
"\">");
//append the hidden warehouse and district fields
html+=appendHiddenFields(html,txnHandle);
//int buffer for warehouse
char buffer[15];
/*appendText(&html,"<PRE> 1 2 3 4 5
6 7 8 9\r\n"
"123456789012345678901234567890123456789012345678901234567890123
456789012345678901234567890\r\n"
"Warehouse: ");*/
appendText(&html,"<PRE>Warehouse: ");
appendText(&html,itoa(txnHandle->w_id,buffer,10),7,1);
appendText(&html,"District: <INPUT NAME=""
CMD_D_ID
"\" SIZE=1>
Date:<BR>"
"Customer <INPUT NAME=""
CMD_C_ID
"\" SIZE=6> Name:
Credit: %Disc.:<BR>"
"Order Number:
Number of Lines: W_tax: D_tax:<BR><BR>"
// " 1 2 3
4 5 6 7 8 9\r\n"
// "12345678901234567890123456789012345678901234567890123456789012
3456789012345678901234567890\r\n"
" Supp_W Item_Num
Item_Name Qty Stock B/G Price Amount <BR> ");
//append the 15 items commands
html+=appendItems(html,NORD_ITEMS,ITEM_START);
appendText(&html,message);
//seal up html page
appendText(&html,"</PRE></BODY></HTML>");
return OK;
}
/*
*****
** Name : doPaymentForm
** Description :
** Parameters : HTML payment page entry point
** block htmlPhraser* command
** TXN_HANDLE* txn handle
** struct

```

```

** Returns : int - return code
** Comments :
**
*****
*/
int doPaymentForm(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
char *html=txnHandle->htmlPage;
appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD></BODY><FORM
ACTION=""
APP_NAME
"\"
METHOD=""GET"">\r\n"
"<CENTER><H3>Please Fill In Payment Form.</H3></CENTER><BR>\r\n"
"Submit Transaction
CMD_TXN_ID
"\" VALUE=""
CMD_PYMT
"\">");
html+=appendHiddenFields(html,txnHandle);
appendText(&html,"<BR><PRE>\r\n"
"Date:<BR>"
"Warehouse: ");
char buffer[15];
appendText(&html,itoa(txnHandle->w_id,buffer,10));
appendSpaces(&html,10);
appendText(&html,"District: <INPUT NAME=""
CMD_D_ID
"\" SIZE=1>\r\n<BR>"
"<BR> <BR> <BR>"
"Customer: "
"<INPUT NAME=""
CMD_C_ID
"\" SIZE=5>"
" "
"Cust-Warehouse: "
"<INPUT NAME=""
CMD_C_W_ID
"\" SIZE=5>"
" "
"Cust-District: "
"<INPUT NAME=""
CMD_C_D_ID
"\" SIZE=1><BR>"
"Name: <INPUT
NAME=""
CMD_C_NAME
"\" SIZE=20>");
appendText(&html," Since: <BR>"
" "
" "
Credit: <BR>"
" "
" "
%Disc: <BR>"
"Amount Paid: "
"<INPUT NAME=""
CMD_AMT_PAID
"\" SIZE=10>"

```

```

"New
Cust-Balance:<BR>"
"Credit Limit:<BR>
<BR>Cust-Data:<BR> <BR> <BR> <BR> </PRE>");
return OK;
}
/*
*****
** Name : doPaymentResults
** Description :
** HTML neworder page entry point
** Parameters :
** htmlPhraser* command
block
** TXN_HANDLE* txn handle
struct
** Returns : int - return code
** Comments :
*****
*/
int doPaymentResults(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
char *html=txnHandle->htmlPage;
char buffer[50];

struct paym_wrapper *pymt = NULL;
pymt = (paym_wrapper*)txnHandle->comInterface.txnBuffer;
ZeroMemory(pymt,maxDataSize);

//set login warehouse id from command block
pymt->in_paym.s_W_ID = txnHandle->w_id;

//set district from command block
if( (pymt->in_paym.s_D_ID = atoi(commandBlock->get_D_ID()))
== 0)
{
doPaymentErrorPage(html,ERR_INVALID_D_ID,commandBlock,txnHandle);
return OK;
}

//set customer id from command block
if( (pymt->in_paym.s_C_ID = atoi(commandBlock->get_C_ID()))
== 0)
{
if(*(commandBlock->get_C_NAME()) == NULL)
{
//no customer id nor customer last name
specified.

doPaymentErrorPage(html,ERR_MISSING_C_ID_OR_CLAST,commandBloc
k,txnHandle);
return OK;
}
else
strcpy(pymt->in_paym.s_C_LAST,commandBlock->get_C_NAME());
}
else
{
//make sure that the user only inserted just c_id

```

```

if(*(commandBlock->get_C_NAME()) != NULL)
{
doPaymentErrorPage(html,ERR_C_ID_OR_CLAST_ONLY,commandBlock,tx
nHandle);
return OK;
}
}
//get customer warehose id field
if( (pymt->in_paym.s_C_W_ID =
atoi(commandBlock->get_C_W_ID())) == 0)
{
doPaymentErrorPage(html,ERR_INVALID_C_W_ID,commandBlock,txnHand
le);
return OK;
}
//get customer district id field
if( ( (pymt->in_paym.s_C_D_ID =
atoi(commandBlock->get_C_D_ID())) == 0)
{
doPaymentErrorPage(html,ERR_INVALID_C_D_ID,commandBlock,txnHandl
e);
return OK;
}
}
if(!copyInMoney64(commandBlock->get_AMT_PAID(),&pymt->in_paym.s_H
_AMOUNT))
{
doPaymentErrorPage(html,ERR_INVALID_PAYMENT_AMOUNT,command
Block,txnHandle);
return OK;
}
}
appendText(&html,"<HTML><HEAD><TITLE>TPC-C Payment
Results</TITLE></HEAD>\r\n"
" <BODY><FORM
ACTION=\"\"
APP_NAME
\"
METHOD=\"GET\">\r\n");
html+=appendButtons(html);
html+=appendHiddenFields(html,txnHandle);
appendText(&html,"</FORM><CENTER><H3>Payment</H3></CENTER>")
;
DEBUGMSG("Calling com entry api payment,
w_id:<<pymt->in_paym.s_W_ID<<"
d_id:<<pymt->in_paym.s_D_ID<<endl);

//assume failure
pymt->out_paym.s_transtatus = -1;
HRESULThres;
try
{
hres =
txnHandle->comInterface.comHandle->doPayment(&txnHandle->comInterface
.size,(UCHAR*)&txnHandle->comInterface.txnBuffer);
}
catch(...)

```

```

    {
        html+=sprintf(html,"ERROR: Com Payment call caused
exeception to occur.</PRE></BODY></HTML>");
        ERRORMSG("ERROR : Com Payment call caused
exeception to occur."<<endl);
        return OK;
    }

    if(FAILED(hres))
    {
        html+=sprintf(html,"ERROR: pymt com call failed,
rc:%x</PRE></BODY></HTML>",hres);
        ERRORMSG("ERROR : pymt com call failed,
rc:"<<hres<<endl);
        return OK;
    }

    hres = txnHandle->comInterface.comHandle->doSetComplete();
    if(FAILED(hres))
    {
        html+=sprintf(html,"ERROR: pymt com doSetComplete
failed, rc:%ld</PRE></BODY></HTML>",hres);
        ERRORMSG("ERROR : pymt com doSetComplete
failed, rc:"<<DEBUGADDRESS(hres)<<endl);
        return OK;
    }
    pymt = (pymt_wrapper *)txnHandle->comInterface.txnBuffer;
    //get return code
    int rc = pymt->out_paym.s_transtatus;
    if( rc != OK)
    {
        html+=displayStatus(html,rc);
        appendText(&html,"</PRE></BODY></HTML>");
        ERRORMSG("Payment TXN ERROR"<<endl

<<"pymt->in_paym.s_C_D_ID:"<<pymt->in_paym.s_C_D_ID<<endl
<<"pymt->in_paym.s_C_ID:"<<pymt->in_paym.s_C_ID<<endl
<<"pymt->in_paym.s_C_LAST:"<<pymt->in_paym.s_C_LAST<<endl
<<"pymt->in_paym.s_C_W_ID:"<<pymt->in_paym.s_C_W_ID<<endl
<<"pymt->in_paym.s_D_ID:"<<pymt->in_paym.s_D_ID<<endl
<<"pymt->in_paym.s_H_AMOUNT:"<<pymt->in_paym.s_H_AMOUNT<<en
dl
<<"pymt->in_paym.s_H_DATE_time:"<<pymt->in_paym.s_H_DATE_time<<
endl
<<"pymt->in_paym.s_W_ID:"<<pymt->in_paym.s_W_ID<<endl
<<"pymt->out_paym.deadlocks:"<<pymt->out_paym.deadlocks<<endl
<<"pymt->out_paym.s_C_BALANCE:"<<pymt->out_paym.s_C_BALANCE<
<endl
<<"pymt->out_paym.s_C_CITY:"<<pymt->out_paym.s_C_CITY<<endl
<<"pymt->out_paym.s_C_CREDIT:"<<pymt->out_paym.s_C_CREDIT<<endl
<<"pymt->out_paym.s_C_CREDIT_LIM:"<<pymt->out_paym.s_C_CREDIT
LIM<<endl
<<"pymt->out_paym.s_C_DATA:"<<pymt->out_paym.s_C_DATA<<endl

<<"pymt->out_paym.s_C_DISCOUNT:"<<pymt->out_paym.s_C_DISCOUNT
<<endl
<<"pymt->out_paym.s_C_FIRST:"<<pymt->out_paym.s_C_FIRST<<endl
<<"pymt->out_paym.s_C_ID:"<<pymt->out_paym.s_C_ID<<endl
<<"pymt->out_paym.s_C_LAST:"<<pymt->out_paym.s_C_LAST<<endl
<<"pymt->out_paym.s_C_MIDDLE:"<<pymt->out_paym.s_C_MIDDLE<<endl
1
<<"pymt->out_paym.s_C_PHONE:"<<pymt->out_paym.s_C_PHONE<<endl
<<"pymt->out_paym.s_C_SINCE_time:"<<pymt->out_paym.s_C_SINCE_time
<<endl
<<"pymt->out_paym.s_C_STATE:"<<pymt->out_paym.s_C_STATE<<endl
<<"pymt->out_paym.s_C_STREET_1:"<<pymt->out_paym.s_C_STREET_1<
<endl
<<"pymt->out_paym.s_C_STREET_2:"<<pymt->out_paym.s_C_STREET_2<
<endl
<<"pymt->out_paym.s_C_ZIP:"<<pymt->out_paym.s_C_ZIP<<endl
<<"pymt->out_paym.s_D_CITY:"<<pymt->out_paym.s_D_CITY<<endl
<<"pymt->out_paym.s_D_STATE:"<<pymt->out_paym.s_D_STATE<<endl
<<"pymt->out_paym.s_D_STREET_1:"<<pymt->out_paym.s_D_STREET_1<
<endl
<<"pymt->out_paym.s_D_STREET_2:"<<pymt->out_paym.s_D_STREET_2<
<endl
<<"pymt->out_paym.s_D_ZIP:"<<pymt->out_paym.s_D_ZIP<<endl
<<"pymt->out_paym.s_H_DATE_time:"<<pymt->out_paym.s_H_DATE_time
<<endl
<<"pymt->out_paym.s_transtatus:"<<pymt->out_paym.s_transtatus<<endl
<<"pymt->out_paym.s_W_CITY:"<<pymt->out_paym.s_W_CITY<<endl
<<"pymt->out_paym.s_W_STATE:"<<pymt->out_paym.s_W_STATE<<endl
<<"pymt->out_paym.s_W_STREET_1:"<<pymt->out_paym.s_W_STREET_1
<<endl
<<"pymt->out_paym.s_W_STREET_2:"<<pymt->out_paym.s_W_STREET_2
<<endl
<<"pymt->out_paym.s_W_ZIP:"<<pymt->out_paym.s_W_ZIP<<endl);

        return OK;
    }
    //      appendText(&html, "<BR><PRE>\r\n");
    //      appendText(&html, "      1      2      3      4      5      6      7
      8<BR>");
    //
    appendText(&html,"12345678901234567890123456789012345678901234567
890123456789012345678901234567890<BR>");

    //start creating result body
    appendText(&html, "<BR><PRE>\r\n"

```

```

        "Date: ");

copyOutDateTime(buffer,pymt->out_paym.s_H_DATE_time);
appendText(&html,buffer);

appendText(&html, "<BR>"
"Warehouse: ");

appendText(&html,itoa(pymt->in_paym.s_W_ID,buffer,10),6+24,1);
appendText(&html,"District: ");
appendText(&html,itoa(pymt->in_paym.s_D_ID,buffer,10),2,1);
appendText(&html, "<BR>");

//print out warehouse and district information

appendText(&html,pymt->out_paym.s_W_STREET_1,STREET_LEN+21,1);

appendText(&html,pymt->out_paym.s_D_STREET_1,STREET_LEN,1);
appendText(&html, "<BR>");

appendText(&html,pymt->out_paym.s_W_STREET_2,STREET_LEN+21,1);

appendText(&html,pymt->out_paym.s_D_STREET_2,STREET_LEN,1);
appendText(&html, "<BR>");

appendText(&html,pymt->out_paym.s_W_CITY,CITY_LEN+1,1);

appendText(&html,pymt->out_paym.s_W_STATE,STATE_LEN+1,1);
copyOutZip(buffer,pymt->out_paym.s_W_ZIP);
appendText(&html,buffer);

appendText(&html,pymt->out_paym.s_D_CITY,CITY_LEN+1,1);

appendText(&html,pymt->out_paym.s_D_STATE,STATE_LEN+1,1);
copyOutZip(buffer,pymt->out_paym.s_D_ZIP);
appendText(&html,buffer);

//print out customer information
appendText(&html, "<BR> <BR>Customer: ");
appendText(&html,itoa(pymt->out_paym.s_C_ID,buffer,10),5+1,1);

appendText(&html, "Cust-Warehouse: ");

appendText(&html,itoa(pymt->in_paym.s_C_W_ID,buffer,10),6+1,1);

appendText(&html, "Cust-District: ");
appendText(&html,itoa(pymt->in_paym.s_C_D_ID,buffer,10));

//add customer information
appendText(&html, "<BR>Name: ");

appendText(&html,pymt->out_paym.s_C_FIRST,FIRST_NAME_LEN+1,1);

appendText(&html,pymt->out_paym.s_C_MIDDLE,INITIALS_LEN+1,1);
DEBUGMSG("Last name:"<<pymt->out_paym.s_C_LAST<<endl);

appendText(&html,pymt->out_paym.s_C_LAST,LAST_NAME_LEN+5,1);

appendText(&html, "Since: ");
copyOutDateTime(buffer,pymt->out_paym.s_C_SINCE_time);
appendText(&html,buffer);

appendText(&html, "<BR>");
appendSpaces(&html,8);

appendText(&html,pymt->out_paym.s_C_STREET_1,STREET_LEN+20,1);

```

```

appendText(&html, "Credit: ");
appendText(&html,pymt->out_paym.s_C_CREDIT);

appendText(&html, "<BR>");
appendSpaces(&html,8);

appendText(&html,pymt->out_paym.s_C_STREET_2,STREET_LEN+21,1);
appendText(&html, "%Disc: ");

html+=sprintf(html, "%2.2lf", pymt->out_paym.s_C_DISCOUNT/100.0);

appendText(&html, "<BR>");
appendSpaces(&html,8);

appendText(&html,pymt->out_paym.s_C_CITY,CITY_LEN+1,1);

appendText(&html,pymt->out_paym.s_C_STATE,STATE_LEN+1,1);

copyOutZip(buffer,pymt->out_paym.s_C_ZIP);
appendText(&html,buffer,15,1);

appendText(&html, "Phone: ");
copyOutPhone(buffer,pymt->out_paym.s_C_PHONE);
appendText(&html,buffer);

appendText(&html, "<BR> <BR>Amount Paid: $");

html+=sprintf(html, "%-9.2lf", pymt->in_paym.s_H_AMOUNT/100.0);

appendText(&html, "New Cust-Balance: $");

html+=sprintf(html, "%-9.2lf", pymt->out_paym.s_C_BALANCE/100.0);

appendText(&html, "<BR>Credit Limit: $");

html+=sprintf(html, "%-9.2lf", pymt->out_paym.s_C_CREDIT_LIM/100.0);

appendText(&html, "<BR> <BR>Cust-Data: ");
if(pymt->out_paym.s_C_CREDIT[0] == 'B' &&
pymt->out_paym.s_C_CREDIT[1] == 'C')
{
appendCustData(&html,pymt->out_paym.s_C_DATA);
appendText(&html, "<BR>");
}
else
appendText(&html, "<BR> <BR> <BR>");

html+=displayStatus(html,rc);
appendText(&html, "</PRE></BODY></HTML>");

return OK;
}
/*
*****
** Name : doPaymentErrorPage
** Description :
** Parameters :
** char * html page
result
** char * error
message
** htmlPhraser * command block
** TXN_HANDLE* txn handle
struct
** Returns :

```

```

**                                     int - return code
** Comments                               :
**
*****
*/

int doPaymentErrorPage(char *htmlPage,char *message,htmlPhraser
*commandBlock, TXN_HANDLE *txnHandle)
{
    char *html=htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD>\r\n"
ACTION="\
APP_NAME
"
METHOD="\GET\r\n"
"<CENTER><H3>Please Fill In Payment Form.</H3></CENTER> <BR>\r\n"
"Submit Transaction
<INPUT TYPE=\submit" NAME=\
CMD_TXN_ID
"\" VALUE=\
CMD_PYMT
"\">");
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"<BR><PRE>\r\n"
"Date:<BR>"
"Warehouse: ");
    char buffer[15];
    appendText(&html,itoa(txnHandle->w_id,buffer,10));
    appendSpaces(&html,10);
    appendText(&html,"District: <INPUT NAME=\
CMD_D_ID
"\" SIZE=1>\r\n<BR>"
"<BR> <BR> <BR>"
"Customer: "
"<INPUT NAME=\
CMD_C_ID
"\" SIZE=5>"
" "
"Cust-Warehouse: "
"<INPUT NAME=\
CMD_C_W_ID
"\" SIZE=6>"
" "
"Cust-District: "
"<INPUT NAME=\
CMD_C_D_ID
"\" SIZE=1><BR>"
"Name: <INPUT
NAME=\
CMD_C_NAME
"\" SIZE=20>");
    appendText(&html,"
Since: <BR>"
"
"
"
"
"Amount Paid: "
"<INPUT NAME=\
CMD_AMT_PAID
"\" SIZE=10>"

```

```

"
"
"New
"Credit Limit:<BR>
<BR> <BR> Cust-Data:<BR> <BR> <BR> <BR> ");
    appendText(&html,message);
    appendText(&html,"</PRE>");
    return OK;
}
/*
*****
** Name                               : doOrderStatusForm
** Description                           :
**                                     HTML orderStatus page entry
point
** Parameters                             :
**                                     htmlPhraser*      command
block
**                                     TXN_HANDLE*      txn handle
struct
** Returns                                 :
**                                     int - return code
** Comments                               :
**
*****
*/

int doOrderStatusForm(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
    char *html=txnHandle->htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Order
Status</TITLE></HEAD>\r\n"
"<BODY><FORM
ACTION="\
APP_NAME
"
METHOD="\GET\r\n"
"<CENTER><H3>Please Fill In Order Status Form.</H3></CENTER>
<BR>\r\n"
"Submit Transaction
<INPUT TYPE=\submit" NAME=\
CMD_TXN_ID
"\" VALUE=\
CMD_ORDS
"\">"
"<BR> ");
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"<PRE>\r\n"
"Warehouse: ");
    char buffer[15];
    appendText(&html,itoa(txnHandle->w_id,buffer,10));
    appendText(&html,"
District: <INPUT NAME=\
CMD_D_ID
"\" SIZE=1>\r\n<BR>"
"Customer: "
"<INPUT NAME=\
CMD_C_ID
"\" SIZE=5>"
" "
"Name: "
"<INPUT NAME=\

```



```

                                CMD_C_NAME
                                "\ " SIZE=20><BR>"
                                "Cust-Balance: <BR>"
                                "Order-Number:
Entry-Date:          Carrier-Number<BR>"
                                "Supply-W
Item-Num  Qty    Amount    Delivery<BR></PRE>";
                                appendText(&html,"</BODY><</HTML>");
                                return OK;
}
/*
*****
** Name          : doOrderStatusResults
** Description    :
**              HTML orderStatus page entry
point
** Parameters    :
**              htmlPhraser*      command
block
**              char *           html result
page
** Returns      :
**              int - return code
** Comments     :
**
*****
*/
int doOrderStatusResults(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
    char *html=txnHandle->htmlPage;
    struct ords_wrapper *ords = NULL;
    ords = (ords_wrapper *) txnHandle->comInterface.txnBuffer;
    ZeroMemory(ords,maxDataSize);

    //set warehouse login id from command blk
    ords->in_ords.s_W_ID = txnHandle->w_id;

    //set district login id from command blk
    if( ords->in_ords.s_D_ID = atoi(commandBlock->get_D_ID())) ==
0)
    {
doOrderStatusErrorPage(html,ERR_INVALID_D_ID,commandBlock,txnHandl
e);
        return OK;
    }

    if( ords->in_ords.s_C_ID = atoi(commandBlock->get_C_ID())) ==
0)
    {
        if(*(commandBlock->get_C_NAME()) == NULL)
        {
            //no customer id nor customer last name
            specified.
doOrderStatusErrorPage(html,ERR_MISSING_C_ID_OR_CLAST,commandBl
ock,txnHandle);
                return OK;
            }
        else
strepy(ords->in_ords.s_C_LAST,commandBlock->get_C_NAME());
    }
}

```

```

else
{
    //make sure that the user only inserted just c_id
    if(*(commandBlock->get_C_NAME()) != NULL)
    {
doOrderStatusErrorPage(html,ERR_C_ID_OR_CLAST_ONLY,commandBloc
k,txnHandle);
        return OK;
    }

    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Order
Status Results</TITLE></HEAD>\r\n"
                                "<BODY><FORM
ACTION=\\""
                                APP_NAME
                                "\"
METHOD=\\"GET\ ">\r\n");
    html+=appendButtons(html);
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"</FORM>");
    ords->out_ords.s_transtatus = -1;
    HRESULThres;
    try
    {
        hres =
txnHandle->comInterface.comHandle->doOrderStatus(&txnHandle->comInterf
ace.size,(UCHAR**)&txnHandle->comInterface.txnBuffer);
    }
    catch(...)
    {
        html+=sprintf(html,"ERROR: ords com call caused
exeception.</PRE></BODY><</HTML>");
        return OK;
    }

    if(FAILED(hres))
    {
        html+=sprintf(html,"ERROR: ords com call failed,
rc:%x</PRE></BODY><</HTML>",hres);
        ERRORMSG("ERROR : ords com call failed,
rc:"<<DEBUGADDRESS(hres));
        return OK;
    }

    hres = txnHandle->comInterface.comHandle->doSetComplete();
    if(FAILED(hres))
    {
        html+=sprintf(html,"ERROR: ords com doSetComplete
failed,rc:%ld</PRE></BODY><</HTML>",hres);
        ERRORMSG("ERROR : ords com doSetComplete failed,
rc:"<<DEBUGADDRESS(hres)<<endl);
        return OK;
    }

    ords = (ords_wrapper *)txnHandle->comInterface.txnBuffer;
    int rc = ords->out_ords.s_transtatus;
    if( rc != OK)
    {
        html+=displayStatus(html,rc);
        appendText(&html,"</PRE></BODY><</HTML>");
        ERRORMSG("ERROR order status"<<endl

```

```

<<"ords->in_ords.s_C_ID:"<<ords->in_ords.s_C_ID<<endl
<<"ords->in_ords.s_C_LAST:"<<ords->in_ords.s_C_LAST<<endl
<<"ords->in_ords.s_D_ID:"<<ords->in_ords.s_D_ID<<endl
<<"ords->in_ords.s_W_ID:"<<ords->in_ords.s_W_ID<<endl
<<"ords->out_ords.deadlocks:"<<ords->out_ords.deadlocks<<endl
<<"ords->out_ords.s_C_BALANCE:"<<ords->out_ords.s_C_BALANCE<<en
dl
<<"ords->out_ords.s_C_FIRST:"<<ords->out_ords.s_C_FIRST<<endl
<<"ords->out_ords.s_C_ID:"<<ords->out_ords.s_C_ID<<endl
<<"ords->out_ords.s_C_ID:"<<ords->out_ords.s_C_ID<<endl
<<"ords->out_ords.s_C_MIDDLE:"<<ords->out_ords.s_C_MIDDLE<<endl
<<"ords->out_ords.s_O_CARRIER_ID:"<<ords->out_ords.s_O_CARRIER_ID
<<endl
<<"ords->out_ords.s_O_ENTRY_D_time:"<<ords->out_ords.s_O_ENTRY_D_
time<<endl
<<"ords->out_ords.s_O_ID:"<<ords->out_ords.s_O_ID<<endl
<<"ords->out_ords.s_ol_cnt:"<<ords->out_ords.s_ol_cnt<<endl);

        return OK;
    }

    //start creating result body
appendText(&html,"</FORM><CENTER><H3>Order-Status</H3></CENTE
R>");
    appendText(&html,"<BR><PRE>\r\nWarehouse: ");
    char buffer[50];

    appendText(&html,itoa(ords->in_ords.s_W_ID,buffer,10),6+1,1);
    appendText(&html,"District: ");
    appendText(&html,itoa(ords->in_ords.s_D_ID,buffer,10));
    appendText(&html,"<BR>"

                "Customer: ");

    //get customer id
    appendText(&html,itoa(ords->in_ords.s_C_ID,buffer,10),6+1,1);
    appendText(&html,"Name: ");
    //get first, middle, and last from wrapper
appendText(&html,ords->out_ords.s_C_FIRST,FIRST_NAME_LEN+1,1);
appendText(&html,ords->out_ords.s_C_MIDDLE,INITIALS_LEN+1,1);
appendText(&html,ords->out_ords.s_C_LAST,LAST_NAME_LEN+5,1);

    //get customer balance from wrapper
    appendText(&html,"r\nCust-Balance: $");
    html+=sprintf(html,"% .2lf",ords->out_ords.s_C_BALANCE/100.0);

    //display order number, entry date, and carrier number
    appendText(&html,"<BR> <BR>"

                "Order-Number ");
    appendText(&html,itoa(ords->out_ords.s_O_ID,buffer,10),12,1);
    appendText(&html,"Entry-Date: ");
    copyOutDateTme(buffer,ords->out_ords.s_O_ENTRY_D_time);

```

```

        appendText(&html,buffer,22,1);

        appendText(&html,"Carrier-Number: ");

appendText(&html,itoa(ords->out_ords.s_O_CARRIER_ID,buffer,10));

        //add item title columns
        appendText(&html,"<BR>"

                "Supply-W "
                "Item-Id "
                "Qty "
                "Amount "
                "Delivery-Date<BR>"

");

        //display items
        for (int
itemCount=0;itemCount<ords->out_ords.s_ol_cnt;itemCount++)
        {
        //appendSpaces(&html,2);

                //get supp w

appendText(&html,itoa(ords->out_ords.item[itemCount].s_OL_SUPPLY_W_I
D,buffer,10),11,1);

                //get item num

appendText(&html,itoa(ords->out_ords.item[itemCount].s_OL_I_ID,buffer,10),
11,1);

                //get item oty

appendText(&html,itoa(ords->out_ords.item[itemCount].s_OL_QUANTITY,bu
ffer,10),6,1);

                //get item dollor amount

html+=sprintf(html,"%-14.2lf",ords->out_ords.item[itemCount].s_OL_AMOU
NT/100.0);

                //get delivery date

copyOutDate(buffer,ords->out_ords.item[itemCount].s_OL_DELIVERY_D_ti
me);

                appendText(&html,buffer);
                appendText(&html," <BR> ");

        }

        //append line breaks if item count is less than 15
        for (int itemCount=0;itemCount <
(15-ords->out_ords.s_ol_cnt);itemCount++)
            appendText(&html,"<BR> ");

        html+=displayStatus(html,rc);

        appendText(&html,"</PRE></BODY></HTML>");

        return OK;
    }

/*
*****
** Name          : doOrderStatusErrorPage
** Description   :
**               HTML orderStatus error page
** Parameters    :

```

```

**                               char *          html page
result
**                               char *          error
message
**                               htmlPhraser*    command
block
**                               TXN_HANDLE*     txn handle
** Returns      :
**                               int - return code
** Comments
**
*****
*/

int doOrderStatusErrorPage(char *htmlPage,char *message,htmlPhraser
*commandBlock,TXN_HANDLE *txnHandle)
{
    char *html=htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Order
Status</TITLE></HEAD>\r\n"
    "ACTION=\\"
    APP_NAME
    METHOD=\\"GET\\>\r\n"
    "<CENTER><H3>Please Fill In Order Status Form.</H3></CENTER>
<BR>\r\n"
    "Submit Transaction
    <INPUT TYPE=\\"submit\\" NAME=\\"
    CMD_TXN_ID
    "\\" VALUE=\\"
    CMD_ORDS
    "\\">"
    "<BR> ");
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"<PRE>\r\n"
    "Warehouse: ");
    char buffer[15];
    appendText(&html,itoa(txnHandle->w_id,buffer,10));
    appendText(&html," District: <INPUT NAME=\\"
    CMD_D_ID
    "\\" SIZE=1>\r\n<BR>"
    "Customer: "
    "<INPUT NAME=\\"
    CMD_C_ID
    "\\" SIZE=5>"
    " "
    "Name: "
    "<INPUT NAME=\\"
    CMD_C_NAME
    "\\" SIZE=20><BR>"
    "Cust-Balance: <BR>"
    "Order-Number:
    Entry-Date:      Carrier-Number<BR>"
    "Supply-W
    Item-Num  Qty   Amount   Delivery <BR>");
    appendText(&html,message);
    appendText(&html,"</PRE></BODY></HTML>");
    return OK;
}
/*

*****
** Name      : doDeliveryForm
** Description :
**           HTML payment page entry point
** Parameters :
**           htmlPhraser*    command
**           TXN_HANDLE*     txn handle
block
** Returns   :
**           int - return code
** Comments
**
*****
*/

int doDeliveryForm(htmlPhraser *commandBlock,TXN_HANDLE
*txnHandle)
{
    char *html=txnHandle->htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD>\r\n"
    "ACTION=\\"
    APP_NAME
    METHOD=\\"GET\\>\r\n"
    "<CENTER><H3>Delivery.</H3></CENTER>\r\n"
    "Submit Transaction
    <INPUT TYPE=\\"submit\\" NAME=\\"
    CMD_TXN_ID
    "\\" VALUE=\\"
    CMD_DLVY
    "\\">");
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"<BR> <PRE>"
    "Warehouse: ");
    char buffer[10];
    appendText(&html,itoa(txnHandle->w_id,buffer,10));
    appendText(&html," <BR> <BR>"
    "Carrier Number: "
    "<INPUT NAME=\\"
    CMD_CARRIER_NUM
    "\\" SIZE=1>"
    "</FORM></PRE>");
    appendText(&html,"</BODY></HTML>");
    return OK;
}
/*

*****
** Name      : doDeliveryResults
** Description :
**           HTML payment page entry point
** Parameters :
**           htmlPhraser*    command
**           TXN_HANDLE*     txn handle
block
** Returns   :
**           int - return code
** Comments
**
*****

```

```

*****
*/
int doDeliveryResults(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
    char *html = txnHandle->htmlPage;

    //declare delivery structure
    struct dlvy_wrapper          dlvy;

    //set warehouse login id from command blk
    dlvy.in_dlvy.s_W_ID = txnHandle->w_id;

    //set the carrier id from command blk
    if( dlvy.in_dlvy.s_O_CARRIER_ID =
atoi(commandBlock->get_CARRIER_NUM()) == 0)
    {

doDeliveryErrorPage(html, ERR_INVALID_CARRIER, commandBlock, txnHan
dle);
        return OK;
    }

    //print title, add hidden fields , txn buttons
    appendText(&html, "<HTML><HEAD><TITLE>TPC-C Delivery
Results</TITLE></HEAD>\r\n<BODY><FORM ACTION=\"\"
APP_NAME
METHOD=\"GET\">\r\n");

    html+=appendButtons(html);

    html+=appendHiddenFields(html, txnHandle);

    appendText(&html,
"<FORM><CENTER><H3>Delivery</H3></CENTER>");

    int rc =
queueDlvyTxn(dlvy.in_dlvy.s_W_ID, dlvy.in_dlvy.s_O_CARRIER_ID);
    if( rc != OK)
    {
        html+=displayStatus(html, rc);
        appendText(&html, "</PRE></BODY></HTML>\r\n");

        ERRORMSG("ERROR: Unable to queue dlvy txn,
rc:"<<rc<<endl);
        return OK;
    }

    //start creating result body
    appendText(&html, "Warehouse: ");

    //get w_id from wrapper
    char buffer[15];
    appendText(&html, itoa(dlvy.in_dlvy.s_W_ID, buffer, 10));
    appendText(&html, "<BR> <BR>Carrier Number: ");

    //get carrier_id from wrapper

    appendText(&html, itoa(dlvy.in_dlvy.s_O_CARRIER_ID, buffer, 10));
    appendText(&html, "<BR> <BR>Execution Status: Delivery has
been queued </PRE></BODY></HTML>");

    return OK;
}
*/

```

```

*****
** Name          : doDeliveryErrorPage
** Description    :
**               : HTML payment error page entry
point
** Parameters     :
**               : char *          html result
page
**               : char *          error
message
**               : htmlPhraser     command
block
**               : TXN_HANDLE*    txn handle
**
** Returns       :
**               : int - return code
** Comments      :
**
*****
*/
int doDeliveryErrorPage(char *htmlPage, char *message, htmlPhraser
*commandBlock, TXN_HANDLE *txnHandle)
{
    char *html=htmlPage;

    appendText(&html, "<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD>\r\n"
ACTION=\"\"
APP_NAME
METHOD=\"GET\">\r\n"
"<CENTER><H3>Delivery.</H3></CENTER>\r\n"
"Submit Transaction
<INPUT TYPE=\"submit\" NAME=\"\"
CMD_TXN_ID
VALUE=\"\"
CMD_DLvy
\">");
    html+=appendHiddenFields(html, txnHandle);

    appendText(&html, "<BR> <PRE>"
"Warehouse: ");
    char buffer[15];
    appendText(&html, itoa(txnHandle->w_id, buffer, 10));

    appendText(&html, "<BR> <BR>"
"Carrier Number: "
"<INPUT NAME=\"\"
CMD_CARRIER_NUM
\" SIZE=1> <BR>");

    appendText(&html, message);
    appendText(&html, "</PRE></BODY></HTML>");

    return OK;
}
/*
*****
** Name          : doStockForm
** Description    :
**               : HTML stock page entry point
** Parameters     :

```

```

**                               htmlPhraser    command
block
**                               TXN_HANDLE*    txn handle
** Returns                        :
**                               int - return code
** Comments                       :
**
*****
*/

int doStockForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    char *html=txnHandle->htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Stock
Level</TITLE></HEAD>\r\n"
                                "<BODY><FORM
ACTION=\"\"
                                APP_NAME
                                \"\"
METHOD=\"GET\">\r\n"
"<CENTER><H3>Please Fill In Stock Form.</H3></CENTER> <BR>\r\n"
"Submit Transaction
<INPUT TYPE=\"submit\" NAME=\"\"
                                CMD_TXN_ID
                                \"\" VALUE=\"\"
                                CMD_STOK
                                \"\">");
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"<PRE>"
                                "Warehouse: ");
    char buffer[15];
    appendText(&html,itoa(txnHandle->w_id,buffer,10),6+1,1);
    appendText(&html,"District: ");
    appendText(&html,itoa(txnHandle->d_id,buffer,10));
    appendText(&html," <BR> <BR>"
                                "Stock Level
Threshold: "
                                "<INPUT NAME=\"\"
                                \"\" SIZE=1> <BR>
                                \"Low Stock: <BR>"
                                "</PRE>");
    appendText(&html,"</FORM></BODY></HTML>");
    return OK;
}
/*
*****
** Name                          : doStockResults
** Description                    :
**                               HTML stock page entry point
** Parameters                    :
**                               htmlPhraser*    command
block
**                               TXN_HANDLE*    txn handle
struct
** Returns                      :
**                               int - return code
** Comments                      :
**
*****
*/
int doStockResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    char *html = txnHandle->htmlPage;

    struct stok_wrapper *stok;
    stok = (stok_wrapper*)txnHandle->comInterface.txnBuffer;
    ZeroMemory(stok,maxDataSize);

    //set warehouse login id from command blk
    stok->in_stok.s_W_ID = txnHandle->w_id;

    //set district login id from command blk
    stok->in_stok.s_D_ID = txnHandle->d_id;

    //set stock level threshold id from command blk
    if (stok->in_stok.s_threshold =
atoi(commandBlock->get_STK_THRESHOLD()) == 0)
    {
        doStockErrorPage(html,ERR_INVALID_THRESHOLD,commandBlock,txnHa
ndle);
        return OK;
    }
    //assume failure, set s_transtatus to err
    stok->out_stok.s_transtatus = INVALID_STATUS;

    //print title, add hidden fields , txn buttons
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Stock
Level Results</TITLE></HEAD>\r\n"
                                "<BODY><FORM
ACTION=\"\"
                                APP_NAME
                                \"\"
METHOD=\"GET\">\r\n");
    html+=appendButtons(html);
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"</FORM>");
    stok->out_stok.s_transtatus = -1;
    DEBUGMSG("Calling com entry api for stock call,
w_id:<<stok->in_stok.s_W_ID<< d_id:<<stok->in_stok.s_D_ID<<
" threshold:<<stok->in_stok.s_threshold<<endl);
    HRESULT hres;
    try
    {
        hres =
txnHandle->comInterface.comHandle->doStockLevel(&txnHandle->comInterfa
ce.size,(UCHAR**)&txnHandle->comInterface.txnBuffer);
    }
    catch(...)
    {
        html+=sprintf(html,"ERROR: Com Stock call caused
exeception to occur.<PRE></BODY></HTML>");
        ERRORMSG("ERROR : Com Stock call caused
exeception to occur."<<endl);
        return OK;
    }
    if(FAILED(hres))
    {

```

```

        html+=sprintf(html,"ERROR: stok com call failed,
rc:%ld</PRE></BODY></HTML>",hres);
        ERRORMSG("ERROR : stok com call failed,
rc:"<<DEBUGADDRESS(hres)<<endl);
        return OK;
    }

    hres = txnHandle->comInterface.comHandle->doSetComplete();
    if(FAILED(hres))
    {
        html+=sprintf(html,"ERROR: stok com doSetComplete
failed, rc:%ld</PRE></BODY></HTML>",hres);
        ERRORMSG("ERROR : stok com doSetComplete failed,
rc:"<<DEBUGADDRESS(hres)<<endl);
        return OK;
    }
    stok = (stok_wrapper *)txnHandle->comInterface.txnBuffer;
    int rc = stok->out_stok.s_transtatus;
    if(rc != OK)
    {
        html+=displayStatus(html,rc);
        appendText(&html,"</PRE></BODY></HTML>");
        ERRORMSG("ERROR stok txn failed"<<endl

<<"stok->in_stok.s_D_ID:"<<stok->in_stok.s_D_ID<<endl
<<"stok->in_stok.s_threshold:"<<stok->in_stok.s_threshold<<endl
<<"stok->in_stok.s_W_ID:"<<stok->in_stok.s_W_ID<<endl
<<"stok->out_stok.deadlocks:"<<stok->out_stok.deadlocks<<endl
<<"stok->out_stok.s_low_stock:"<<stok->out_stok.s_low_stock<<endl
<<"stok->out_stok.s_transtatus:"<<stok->out_stok.s_transtatus<<endl);
        return OK;
    }

    //start creating result body

appendText(&html,"<FORM><CENTER><H3>Stock-Level</H3></CENTER>
>");
    appendText(&html,"<BR><PRE>\r\n"
        "Warehouse: ");

    //get w_id from wrapper
    char buffer[10];
    appendText(&html,ittoa(stok->in_stok.s_W_ID,buffer,10),6+1,1);

    appendText(&html,"District: ");
    appendText(&html,ittoa(stok->in_stok.s_D_ID,buffer,10));

    appendText(&html," <BR> <BR>"
        "Stock Level
Threshold: ");
    appendText(&html,ittoa(stok->in_stok.s_threshold,buffer,10));

    appendText(&html," <BR> <BR>"
        "Low Stock: ");
    appendText(&html,ittoa(stok->out_stok.s_low_stock,buffer,10));
    appendText(&html," <BR> <BR>");

    html+=displayStatus(html,rc);
    appendText(&html,"</PRE></BODY></HTML>");

    return OK;
}

```

```

/*
*****
** Name : doStockErrorPage
** Description :
** Parameters :
** char * html result
page
** char * query string
** htmlPhraser command
block
** TXN_HANDLE * handle for
this transaction
** Returns :
** int - return code
** Comments :
**
*****
*/

int doStockErrorPage(char *htmlPage,char *message,htmlPhraser
*commandBlock, TXN_HANDLE *txnHandle)
{
    char *html=htmlPage;

    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Stock
Level</TITLE></HEAD>\r\n"
        "<BODY><FORM
ACTION=\"\"
APP_NAME
\"\"
METHOD=\"GET\">\r\n"

"<CENTER><H3>Please Fill In Stock Form.</H3></CENTER> <BR>\r\n"
        "Submit Transaction
<INPUT TYPE=\"submit\" NAME=\"\"
CMD_TXN_ID
\"\" VALUE=\"\"
CMD_STOK
\">");

    html+=appendHiddenFields(html,txnHandle);

    appendText(&html,"<PRE>"
        "Warehouse: ");

    char buffer[15];
    appendText(&html,ittoa(txnHandle->w_id,buffer,10));
    appendSpaces(&html,2);
    appendText(&html,"District: ");
    appendText(&html,commandBlock->get_D_ID());
    appendText(&html," <BR> <BR>"
        "Stock Level
Threshold: "
        "<INPUT NAME=\"\"
CMD_STK_THRESHOLD
\"\" SIZE=1> <BR>
<BR>"
        "Low Stock: <BR>");

    appendText(&html,message);

    appendText(&html,"</PRE></FORM></BODY></HTML>");

    return OK;
}

```

```

/*
*****
** Name          : doExit
** Description   :
**              : HTML exit page entry point
** Parameters    :
**              : htmlPhraser*    command
block
**              : TXN_HANDLE*    txn handle
struct
** Returns      :
**              : int - return code
** Comments     :
**
*****
*/
int doExit(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    return (doLoginForm(commandBlock, txnHandle));
}

/*
*****
** Name          : displayStatus
** Description   :
**              : appends status string to the html
page
** Parameters    :
**              : char*          html page
**              : int            rc
** Returns      :
**              : amount of characters the function
appended
** Comments     :
**              : to the html page
**
*****
*/
int displayStatus(char *htmlPage, int rc)
{
    char *html = htmlPage;

    appendText(&html, "");

    switch (rc)
    {
    case OK:
        appendText(&html, "Execution Status: Transaction
Committed", 50, 1);
        break;
    case INVALID_ITEM:
        appendText(&html, "Execution Status: Item number is not
valid", 50, 1);
        break;
    case INVALID_STATUS:
        appendText(&html, "Execution Status: ERROR: Rollback
INVALID_STATUS", 50, 1);
        break;
    case INVALID_COM_STATUS:
        appendText(&html, "Execution Status: ERROR: Rollback
COM FAILURE", 50, 1);
        break;
    case ERR_DLVY_QUEUE_FULL:
        appendText(&html, "Execution Status: ERROR: Rollback
DLVY QUEUE FULL", 50, 1);
        break;
    default:

```

```

        appendText(&html, "Execution Status: ERROR:
Rollback", 50, 1);
    };
    appendText(&html, " ");
    return (int)(html - htmlPage);
}
/*
*****
** Name          : appendButtons
** Description   :
**              : append hidden field to recognize
user after login
** Parameters    :
**              : *htmlPage
html result page
**              : *TXN_HANDLE
txn handle
** Returns      :
**              : int
amount of characters the function appended
**
to the html page
** Comments     :
**
*****
*/
int appendHiddenFields(char *htmlPage, TXN_HANDLE *txnHandle)
{
    char *html = htmlPage;
    char buffer[15];

    appendText(&html, "<INPUT TYPE='hidden' NAME='\"
CMD_TERM_ID
\" VALUE='\"");
    appendText(&html, itoa(txnHandle->term_id, buffer, 10));
    appendText(&html, "\">\r\n");

    return (int)(html-htmlPage);
}
/*
*****
** Name          : appendButtons
** Description   :
**              : appends buttons transaction
buttons to result page
** Parameters    :
**              : *htmlPage
** Returns      :
**              : amount of characters the function
appended
**              : to the html page
** Comments     :
**
*****
*/
int appendButtons(char *htmlPage)
{
    char *html = htmlPage;

    appendText(&html, "<INPUT TYPE='submit' NAME='\"
CMD_TXN_ID
\" VALUE='\"

```

```

                                CMD_NORD
                                "\">>\r\n"
                                "<INPUT
TYPE="submit" NAME=""

                                CMD_TXN_ID
                                "\ VALUE=""
                                CMD_PYMT
                                "\">>\r\n"
                                "<INPUT
TYPE="submit" NAME=""

                                CMD_TXN_ID
                                "\ VALUE=""
                                CMD_ORDS
                                "\">>\r\n"
                                "<INPUT
TYPE="submit" NAME=""

                                CMD_TXN_ID
                                "\ VALUE=""
                                CMD_DLVS
                                "\">>\r\n"
                                "<INPUT
TYPE="submit" NAME=""

                                CMD_TXN_ID
                                "\ VALUE=""
                                CMD_STOK
                                "\">>\r\n"
                                "<INPUT
TYPE="submit" NAME=""

                                CMD_TXN_ID
                                "\ VALUE=""
                                CMD_EXIT
                                "\">>\r\n <BR>");

                                return (int)(html - htmlPage);
}

/*
*****
** Name          : appendItems
** Description    :
**               appends items to new order and
order status page
** Parameters    :
**               *htmlPage
**               html result page
**               short
**               items to append
**               short
**               item CMD id start
** Returns       :
**               amount of characters the function
appened
**               to the html page
** Comments      :
**
*****
*/
int appendItems(char *htmlPage,short itemCount,short cmdIDStart)
{
    char *html = htmlPage;
    char numBuffer[MAX_INT_BUFFER];

    for(int item=0;item < itemCount;item++)
    {
        appendText(&html,"<BR> <INPUT NAME=""");
        appendText(&html,itoa(cmdIDStart++,numBuffer,10));
        appendText(&html,"" SIZE=6> <INPUT NAME=""");
                                appendText(&html,itoa(cmdIDStart++,numBuffer,10));
                                appendText(&html,"" SIZE=6>
                                appendText(&html,itoa(cmdIDStart++,numBuffer,10));
                                appendText(&html,"" SIZE=2>\r\n");
                                }
                                return (int)(html - htmlPage);
}

/*
*****
** Name          : dlvyThreadEntry
** Description    :
**               dlvy thread worker entry point
** Parameters    :
** Returns       :
** Comments      :
**               All dlvy threads created by
initDly enter at
**               this point. They must first make a
connection
**               to the database, then go to sleep.
**               Main isapi threads control dlvy
worker semaphore
**               and signal when a dlvy txn is
queued.
**
*****
*/

void dlvyThreadEntry(void *)
{
    int rc = 0;

    DEBUGMSG("dlvyThread " << GetCurrentThreadId() << " entered
dlvyThreadEntry, calling db_connect to db:" << dbName << endl);

    void *connectHandle;
    //connect to database.
    DEBUGMSG("ptr created. calling db_connect to db:" << dbName
<< endl);
    rc = db_connect(dbName,&connectHandle);

    if(rc != OK)
    {
        ERRORMSG("dlvyThread " << GetCurrentThreadId()
<<" unable to connect to database, rc:" << rc << endl);
        DEBUGMSG("dlvyThread " << GetCurrentThreadId()
<<" unable to connect to database, rc:" << rc << endl);
        return;
    }

    DEBUGMSG("dlvyThread " << GetCurrentThreadId() << " connect
to db:" << dbName << " successful" << endl);

    FILE *dlvyLog = NULL;
    char logFileName[MAX_STRING_LEN] = {NULL};

    EnterCriticalSection(&isapiLock);
    //open dlvy log file for this thread
    sprintf(logFileName,"%s\\del_%d.txt",dlvyLogPath,dlvyThreadID);
    dlvyLog = fopen(logFileName,"w");
    if(!dlvyLog)
    {

```



```

        ERRORMSG("dlvyThread " << GetCurrentThreadId()
<< " unable to open dlvy log "
        << dlvyLogPath << "\\del_" <<
dlvyThreadID << endl);
        DEBUGMSG("dlvyThread " << GetCurrentThreadId()
<< " unable to open dlvy log "
        << dlvyLogPath << "\\del_" <<
dlvyThreadID << endl);
        return;
    }

    //increment the global dlvy thread id
    dlvyThreadID++;

    LeaveCriticalSection(&isapiLock);

    DEBUGMSG("dlvyThread " << GetCurrentThreadId() <<" dlvy log
file name: " << logFileName << " open." << endl);

    HANDLE workerHandles[2];
    //handle array to store event to wait on

    struct DLVYQUEUEDATA          dlvyQueueData;
    //dlvy queue struct to store queued txn
    struct dlvy_wrapper          dlvyTxn;
    //dlvy wrapper of db2 structs

    struct _timeb
endQueueTime;                //time stamp to queue removal time
    struct _timeb
endProcessTime;            //time stamp for end process time

    char    orderIDs[MAX_STRING_LEN] = {NULL};
    //string to store oids for each district
    int     bytesWritten =    0;
    int     dlvyCount =    0;

    DEBUGMSG("dlvyThread entering work loop" << endl);

    //successful, while true
    while(true)
    {
        try
        {
            handles" << endl);
            DEBUGMSG("dlvyThread initializing wait
            //wait for both program exit AND if there is
            work to do
            workerHandles[0] = dlvyThreadDone;
            workerHandles[1] = dlvyThreadSemaphore;

            DEBUGMSG("dlvyThread going to sleep
            waiting for wrk" << endl);

            rc =
            WaitForMultipleObjects(2,&workerHandles[0],FALSE,INFINITE);

            DEBUGMSG("dlvyThread awake, checking
            wake condition" << endl);

            if(rc == WAIT_OBJECT_0)
                break;
            else if(rc == (WAIT_OBJECT_0+1) )
            {
                DEBUGMSG("dlvyThread awake,
                wake condition of dlvyThreadSemaphore" << endl);
            }

            DEBUGMSG("dlvyThread trying to enter
            critical section" << endl);
            EnterCriticalSection(&dlvyQueueLock);

            DEBUGMSG("dlvyThread entered critical
            section" << endl);

            //remove queued dlvy txn
            dlvyQueueData.enqueueTime.time
            = dlvyQueue[dlvyBufferThreadIndex].enqueueTime.time;
            dlvyQueueData.enqueueTime.millitm
            = dlvyQueue[dlvyBufferThreadIndex].enqueueTime.millitm;
            dlvyQueueData.in_s_0_CARRIER_ID
            = dlvyQueue[dlvyBufferThreadIndex].in_s_0_CARRIER_ID;
            dlvyQueueData.warehouse
            = dlvyQueue[dlvyBufferThreadIndex].warehouse;

            DEBUGMSG("dlvyThread removed dlvy:"
            << dlvyCount << ",w_id:" << dlvyQueueData.warehouse
            << " carrier_id:" << dlvyQueueData.in_s_0_CARRIER_ID << endl);

            DEBUGMSG("dlvyThread removed dlvy in
            queue index: " <<dlvyBufferThreadIndex<< " w_id: " <<
            dlvyQueueData.warehouse
            << " carrier_id: " << dlvyQueueData.in_s_0_CARRIER_ID << endl);

            //increment the number of free slots
            dlvyBufferFreeSlots++;

            //increment the thread index to next slot in
            dlvy queue
            dlvyBufferThreadIndex++;

            DEBUGMSG("dlvyThread incremented
            amount of free slots:" << dlvyBufferFreeSlots << " and thread index:" <<
            dlvyBufferThreadIndex << endl);

            //check if we reached the end of dlvy queue, if
            so, reset back index back to 0
            if(dlvyBufferThreadIndex == dlvyQueueLen)
            {
                DEBUGMSG("dlvyThread reset
                dlvyBufferThreadIndex to 0, current dlvyBufferThreadIndex:" <<
                dlvyBufferThreadIndex
                << " free
                slots:"<<dlvyBufferFreeSlots<<endl);
                dlvyBufferThreadIndex=0;
            }
            DEBUGMSG("dlvyThread releasing critical
            section" << endl);

            LeaveCriticalSection(&dlvyQueueLock);

            //take enqueue time
            _ftime(&endQueueTime);

            DEBUGMSG("dlvyThread executing txn
            w_id:" << dlvyQueueData.warehouse
            << " carrier_id:" <<
            dlvyQueueData.in_s_0_CARRIER_ID << endl);

            //prepare to call database

```

```

        dlvyTxn.in_dlvy.s_O_CARRIER_ID =
dlvyQueueData.in_s_0_CARRIER_ID;
        dlvyTxn.in_dlvy.s_W_ID
=      dlvyQueueData.warehouse;
        dlvyTxn.out_dlvy.s_transtatus = -1;

        //increment dlvy count
        dlvyCount++;

        DEBUGMSG("dlvyThread %d calling dlvy
txn" << rc << endl);

        //call dlvy txn
        rc = dlvyCall(&dlvyTxn,connectHandle);

        _ftime(&endProcessTime);

        rc = dlvyTxn.out_dlvy.s_transtatus;

        DEBUGMSG("dlvy txn response time:"<<
(((endProcessTime.time -
endQueueTime.time)*1000)+
(endProcessTime.millitm - endQueueTime.millitm))/1000.0)<<
"
w_id:"<<dlvyTxn.in_dlvy.s_W_ID<<" carrier:"
<<dlvyTxn.in_dlvy.s_O_CARRIER_ID<<
"
deadLocks:"<<dlvyTxn.out_dlvy.deadlocks<<" rc: "<< rc <<endl);

        DEBUGMSG("dlvyThread dlvy s_transtatus:"
<< rc << endl);

        if(rc == OK)
        {
                bytesWritten=0;
                char *buffer = orderIDs;

                for(int
districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
                {
                        if(dlvyTxn.out_dlvy.s_O_ID[districtIndex] == 0)
                                bytesWritten
= sprintf(buffer,"%d had no new orders",districtIndex);
                                else
                                        bytesWritten
= sprintf(buffer,"%d ",dlvyTxn.out_dlvy.s_O_ID[districtIndex]);

                                buffer+=bytesWritten;
                }
                else
                        sprintf(orderIDs,"%nDelivery
transaction failed");

                fprintf(dlvyLog,DELIVERY_LOG_SUCCESS_STR,
dlvyCount,
dlvyQueueData.enqueueTime.time,
dlvyQueueData.enqueueTime.millitm,
endQueueTime.time,
endQueueTime.millitm,
dlvyQueueData.warehouse,
dlvyQueueData.in_s_0_CARRIER_ID,
orderIDs,
endProcessTime.time,
endProcessTime.millitm);

                fflush(dlvyLog);
                catch(...)
                {
                        ERRORMSG("ERROR: Unhandled
exeception in dlvy thread. Thread exiting"<<endl);
                        fprintf(dlvyLog,"ERROR: Unhandled
exeception in dlvy thread %ld. Thread exiting.\n",GetCurrentThreadId());
                        fflush(dlvyLog);

                        LeaveCriticalSection(&dlvyQueueLock);
                }
        } //end while true
}

/*
*****
** Name          : queueDlvyTxn
** Description   :
**              : function queues dlvy txn in dlvy
queue
** Parameters    :
**              : int warehouse
**              : short carrier
** Returns      :
**              : int error code
** Comments     :
**              : Function will queue
dlvy txn if 2 points are true
**              : 1) We have room in our
dlvy buffer
**              : 2) We writing over the
end of the queue
**
*****
*/
int queueDlvyTxn(int warehouse, short carrier_id)
{
        DEBUGMSG("Taking lock to queue dlvy txn.");

        EnterCriticalSection(&dlvyQueueLock);

        DEBUGMSG("Lock aquired to queue dlvy txn");

        if(dlvyBufferFreeSlots)
        {
                DEBUGMSG("Checking if we are inserting at tail of
dlvy queue."<<endl);
                if( dlvyBufferSlotIndex == (dlvyBufferThreadIndex-1))
                {
                        ERRORMSG("Error dlvy queue inserting
over unserviced queued dlvy txn."<<endl);
                }
        }
}

```

```

        DEBUGMSG("Error dlvy queue inserting
over unserviced queued dlvy txn."<<endl);
        LeaveCriticalSection(&dlvyQueueLock);
        return
ERR_DLVE_QUEUE_EATING_TAIL;
    }
    DEBUGMSG("free slots dlvy
queue:"<<dlvyBufferFreeSlots<<" inserting txn in slot: "
<<dlvyBufferSlotIndex<<
        "w_id: "<<warehouse<<" carrier:
"<<carrier_id<<endl);

warehouse;
        dlvyQueue[dlvyBufferSlotIndex].warehouse =
warehouse;
        dlvyQueue[dlvyBufferSlotIndex].in_s_0_CARRIER_ID
= carrier_id;

_ftime(&dlvyQueue[dlvyBufferSlotIndex].enqueueTime);

        //decrement the number of free slots in the buffer
dlvyBufferFreeSlots--;

        //increment the index to the next dlvy queue slot.
dlvyBufferSlotIndex++;

        DEBUGMSG("dlvy txn queued, slots available in
queue:"<<dlvyBufferFreeSlots<<" queue slot index:"<<dlvyBufferSlotIndex
<<"w_id:"<<warehouse<<"
carrier:"<<carrier_id<<endl);

        DEBUGMSG("dlvy txn queued, slots available in queue:
"<<dlvyBufferFreeSlots<<" queue slot index: "<<dlvyBufferSlotIndex
<<" w_id: "<<warehouse<<"
carrier: "<<carrier_id<<endl);

        if(dlvyBufferSlotIndex == dlvyQueueLen)
        {
            DEBUGMSG("queue slot index hit end of
queue, reset to 0, current index:"<<dlvyBufferSlotIndex<<" free
slots:"<<dlvyBufferFreeSlots<<endl);
            dlvyBufferSlotIndex=0;
        }
        else
        {
            //no slots available in dlvy buffer, release critical section
and return an nord->in_nord.in_item
            LeaveCriticalSection(&dlvyQueueLock);
            ERRORMSG("dlvy queue buffer full, increase the dlvy
queue length."<<endl);
            return ERR_DLVE_QUEUE_FULL;
        }

        LeaveCriticalSection(&dlvyQueueLock);

        //release semaphore to wake thread that there is work
ReleaseSemaphore(dlvyThreadSemaphore,1,NULL);

        return OK;
}

/*
*****
** Name          : doHtml

```

```

** Description   :
**              : HTML processing page entry
point
** Parameters    :
**              : txn handle
** Returns      :
**              : int - return code
** Comments     :
**              :
*****
*/

void doHtml(TXN_HANDLE *txnHandle)
{
    DEBUGMSG("Entered doHtml(), parsing query string:"<<
txnHandle->urlString <<" into command block"<<endl);
    htmlPhraser        commandBlock(txnHandle->urlString);
    DEBUGMSG("Query string parsed. command:"<<
commandBlock.getCommandId() <<" user's terminal id:" <<
commandBlock.get_TERM_ID() <<endl);

    int terminalID = atoi(commandBlock.get_TERM_ID());
    int commandID = commandBlock.getCommandId();

    DEBUGMSG("User sent in a terimal id:"<<terminalID<<"; checking
to see if user has logged in before"<<endl);
    if(terminalID > 0)
    {
        DEBUGMSG("Terminal id > 0, user has logged in
already, terminalID:"<<terminalID<<" retrieving warehouse district
pair"<<endl);
        if(getTerminal(terminalID,txnHandle) != OK)
            return;
        DEBUGMSG("User had valid terminal id, user's login
warehouse:"<<txnHandle->w_id<<" district:"<<txnHandle->d_id<<endl);
    }
    else
    {
        DEBUGMSG("User did not submit a terminal id or valid
terminal id, ensure that the user is trying to log in."<<endl);
        if( (commandID != TXN_LOGIN) && (commandID !=
TXN_LOGIN_RESULTS) )
        {
            DEBUGMSG("ERROR : User has not logged
in."<<endl);
            ERRORMSG("ERROR : User has not logged
in."<<endl);
            sprintf(txnHandle->htmlPage,"ERROR: User
has not logged in or did not submit a valid terminal.");
            return;
        }
        DEBUGMSG("User is in process of logging in,
commandID:"<<commandID<<endl);
    }

    DEBUGMSG("Calling html page
function:"<<commandBlock.getCommandId()<<endl);
    int rc =
htmlPageFunctions[commandBlock.getCommandId()](&commandBlock,txnHa
ndle);
    DEBUGMSG("Return from html page
function:"<<commandBlock.getCommandId()<<endl);

    return;
}

/*
*****

```

```

** Name : getTerminal
** Description :
** retrieves terminal information
based on terminal id
** Parameters :
** int
terminal id
** TERM_HANDLE* txn handle
** Returns :
** int - return code
** Comments :
**
*****
*/
int getTerminal(int terminal, TXN_HANDLE *txnHandle)
{
    //check to see if terminal id is out of range
    if(terminal >= numUsers)
    {
        //terminal id not valid.
        sprintf(txnHandle->htmlPage, "ERROR: Client does not
support more than %d users, terminal id:%d", numUsers, terminal);
        ERRORMSG("ERROR : Client does not support more
than "<<numUsers<<" users, terminal id:<<terminal<<endl);
        return ERR;
    }

    //check if terminal id is points to a not in use terminal
    if(!(termArray+terminal)->terminalInUse)
    {
        sprintf(txnHandle->htmlPage, "ERROR: Terminal id
given points to a not in use terminal.");
        ERRORMSG("ERROR : Terminal id given points to a
not in use terminal.<<endl);
        return ERR;
    }

    DEBUGMSG("Storing terminal warehouse, district , and initial term
id for user:<<terminal<<endl);

    //assign terminal values to txn_handle
    txnHandle->d_id = termArray[terminal].d_id;
    txnHandle->w_id = termArray[terminal].w_id;
    txnHandle->term_id = termArray[terminal].terminalID;

    DEBUGMSG("Users terminal:<<terminal<<, stored
warehouse:<<txnHandle->w_id<<
" district:<<txnHandle->d_id<<" terminalID
stored:<<txnHandle->term_id<<endl);

    return OK;
}

/*
*****
** Name : assignTerminal
** Description :
** assigns terminal index to user
** Parameters :
** TERM_HANDLE* txn handle
** Returns :
** int - return code
** Comments :
**
*****
*/
int assignTerminal(TXN_HANDLE *txnHandle)
{

```

```

EnterCriticalSection(&termLock);

//check if terminal array is full.
if(termNextFree == numUsers)
{
    LeaveCriticalSection(&termLock);
    return ERR;
}

    DEBUGMSG("Storing user warehouse:<<txnHandle->w_id<<"
district:<< txnHandle->d_id<<
" in terminal slot:<<termNextFree<<endl);

    //store users w_id and d_id
    termArray[termNextFree].d_id = txnHandle->d_id;
    termArray[termNextFree].w_id = txnHandle->w_id;

    //set terminal slot to be in use
    termArray[termNextFree].terminalInUse = true;
    termArray[termNextFree].terminalID = termNextFree;
    //in txn handle, set the terminal id
    txnHandle->term_id = termNextFree;

    //increment to next free terminal.
    termNextFree++;

    DEBUGMSG("User warehouse:<<txnHandle->w_id<<"
district:<< txnHandle->d_id <<
" stored in terminal slot:<<txnHandle->term_id<<" next
terminal free:<<termNextFree<<endl);

    LeaveCriticalSection(&termLock);

    return OK;
}

```

TpccIsapi.def

; tpccIsapi.def : declares the module parameters for the DLL.

```

LIBRARY "tpccIsapi"

EXPORTS
    HttpExtensionProc
    GetExtensionVersion
    TerminateExtension

```

TpccIsapi.hpp

```

/*
*****
** Project : AIX
** Component : Performance/TPC-W Benchmark
** Name : tpccIsapi.hpp
** Title : ISAPI interface for tpcc
*****
** Copyright (c) 2001,2002 IBM Corporation
** All rights reserved
*****
** History :
** Developed at IBM Austin by the AIX RS/6000
** performance group.
**

```

```

** Comments      :
**
*****
*/

#ifndef __tpccISAPI_hpp__
#define __tpccISAPI_hpp__

#include <windows.h>
#include <httpext.h>

#include <tpcc.h>
#include <htmlPhraser.h>
#include <iomanip>

#include <db2tpcc.h>
#include <comsvcs.h>

/////////////////////////////////////////////////////////////////
// Terminal struct
/////////////////////////////////////////////////////////////////
struct TERM_ENTRY
{
    int          terminalID;
    bool         terminalInUse;
    int          w_id;
    short        d_id;
};

/////////////////////////////////////////////////////////////////
// COM interface
/////////////////////////////////////////////////////////////////
struct COM_HANDLE
{
    Itpcc_com *comHandle;
    char      *txnBuffer;
    int       size;
};

/////////////////////////////////////////////////////////////////
// TXN handle
/////////////////////////////////////////////////////////////////
struct TXN_HANDLE
{
    char      htmlPage[MAX_HTML_PAGE_LEN];
    char      htmlHeader[MAX_HTML_HEADER_LEN];
    char      *urlString;

    //user data
    int       w_id;
    int       d_id;
    int       sync_id;
    int       term_id;
    int       conn_id;

    COM_HANDLE comInterface;
};

struct DLVYQUEUEDATA
{
    int       warehouse;
    short    in_s_0_CARRIER_ID;
    struct _timeb enqueueTime;
};

/////////////////////////////////////////////////////////////////
// Definitions
/////////////////////////////////////////////////////////////////

#define INVALID_ITEM          100
#define HEADER
"Content-Type:text/html\r\nContent-Length: %d\r\nConnection:
Keep-Alive\r\n\r\n"
#define TLS_NULL
0xFFFFFFFF
#define ACCESS_TIMEOUT      3600000
seconds //One hour in milli

#define DELIVERY_LOG_SUCCESS_STR "--Tran %d
Queue %d.%03d Start %d.%03d\r\nW_ID: %d CARRIER_ID: %d
%s\r\nend-time: %d.%03d\r\n"

/////////////////////////////////////////////////////////////////
// Function Prototypes
/////////////////////////////////////////////////////////////////

int initDlvy();
int initTxnHandle(TXN_HANDLE **txnHandle);
int closeTxnHandle(TXN_HANDLE *txnHandle);
int readRegistryValues();
int getTerminal(int terminal, TXN_HANDLE *txnHandle);
int assignTerminal(TXN_HANDLE *txnHandle);
int getDBInstance();

void doHtml(TXN_HANDLE *txnHandle);
int doLoginForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doLoginResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doNewOrderForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doNewOrderResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doPaymentForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doPaymentResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doOrderStatusForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doOrderStatusResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doDeliveryForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doDeliveryResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doStockForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doStockResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doExit(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);

int doLoginErrorPage(char *htmlPage, char *message);
int doNewOrderErrorPage(char *htmlPage, char *message, htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doPaymentErrorPage(char *htmlPage, char *message, htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doOrderStatusErrorPage(char *htmlPage, char *message, htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doDeliveryErrorPage(char *htmlPage, char *message, htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doStockErrorPage(char *htmlPage, char *message, htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);

void dlvyThreadEntry(void *);
int queueDlvyTxn(int warehouse, short carrier_id);

int appendButtons(char *htmlPage);
int appendItems(char *htmlPage, short itemCount, short cmdIDStart);
int appendHiddenFields(char *htmlPage, TXN_HANDLE *txnHandle);

```

```
int displayStatus(char *htmlPage,int rc);
```

```
#endif
```

Tpcclsapi.rc

```
// Microsoft Visual C++ generated resource script.
```

```
//  
#include "resource.h"
```

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

```
#include "winres.h"  
#include "atlsrvres.h"
```

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

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

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

```
2 TEXTINCLUDE  
BEGIN  
"#include ""winres.h""\r\n"  
"#include ""atlsrvres.h""\r\n"  
"\0"  
END
```

```
3 TEXTINCLUDE  
BEGIN  
  
"LANGUAGE LANG_ENGLISH,  
SUBLANG_ENGLISH_US\r\n"  
"#pragma code_page(1252)\r\n"  
"#include ""atlsrv.rc""\r\n"  
"\0"  
END
```

```
#endif // APSTUDIO_INVOKED
```

```
////////////////////////////////////  
//  
// Version  
//
```

```
VS_VERSION_INFO VERSIONINFO  
FILEVERSION 1,0,0,1  
PRODUCTVERSION 1,0,0,1  
FILEFLAGSMASK 0x3fL  
#ifdef _DEBUG  
FILEFLAGS 0x1L  
#else  
FILEFLAGS 0x0L
```

```
#endif  
FILEOS 0x4L  
FILETYPE 0x2L  
FILESUBTYPE 0x0L  
BEGIN  
BLOCK "StringFileInfo"  
BEGIN  
BLOCK "040904e4"  
BEGIN  
  
VALUE "CompanyName", "TODO:  
<Company name>"  
VALUE "FileDescription", "TODO: <File description>"  
VALUE "FileVersion", "1.0.0.1"  
VALUE "InternalName", "isapi.dll"  
VALUE "LegalCopyright", "TODO: (c) <Company name>. All rights  
reserved."  
VALUE "OriginalFilename", "isapi.dll"  
VALUE "ProductName", "TODO: <Product name>"  
VALUE "ProductVersion", "1.0.0.1"  
VALUE "OLESelfRegister", ""  
END  
END  
BLOCK "VarFileInfo"  
BEGIN  
  
VALUE "Translation", 0x0409, 1252  
END  
END
```

```
LANGUAGE 9, 1  
#pragma code_page(1252)  
////////////////////////////////////  
//  
// String Table  
//
```

```
STRINGTABLE  
BEGIN  
IDS_PROJNAME "tpcclsapi"  
END
```

```
////////////////////////////////////
```

```
#ifndef APSTUDIO_INVOKED  
////////////////////////////////////  
//  
// Generated from the TEXTINCLUDE 3 resource.  
//  
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US  
#pragma code_page(1252)  
#include "atlsrv.rc"
```

```
////////////////////////////////////  
#endif // not APSTUDIO_INVOKED
```

Appendix B: Database Design Scripts

create_tablespace.ddl

connect to tpcc;

-- WAR

```
create regular tablespace WAR PAGESIZE 4096 managed by database using (
DEVICE 'C:\Containers\WAR\001' 256,
DEVICE 'C:\Containers\WAR\002' 256,
DEVICE 'C:\Containers\WAR\003' 256,
DEVICE 'C:\Containers\WAR\004' 256,
DEVICE 'C:\Containers\WAR\005' 256,
DEVICE 'C:\Containers\WAR\006' 256,
DEVICE 'C:\Containers\WAR\007' 256,
DEVICE 'C:\Containers\WAR\008' 256,
DEVICE 'C:\Containers\WAR\009' 256,
DEVICE 'C:\Containers\WAR\010' 256,
DEVICE 'C:\Containers\WAR\011' 256,
DEVICE 'C:\Containers\WAR\012' 256
) EXTENTSIZE 16 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
```

-- DIS

```
create regular tablespace DIS PAGESIZE 4096 managed by database using (
DEVICE 'C:\Containers\DIS\001' 512,
DEVICE 'C:\Containers\DIS\002' 512,
DEVICE 'C:\Containers\DIS\003' 512,
DEVICE 'C:\Containers\DIS\004' 512,
DEVICE 'C:\Containers\DIS\005' 512,
DEVICE 'C:\Containers\DIS\006' 512,
DEVICE 'C:\Containers\DIS\007' 512,
DEVICE 'C:\Containers\DIS\008' 512,
DEVICE 'C:\Containers\DIS\009' 512,
DEVICE 'C:\Containers\DIS\010' 512,
DEVICE 'C:\Containers\DIS\011' 512,
DEVICE 'C:\Containers\DIS\012' 512
) EXTENTSIZE 32 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
```

-- ITM

```
create regular tablespace ITM PAGESIZE 8192 managed by database using (
DEVICE 'C:\Containers\ITM\001' 256,
DEVICE 'C:\Containers\ITM\002' 256,
DEVICE 'C:\Containers\ITM\003' 256,
DEVICE 'C:\Containers\ITM\004' 256,
DEVICE 'C:\Containers\ITM\005' 256,
DEVICE 'C:\Containers\ITM\006' 256,
DEVICE 'C:\Containers\ITM\007' 256,
DEVICE 'C:\Containers\ITM\008' 256,
DEVICE 'C:\Containers\ITM\009' 256,
DEVICE 'C:\Containers\ITM\010' 256,
DEVICE 'C:\Containers\ITM\011' 256,
DEVICE 'C:\Containers\ITM\012' 256
) EXTENTSIZE 32 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;
```

-- STK

```
create regular tablespace STK_001 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\001' 7875840,
DEVICE 'C:\Containers\STK\002' 7875840
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
```

```
create regular tablespace STK_002 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\003' 7875840,
```

```
DEVICE 'C:\Containers\STK\004' 7875840
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
```

```
create regular tablespace STK_003 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\005' 7875840,
DEVICE 'C:\Containers\STK\006' 7875840
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
```

```
create regular tablespace STK_004 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\007' 7875840,
DEVICE 'C:\Containers\STK\008' 7875840
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
```

```
create regular tablespace STK_005 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\009' 7875840,
DEVICE 'C:\Containers\STK\010' 7875840
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
```

```
create regular tablespace STK_006 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\011' 7875840,
DEVICE 'C:\Containers\STK\012' 7875840
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
```

```
create regular tablespace STK_007 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\013' 1313024,
DEVICE 'C:\Containers\STK\014' 1313024,
DEVICE 'C:\Containers\STK\015' 1313024,
DEVICE 'C:\Containers\STK\016' 1313024,
DEVICE 'C:\Containers\STK\017' 1313024,
DEVICE 'C:\Containers\STK\018' 1313024,
DEVICE 'C:\Containers\STK\019' 1313024,
DEVICE 'C:\Containers\STK\020' 1313024,
DEVICE 'C:\Containers\STK\021' 1313024,
DEVICE 'C:\Containers\STK\022' 1313024,
DEVICE 'C:\Containers\STK\023' 1313024,
DEVICE 'C:\Containers\STK\024' 1313024
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
```

-- CST

```
create regular tablespace CST_001 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\CST\001' 5670912,
DEVICE 'C:\Containers\CST\002' 5670912
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
```

```
create regular tablespace CST_002 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\CST\003' 5670912,
DEVICE 'C:\Containers\CST\004' 5670912
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
```

```
create regular tablespace CST_003 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\CST\005' 5670912,
DEVICE 'C:\Containers\CST\006' 5670912
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
```

```
create regular tablespace CST_004 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\CST\007' 5670912,
DEVICE 'C:\Containers\CST\008' 5670912
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
```

```

create regular tablespace CST_005 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\CST\009' 5670912,
DEVICE 'C:\Containers\CST\010' 5670912
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace CST_006 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\CST\011' 5670912,
DEVICE 'C:\Containers\CST\012' 5670912
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace CST_007 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\CST\013' 945408,
DEVICE 'C:\Containers\CST\014' 945408,
DEVICE 'C:\Containers\CST\015' 945408,
DEVICE 'C:\Containers\CST\016' 945408,
DEVICE 'C:\Containers\CST\017' 945408,
DEVICE 'C:\Containers\CST\018' 945408,
DEVICE 'C:\Containers\CST\019' 945408,
DEVICE 'C:\Containers\CST\020' 945408,
DEVICE 'C:\Containers\CST\021' 945408,
DEVICE 'C:\Containers\CST\022' 945408,
DEVICE 'C:\Containers\CST\023' 945408,
DEVICE 'C:\Containers\CST\024' 945408
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

-- CSTI
create regular tablespace CSTI_001 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\CSTI\001' 179712,
DEVICE 'C:\Containers\CSTI\002' 179712
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace CSTI_002 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\CSTI\003' 179712,
DEVICE 'C:\Containers\CSTI\004' 179712
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace CSTI_003 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\CSTI\005' 179712,
DEVICE 'C:\Containers\CSTI\006' 179712
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace CSTI_004 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\CSTI\007' 179712,
DEVICE 'C:\Containers\CSTI\008' 179712
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace CSTI_005 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\CSTI\009' 179712,
DEVICE 'C:\Containers\CSTI\010' 179712
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace CSTI_006 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\CSTI\011' 179712,
DEVICE 'C:\Containers\CSTI\012' 179712
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace CSTI_007 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\CSTI\013' 29952,
DEVICE 'C:\Containers\CSTI\014' 29952,
DEVICE 'C:\Containers\CSTI\015' 29952,
DEVICE 'C:\Containers\CSTI\016' 29952,
DEVICE 'C:\Containers\CSTI\017' 29952,
DEVICE 'C:\Containers\CSTI\018' 29952,
DEVICE 'C:\Containers\CSTI\019' 29952,
DEVICE 'C:\Containers\CSTI\020' 29952,
DEVICE 'C:\Containers\CSTI\021' 29952,
DEVICE 'C:\Containers\CSTI\022' 29952,
DEVICE 'C:\Containers\CSTI\023' 29952,
DEVICE 'C:\Containers\CSTI\024' 29952
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

-- ORD
create regular tablespace ORD PAGESIZE 8192 managed by database using (
DEVICE 'C:\Containers\ORD\001' 190720,
DEVICE 'C:\Containers\ORD\002' 190720,
DEVICE 'C:\Containers\ORD\003' 190720,
DEVICE 'C:\Containers\ORD\004' 190720,
DEVICE 'C:\Containers\ORD\005' 190720,
DEVICE 'C:\Containers\ORD\006' 190720,
DEVICE 'C:\Containers\ORD\007' 190720,
DEVICE 'C:\Containers\ORD\008' 190720,
DEVICE 'C:\Containers\ORD\009' 190720,
DEVICE 'C:\Containers\ORD\010' 190720,
DEVICE 'C:\Containers\ORD\011' 190720,
DEVICE 'C:\Containers\ORD\012' 190720
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

-- ORDI
create regular tablespace ORDI PAGESIZE 8192 managed by database using (
DEVICE 'C:\Containers\ORDI\001' 169472,
DEVICE 'C:\Containers\ORDI\002' 169472,
DEVICE 'C:\Containers\ORDI\003' 169472,
DEVICE 'C:\Containers\ORDI\004' 169472,
DEVICE 'C:\Containers\ORDI\005' 169472,
DEVICE 'C:\Containers\ORDI\006' 169472,
DEVICE 'C:\Containers\ORDI\007' 169472,
DEVICE 'C:\Containers\ORDI\008' 169472,
DEVICE 'C:\Containers\ORDI\009' 169472,
DEVICE 'C:\Containers\ORDI\010' 169472,
DEVICE 'C:\Containers\ORDI\011' 169472,
DEVICE 'C:\Containers\ORDI\012' 169472
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

-- OLN
create regular tablespace OLN_001 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\OLN\001' 4378624,
DEVICE 'C:\Containers\OLN\002' 4378624
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace OLN_002 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\OLN\003' 4378624,
DEVICE 'C:\Containers\OLN\004' 4378624
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace OLN_003 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\OLN\005' 4378624,
DEVICE 'C:\Containers\OLN\006' 4378624
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace OLN_004 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\OLN\007' 4378624,
DEVICE 'C:\Containers\OLN\008' 4378624
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

```



```

) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace OLN_005 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\OLN\009' 4378624,
DEVICE 'C:\Containers\OLN\010' 4378624
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace OLN_006 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\OLN\011' 4378624,
DEVICE 'C:\Containers\OLN\012' 4378624
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace OLN_007 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\OLN\013' 730112,
DEVICE 'C:\Containers\OLN\014' 730112,
DEVICE 'C:\Containers\OLN\015' 730112,
DEVICE 'C:\Containers\OLN\016' 730112,
DEVICE 'C:\Containers\OLN\017' 730112,
DEVICE 'C:\Containers\OLN\018' 730112,
DEVICE 'C:\Containers\OLN\019' 730112,
DEVICE 'C:\Containers\OLN\020' 730112,
DEVICE 'C:\Containers\OLN\021' 730112,
DEVICE 'C:\Containers\OLN\022' 730112,
DEVICE 'C:\Containers\OLN\023' 730112,
DEVICE 'C:\Containers\OLN\024' 730112
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

-- NEWA
create regular tablespace NEWA_001 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWA\001' 84224,
DEVICE 'C:\Containers\NEWA\002' 84224
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWA_002 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWA\003' 84224,
DEVICE 'C:\Containers\NEWA\004' 84224
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWA_003 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWA\005' 84224,
DEVICE 'C:\Containers\NEWA\006' 84224
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWA_004 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWA\007' 84224,
DEVICE 'C:\Containers\NEWA\008' 84224
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWA_005 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWA\009' 84224,
DEVICE 'C:\Containers\NEWA\010' 84224
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWA_006 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWA\011' 84224,
DEVICE 'C:\Containers\NEWA\012' 84224
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWA_007 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWA\013' 14336,
DEVICE 'C:\Containers\NEWA\014' 14336,
DEVICE 'C:\Containers\NEWA\015' 14336,
DEVICE 'C:\Containers\NEWA\016' 14336,
DEVICE 'C:\Containers\NEWA\017' 14336,
DEVICE 'C:\Containers\NEWA\018' 14336,
DEVICE 'C:\Containers\NEWA\019' 14336,
DEVICE 'C:\Containers\NEWA\020' 14336,
DEVICE 'C:\Containers\NEWA\021' 14336,
DEVICE 'C:\Containers\NEWA\022' 14336,
DEVICE 'C:\Containers\NEWA\023' 14336,
DEVICE 'C:\Containers\NEWA\024' 14336
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

-- HST
create regular tablespace HST PAGESIZE 16384 managed by database using (
DEVICE 'C:\Containers\HST\001' 186624,
DEVICE 'C:\Containers\HST\002' 186624,
DEVICE 'C:\Containers\HST\003' 186624,
DEVICE 'C:\Containers\HST\004' 186624,
DEVICE 'C:\Containers\HST\005' 186624,
DEVICE 'C:\Containers\HST\006' 186624,
DEVICE 'C:\Containers\HST\007' 186624,
DEVICE 'C:\Containers\HST\008' 186624,
DEVICE 'C:\Containers\HST\009' 186624,
DEVICE 'C:\Containers\HST\010' 186624,
DEVICE 'C:\Containers\HST\011' 186624,
DEVICE 'C:\Containers\HST\012' 186624
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT16K;

-- NEWB
create regular tablespace NEWB_001 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\001' 84224,
DEVICE 'C:\Containers\NEWB\002' 84224
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWB_002 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\003' 84224,
DEVICE 'C:\Containers\NEWB\004' 84224
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWB_003 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\005' 84224,
DEVICE 'C:\Containers\NEWB\006' 84224
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWB_004 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\007' 84224,
DEVICE 'C:\Containers\NEWB\008' 84224
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWB_005 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\009' 84224,
DEVICE 'C:\Containers\NEWB\010' 84224
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWB_006 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\011' 84224,
DEVICE 'C:\Containers\NEWB\012' 84224
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

```

```

create regular tablespace NEWB_007 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\013' 14336,
DEVICE 'C:\Containers\NEWB\014' 14336,
DEVICE 'C:\Containers\NEWB\015' 14336,
DEVICE 'C:\Containers\NEWB\016' 14336,
DEVICE 'C:\Containers\NEWB\017' 14336,
DEVICE 'C:\Containers\NEWB\018' 14336,
DEVICE 'C:\Containers\NEWB\019' 14336,
DEVICE 'C:\Containers\NEWB\020' 14336,
DEVICE 'C:\Containers\NEWB\021' 14336,
DEVICE 'C:\Containers\NEWB\022' 14336,
DEVICE 'C:\Containers\NEWB\023' 14336,
DEVICE 'C:\Containers\NEWB\024' 14336
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

```

```
connect reset;
```

alter_tablespace.ddl

```

-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
-----

```

```
-- Set Bufferpools For Tablespaces
```

```

connect to tpc;
alter tablespace WAR    bufferpool WAR;
alter tablespace DIS    bufferpool DIS;
alter tablespace ITM    bufferpool ITM;
alter tablespace STK_001 bufferpool STK;
alter tablespace STK_002 bufferpool STK;
alter tablespace STK_003 bufferpool STK;
alter tablespace STK_004 bufferpool STK;
alter tablespace STK_005 bufferpool STK;
alter tablespace STK_006 bufferpool STK;
alter tablespace STK_007 bufferpool STK;
alter tablespace CST_001 bufferpool CST;
alter tablespace CST_002 bufferpool CST;
alter tablespace CST_003 bufferpool CST;
alter tablespace CST_004 bufferpool CST;
alter tablespace CST_005 bufferpool CST;
alter tablespace CST_006 bufferpool CST;
alter tablespace CST_007 bufferpool CST;
alter tablespace CSTI_001 bufferpool CST_I;
alter tablespace CSTI_002 bufferpool CST_I;
alter tablespace CSTI_003 bufferpool CST_I;
alter tablespace CSTI_004 bufferpool CST_I;
alter tablespace CSTI_005 bufferpool CST_I;
alter tablespace CSTI_006 bufferpool CST_I;
alter tablespace CSTI_007 bufferpool CST_I;
alter tablespace ORD    bufferpool ORD;
alter tablespace ORDI   bufferpool ORD_I;
alter tablespace OLN_001 bufferpool OLN;
alter tablespace OLN_002 bufferpool OLN;
alter tablespace OLN_003 bufferpool OLN;
alter tablespace OLN_004 bufferpool OLN;
alter tablespace OLN_005 bufferpool OLN;

```

```

alter tablespace OLN_006 bufferpool OLN;
alter tablespace OLN_007 bufferpool OLN;
alter tablespace NEWA_001 bufferpool NEW;
alter tablespace NEWA_002 bufferpool NEW;
alter tablespace NEWA_003 bufferpool NEW;
alter tablespace NEWA_004 bufferpool NEW;
alter tablespace NEWA_005 bufferpool NEW;
alter tablespace NEWA_006 bufferpool NEW;
alter tablespace NEWA_007 bufferpool NEW;
alter tablespace NEWB_001 bufferpool NEW;
alter tablespace NEWB_002 bufferpool NEW;
alter tablespace NEWB_003 bufferpool NEW;
alter tablespace NEWB_004 bufferpool NEW;
alter tablespace NEWB_005 bufferpool NEW;
alter tablespace NEWB_006 bufferpool NEW;
alter tablespace NEWB_007 bufferpool NEW;
alter tablespace HST    bufferpool HST;

```

```
connect reset;
terminate;
```

alter_bufferpool.ddl

```

-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
-----

```

```
-- Alter Size of Bufferpools
```

```

connect to tpc;

alter bufferpool IBMDEFAULT8K deferred size 16;
alter bufferpool IBMDEFAULT16K deferred size 16;
alter bufferpool WAR          deferred size 325;
alter bufferpool DIS          deferred size 4000;
alter bufferpool ITM          deferred size 1235;
alter bufferpool HST          deferred size 825;
alter bufferpool NEW          deferred size 240000;
alter bufferpool ORD          deferred size 131500;
alter bufferpool CST          deferred size 20000;
alter bufferpool STK          deferred size 12420000;
alter bufferpool OLN          deferred size 480000;
alter bufferpool CST_I        deferred size 150000;
alter bufferpool ORD_I        deferred size 350000;

```

```
connect reset;
terminate;
```

create_bufferpool.ddl

```

-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2005

```

```
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
```

```
-----
-- Create Bufferpools
connect to tpcc;
create bufferpool IBMDEFAULTBP size 1310720 pagesize 4096;
create bufferpool IBMDEFAULT8K size 655360 pagesize 8192;
create bufferpool IBMDEFAULT16K size 5000 pagesize 16384;
create bufferpool WAR size 1000 pagesize 4096;
create bufferpool DIS size 1000 pagesize 4096;
create bufferpool ITM size 1000 pagesize 8192;
create bufferpool HST size 1000 pagesize 16384;
create bufferpool NEW size 1000 pagesize 4096;
create bufferpool ORD size 1000 pagesize 8192;
create bufferpool CST size 1000 pagesize 4096;
create bufferpool STK size 1000 pagesize 4096;
create bufferpool OLN size 1000 pagesize 8192;
create bufferpool CST_I size 1000 pagesize 8192;
create bufferpool ORD_I size 1000 pagesize 8192;
connect reset;
terminate;
```

create_database.ddl

```
-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
```

```
-----
-- Create Database
drop database tpcc;
create database tpcc collate using identity;
```

altpbsp_pf_0.ddl

```
connect to TPCC;
alter tablespace CSTI_001 prefetchsize 0;
alter tablespace CSTI_002 prefetchsize 0;
alter tablespace CSTI_003 prefetchsize 0;
alter tablespace CSTI_004 prefetchsize 0;
alter tablespace CSTI_005 prefetchsize 0;
alter tablespace CSTI_006 prefetchsize 0;
alter tablespace CSTI_007 prefetchsize 0;
alter tablespace CST_001 prefetchsize 0;
alter tablespace CST_002 prefetchsize 0;
alter tablespace CST_003 prefetchsize 0;
alter tablespace CST_004 prefetchsize 0;
alter tablespace CST_005 prefetchsize 0;
alter tablespace CST_006 prefetchsize 0;
alter tablespace CST_007 prefetchsize 0;
alter tablespace DIS prefetchsize 0;
alter tablespace HST prefetchsize 0;
alter tablespace ITM prefetchsize 0;
alter tablespace NEWA_001 prefetchsize 0;
alter tablespace NEWA_002 prefetchsize 0;
alter tablespace NEWA_003 prefetchsize 0;
```

```
alter tablespace NEWA_004 prefetchsize 0;
alter tablespace NEWA_005 prefetchsize 0;
alter tablespace NEWA_006 prefetchsize 0;
alter tablespace NEWA_007 prefetchsize 0;
alter tablespace NEWB_001 prefetchsize 0;
alter tablespace NEWB_002 prefetchsize 0;
alter tablespace NEWB_003 prefetchsize 0;
alter tablespace NEWB_004 prefetchsize 0;
alter tablespace NEWB_005 prefetchsize 0;
alter tablespace NEWB_006 prefetchsize 0;
alter tablespace NEWB_007 prefetchsize 0;
alter tablespace OLN_001 prefetchsize 0;
alter tablespace OLN_002 prefetchsize 0;
alter tablespace OLN_003 prefetchsize 0;
alter tablespace OLN_004 prefetchsize 0;
alter tablespace OLN_005 prefetchsize 0;
alter tablespace OLN_006 prefetchsize 0;
alter tablespace OLN_007 prefetchsize 0;
alter tablespace ORD prefetchsize 0;
alter tablespace ORD_I prefetchsize 0;
alter tablespace STK_001 prefetchsize 0;
alter tablespace STK_002 prefetchsize 0;
alter tablespace STK_003 prefetchsize 0;
alter tablespace STK_004 prefetchsize 0;
alter tablespace STK_005 prefetchsize 0;
alter tablespace STK_006 prefetchsize 0;
alter tablespace STK_007 prefetchsize 0;
alter tablespace WAR prefetchsize 0;
connect reset;
```

altpbsp_pf_1024.ddl

```
connect to TPCC;
alter tablespace CSTI_001 prefetchsize 1024;
alter tablespace CSTI_002 prefetchsize 1024;
alter tablespace CSTI_003 prefetchsize 1024;
alter tablespace CSTI_004 prefetchsize 1024;
alter tablespace CSTI_005 prefetchsize 1024;
alter tablespace CSTI_006 prefetchsize 1024;
alter tablespace CSTI_007 prefetchsize 1024;
alter tablespace CST_001 prefetchsize 1024;
alter tablespace CST_002 prefetchsize 1024;
alter tablespace CST_003 prefetchsize 1024;
alter tablespace CST_004 prefetchsize 1024;
alter tablespace CST_005 prefetchsize 1024;
alter tablespace CST_006 prefetchsize 1024;
alter tablespace CST_007 prefetchsize 1024;
alter tablespace DIS prefetchsize 1024;
alter tablespace HST prefetchsize 1024;
alter tablespace ITM prefetchsize 1024;
alter tablespace NEWA_001 prefetchsize 1024;
alter tablespace NEWA_002 prefetchsize 1024;
alter tablespace NEWA_003 prefetchsize 1024;
alter tablespace NEWA_004 prefetchsize 1024;
alter tablespace NEWA_005 prefetchsize 1024;
alter tablespace NEWA_006 prefetchsize 1024;
alter tablespace NEWA_007 prefetchsize 1024;
alter tablespace NEWB_001 prefetchsize 1024;
alter tablespace NEWB_002 prefetchsize 1024;
alter tablespace NEWB_003 prefetchsize 1024;
alter tablespace NEWB_004 prefetchsize 1024;
alter tablespace NEWB_005 prefetchsize 1024;
alter tablespace NEWB_006 prefetchsize 1024;
alter tablespace NEWB_007 prefetchsize 1024;
alter tablespace OLN_001 prefetchsize 1024;
alter tablespace OLN_002 prefetchsize 1024;
alter tablespace OLN_003 prefetchsize 1024;
```

```

alter tablespace OLN_004 prefetchsize 1024;
alter tablespace OLN_005 prefetchsize 1024;
alter tablespace OLN_006 prefetchsize 1024;
alter tablespace OLN_007 prefetchsize 1024;
alter tablespace ORD prefetchsize 1024;
alter tablespace ORDI prefetchsize 1024;
alter tablespace STK_001 prefetchsize 1024;
alter tablespace STK_002 prefetchsize 1024;
alter tablespace STK_003 prefetchsize 1024;
alter tablespace STK_004 prefetchsize 1024;
alter tablespace STK_005 prefetchsize 1024;
alter tablespace STK_006 prefetchsize 1024;
alter tablespace STK_007 prefetchsize 1024;
alter tablespace WAR prefetchsize 1024;
connect reset;

```

crconst_customer.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER1 OFF;
ALTER TABLE CUSTOMER1 DROP CONSTRAINT CUSTOMER1CKC;
ALTER TABLE CUSTOMER1 ADD CONSTRAINT CUSTOMER1CKC
CHECK (C_W_ID BETWEEN 1 AND 1800);
SET INTEGRITY FOR CUSTOMER1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER2 OFF;
ALTER TABLE CUSTOMER2 DROP CONSTRAINT CUSTOMER2CKC;
ALTER TABLE CUSTOMER2 ADD CONSTRAINT CUSTOMER2CKC
CHECK (C_W_ID BETWEEN 1801 AND 3600);
SET INTEGRITY FOR CUSTOMER2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER3 OFF;
ALTER TABLE CUSTOMER3 DROP CONSTRAINT CUSTOMER3CKC;
ALTER TABLE CUSTOMER3 ADD CONSTRAINT CUSTOMER3CKC
CHECK (C_W_ID BETWEEN 3601 AND 5400);
SET INTEGRITY FOR CUSTOMER3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER4 OFF;
ALTER TABLE CUSTOMER4 DROP CONSTRAINT CUSTOMER4CKC;
ALTER TABLE CUSTOMER4 ADD CONSTRAINT CUSTOMER4CKC
CHECK (C_W_ID BETWEEN 5401 AND 7200);
SET INTEGRITY FOR CUSTOMER4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER5 OFF;
ALTER TABLE CUSTOMER5 DROP CONSTRAINT CUSTOMER5CKC;
ALTER TABLE CUSTOMER5 ADD CONSTRAINT CUSTOMER5CKC
CHECK (C_W_ID BETWEEN 7201 AND 9000);
SET INTEGRITY FOR CUSTOMER5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER6 OFF;
ALTER TABLE CUSTOMER6 DROP CONSTRAINT CUSTOMER6CKC;
ALTER TABLE CUSTOMER6 ADD CONSTRAINT CUSTOMER6CKC
CHECK (C_W_ID BETWEEN 9001 AND 10800);
SET INTEGRITY FOR CUSTOMER6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER7 OFF;
ALTER TABLE CUSTOMER7 DROP CONSTRAINT CUSTOMER7CKC;
ALTER TABLE CUSTOMER7 ADD CONSTRAINT CUSTOMER7CKC
CHECK (C_W_ID >= 10801);
SET INTEGRITY FOR CUSTOMER7 ALL IMMEDIATE UNCHECKED;
connect reset;

```

crconst_new_ordera.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA1 OFF;
ALTER TABLE NEW_ORDERA1 DROP CONSTRAINT
NEW_ORDERA1CKC;
ALTER TABLE NEW_ORDERA1 ADD CONSTRAINT
NEW_ORDERA1CKC CHECK ((NO_W_ID BETWEEN 1 AND 1800) AND
(NO_O_ID <= 3643));
SET INTEGRITY FOR NEW_ORDERA1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA2 OFF;
ALTER TABLE NEW_ORDERA2 DROP CONSTRAINT
NEW_ORDERA2CKC;
ALTER TABLE NEW_ORDERA2 ADD CONSTRAINT
NEW_ORDERA2CKC CHECK ((NO_W_ID BETWEEN 1801 AND 3600)
AND (NO_O_ID <= 3643));
SET INTEGRITY FOR NEW_ORDERA2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA3 OFF;
ALTER TABLE NEW_ORDERA3 DROP CONSTRAINT
NEW_ORDERA3CKC;
ALTER TABLE NEW_ORDERA3 ADD CONSTRAINT
NEW_ORDERA3CKC CHECK ((NO_W_ID BETWEEN 3601 AND 5400)
AND (NO_O_ID <= 3643));
SET INTEGRITY FOR NEW_ORDERA3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA4 OFF;
ALTER TABLE NEW_ORDERA4 DROP CONSTRAINT
NEW_ORDERA4CKC;
ALTER TABLE NEW_ORDERA4 ADD CONSTRAINT
NEW_ORDERA4CKC CHECK ((NO_W_ID BETWEEN 5401 AND 7200)
AND (NO_O_ID <= 3643));
SET INTEGRITY FOR NEW_ORDERA4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA5 OFF;
ALTER TABLE NEW_ORDERA5 DROP CONSTRAINT
NEW_ORDERA5CKC;
ALTER TABLE NEW_ORDERA5 ADD CONSTRAINT
NEW_ORDERA5CKC CHECK ((NO_W_ID BETWEEN 7201 AND 9000)
AND (NO_O_ID <= 3643));
SET INTEGRITY FOR NEW_ORDERA5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA6 OFF;
ALTER TABLE NEW_ORDERA6 DROP CONSTRAINT
NEW_ORDERA6CKC;
ALTER TABLE NEW_ORDERA6 ADD CONSTRAINT
NEW_ORDERA6CKC CHECK ((NO_W_ID BETWEEN 9001 AND 10800)
AND (NO_O_ID <= 3643));
SET INTEGRITY FOR NEW_ORDERA6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA7 OFF;
ALTER TABLE NEW_ORDERA7 DROP CONSTRAINT
NEW_ORDERA7CKC;
ALTER TABLE NEW_ORDERA7 ADD CONSTRAINT
NEW_ORDERA7CKC CHECK ((NO_W_ID >= 10801) AND (NO_O_ID <=
3643));
SET INTEGRITY FOR NEW_ORDERA7 ALL IMMEDIATE UNCHECKED;
connect reset;

```

crconst_new_orderb.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB1 OFF;
ALTER TABLE NEW_ORDERB1 DROP CONSTRAINT
NEW_ORDERB1CKC;
ALTER TABLE NEW_ORDERB1 ADD CONSTRAINT
NEW_ORDERB1CKC CHECK ((NO_W_ID BETWEEN 1 AND 1800) AND
(NO_O_ID >= 3644));
SET INTEGRITY FOR NEW_ORDERB1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB2 OFF;
ALTER TABLE NEW_ORDERB2 DROP CONSTRAINT
NEW_ORDERB2CKC;
ALTER TABLE NEW_ORDERB2 ADD CONSTRAINT
NEW_ORDERB2CKC CHECK ((NO_W_ID BETWEEN 1801 AND 3600)
AND (NO_O_ID >= 3644));
SET INTEGRITY FOR NEW_ORDERB2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB3 OFF;
ALTER TABLE NEW_ORDERB3 DROP CONSTRAINT
NEW_ORDERB3CKC;
ALTER TABLE NEW_ORDERB3 ADD CONSTRAINT
NEW_ORDERB3CKC CHECK ((NO_W_ID BETWEEN 3601 AND 5400)
AND (NO_O_ID >= 3644));
SET INTEGRITY FOR NEW_ORDERB3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB4 OFF;
ALTER TABLE NEW_ORDERB4 DROP CONSTRAINT
NEW_ORDERB4CKC;
ALTER TABLE NEW_ORDERB4 ADD CONSTRAINT
NEW_ORDERB4CKC CHECK ((NO_W_ID BETWEEN 5401 AND 7200)
AND (NO_O_ID >= 3644));
SET INTEGRITY FOR NEW_ORDERB4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB5 OFF;
ALTER TABLE NEW_ORDERB5 DROP CONSTRAINT
NEW_ORDERB5CKC;
ALTER TABLE NEW_ORDERB5 ADD CONSTRAINT
NEW_ORDERB5CKC CHECK ((NO_W_ID BETWEEN 7201 AND 9000)
AND (NO_O_ID >= 3644));
SET INTEGRITY FOR NEW_ORDERB5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB6 OFF;
ALTER TABLE NEW_ORDERB6 DROP CONSTRAINT
NEW_ORDERB6CKC;
ALTER TABLE NEW_ORDERB6 ADD CONSTRAINT
NEW_ORDERB6CKC CHECK ((NO_W_ID BETWEEN 9001 AND 10800)
AND (NO_O_ID >= 3644));
SET INTEGRITY FOR NEW_ORDERB6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB7 OFF;
ALTER TABLE NEW_ORDERB7 DROP CONSTRAINT
NEW_ORDERB7CKC;
ALTER TABLE NEW_ORDERB7 ADD CONSTRAINT
NEW_ORDERB7CKC CHECK ((NO_W_ID >= 10801) AND (NO_O_ID >=
3644));
SET INTEGRITY FOR NEW_ORDERB7 ALL IMMEDIATE UNCHECKED;
connect reset;
```

crconst_order_line.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE1 OFF;
ALTER TABLE ORDER_LINE1 DROP CONSTRAINT
ORDER_LINE1CKC;
ALTER TABLE ORDER_LINE1 ADD CONSTRAINT ORDER_LINE1CKC
CHECK (OL_W_ID BETWEEN 1 AND 1800);
SET INTEGRITY FOR ORDER_LINE1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE2 OFF;
ALTER TABLE ORDER_LINE2 DROP CONSTRAINT
ORDER_LINE2CKC;
ALTER TABLE ORDER_LINE2 ADD CONSTRAINT ORDER_LINE2CKC
CHECK (OL_W_ID BETWEEN 1801 AND 3600);
SET INTEGRITY FOR ORDER_LINE2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE3 OFF;
ALTER TABLE ORDER_LINE3 DROP CONSTRAINT
ORDER_LINE3CKC;
ALTER TABLE ORDER_LINE3 ADD CONSTRAINT ORDER_LINE3CKC
CHECK (OL_W_ID BETWEEN 3601 AND 5400);
SET INTEGRITY FOR ORDER_LINE3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE4 OFF;
ALTER TABLE ORDER_LINE4 DROP CONSTRAINT
ORDER_LINE4CKC;
ALTER TABLE ORDER_LINE4 ADD CONSTRAINT ORDER_LINE4CKC
CHECK (OL_W_ID BETWEEN 5401 AND 7200);
SET INTEGRITY FOR ORDER_LINE4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE5 OFF;
ALTER TABLE ORDER_LINE5 DROP CONSTRAINT
ORDER_LINE5CKC;
ALTER TABLE ORDER_LINE5 ADD CONSTRAINT ORDER_LINE5CKC
CHECK (OL_W_ID BETWEEN 7201 AND 9000);
SET INTEGRITY FOR ORDER_LINE5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE6 OFF;
ALTER TABLE ORDER_LINE6 DROP CONSTRAINT
ORDER_LINE6CKC;
ALTER TABLE ORDER_LINE6 ADD CONSTRAINT ORDER_LINE6CKC
CHECK (OL_W_ID BETWEEN 9001 AND 10800);
SET INTEGRITY FOR ORDER_LINE6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE7 OFF;
ALTER TABLE ORDER_LINE7 DROP CONSTRAINT
ORDER_LINE7CKC;
ALTER TABLE ORDER_LINE7 ADD CONSTRAINT ORDER_LINE7CKC
CHECK (OL_W_ID >= 10801);
SET INTEGRITY FOR ORDER_LINE7 ALL IMMEDIATE UNCHECKED;
connect reset;
```

crconst_stock.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK1 OFF;
ALTER TABLE STOCK1 DROP CONSTRAINT STOCK1CKC;
ALTER TABLE STOCK1 ADD CONSTRAINT STOCK1CKC CHECK
(S_W_ID BETWEEN 1 AND 1800);
SET INTEGRITY FOR STOCK1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK2 OFF;
```

```

ALTER TABLE STOCK2 DROP CONSTRAINT STOCK2CKC;
ALTER TABLE STOCK2 ADD CONSTRAINT STOCK2CKC CHECK
(S_W_ID BETWEEN 1801 AND 3600);
SET INTEGRITY FOR STOCK2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK3 OFF;
ALTER TABLE STOCK3 DROP CONSTRAINT STOCK3CKC;
ALTER TABLE STOCK3 ADD CONSTRAINT STOCK3CKC CHECK
(S_W_ID BETWEEN 3601 AND 5400);
SET INTEGRITY FOR STOCK3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK4 OFF;
ALTER TABLE STOCK4 DROP CONSTRAINT STOCK4CKC;
ALTER TABLE STOCK4 ADD CONSTRAINT STOCK4CKC CHECK
(S_W_ID BETWEEN 5401 AND 7200);
SET INTEGRITY FOR STOCK4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK5 OFF;
ALTER TABLE STOCK5 DROP CONSTRAINT STOCK5CKC;
ALTER TABLE STOCK5 ADD CONSTRAINT STOCK5CKC CHECK
(S_W_ID BETWEEN 7201 AND 9000);
SET INTEGRITY FOR STOCK5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK6 OFF;
ALTER TABLE STOCK6 DROP CONSTRAINT STOCK6CKC;
ALTER TABLE STOCK6 ADD CONSTRAINT STOCK6CKC CHECK
(S_W_ID BETWEEN 9001 AND 10800);
SET INTEGRITY FOR STOCK6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK7 OFF;
ALTER TABLE STOCK7 DROP CONSTRAINT STOCK7CKC;
ALTER TABLE STOCK7 ADD CONSTRAINT STOCK7CKC CHECK
(S_W_ID >= 10801);
SET INTEGRITY FOR STOCK7 ALL IMMEDIATE UNCHECKED;
connect reset;

```

cridx_cust_idxb.ddl

```

connect to TPCC in share mode;
DROP INDEX CUST_IDXB1;
CREATE INDEX CUST_IDXB1
    ON CUSTOMER1(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB2;
CREATE INDEX CUST_IDXB2
    ON CUSTOMER2(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB3;
CREATE INDEX CUST_IDXB3
    ON CUSTOMER3(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB4;
CREATE INDEX CUST_IDXB4
    ON CUSTOMER4(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;

```

```

DROP INDEX CUST_IDXB5;
CREATE INDEX CUST_IDXB5
    ON CUSTOMER5(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB6;
CREATE INDEX CUST_IDXB6
    ON CUSTOMER6(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB7;
CREATE INDEX CUST_IDXB7
    ON CUSTOMER7(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;

```

cridx_ordr_idxb.ddl

```

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB;
CREATE INDEX ORDR_IDXB
    ON ORDERS(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

```

crtb_customer.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER1;
CREATE TABLE CUSTOMER1
(
    C_ID      INTEGER      NOT NULL,
    C_STATE  CHAR(2)      NOT NULL,
    C_ZIP    CHAR(9)      NOT NULL,
    C_PHONE  CHAR(16)     NOT NULL,
    C_SINCE  BIGINT       NOT NULL,
    C_CREDIT LIM BIGINT   NOT NULL,
    C_MIDDLE CHAR(2)      NOT NULL,
    C_CREDIT CHAR(2)      NOT NULL,
    C_DISCOUNT INTEGER   NOT NULL,
    C_DATA   VARCHAR(500) NOT NULL,
    C_LAST   VARCHAR(16)  NOT NULL,
    C_FIRST  VARCHAR(16)  NOT NULL,
    C_STREET_1 VARCHAR(20) NOT NULL,
    C_STREET_2 VARCHAR(20) NOT NULL,
    C_CITY   VARCHAR(20)  NOT NULL,
    C_D_ID   SMALLINT     NOT NULL,
    C_W_ID   INTEGER      NOT NULL,
    C_DELIVERY_CNT INTEGER NOT NULL,
    C_BALANCE BIGINT      NOT NULL,
    C_YTD_PAYMENT BIGINT  NOT NULL,
    C_PAYMENT_CNT INTEGER NOT NULL
)
IN CST_001
INDEX IN CSTI_001
ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 1 ENDING AT 1800,
    C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER2;
CREATE TABLE CUSTOMER2

```

```
(
C_ID      INTEGER    NOT NULL,
C_STATE   CHAR(2)    NOT NULL,
C_ZIP     CHAR(9)    NOT NULL,
C_PHONE   CHAR(16)   NOT NULL,
C_SINCE   BIGINT     NOT NULL,
C_CREDIT_LIM BIGINT  NOT NULL,
C_MIDDLE  CHAR(2)    NOT NULL,
C_CREDIT  CHAR(2)    NOT NULL,
C_DISCOUNT INTEGER  NOT NULL,
C_DATA    VARCHAR(500) NOT NULL,
C_LAST    VARCHAR(16) NOT NULL,
C_FIRST   VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY    VARCHAR(20) NOT NULL,
C_D_ID    SMALLINT   NOT NULL,
C_W_ID    INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN CST_002
INDEX IN CSTI_002
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 1801 ENDING AT 3600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
```

connect reset;

connect to TPCC in share mode;

DROP TABLE CUSTOMER3;

CREATE TABLE CUSTOMER3

```
(
C_ID      INTEGER    NOT NULL,
C_STATE   CHAR(2)    NOT NULL,
C_ZIP     CHAR(9)    NOT NULL,
C_PHONE   CHAR(16)   NOT NULL,
C_SINCE   BIGINT     NOT NULL,
C_CREDIT_LIM BIGINT  NOT NULL,
C_MIDDLE  CHAR(2)    NOT NULL,
C_CREDIT  CHAR(2)    NOT NULL,
C_DISCOUNT INTEGER  NOT NULL,
C_DATA    VARCHAR(500) NOT NULL,
C_LAST    VARCHAR(16) NOT NULL,
C_FIRST   VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY    VARCHAR(20) NOT NULL,
C_D_ID    SMALLINT   NOT NULL,
C_W_ID    INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN CST_003
INDEX IN CSTI_003
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 3601 ENDING AT 5400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
```

connect reset;

connect to TPCC in share mode;

DROP TABLE CUSTOMER4;
CREATE TABLE CUSTOMER4

```
(
C_ID      INTEGER    NOT NULL,
C_STATE   CHAR(2)    NOT NULL,
C_ZIP     CHAR(9)    NOT NULL,
C_PHONE   CHAR(16)   NOT NULL,
C_SINCE   BIGINT     NOT NULL,
C_CREDIT_LIM BIGINT  NOT NULL,
C_MIDDLE  CHAR(2)    NOT NULL,
C_CREDIT  CHAR(2)    NOT NULL,
C_DISCOUNT INTEGER  NOT NULL,
C_DATA    VARCHAR(500) NOT NULL,
C_LAST    VARCHAR(16) NOT NULL,
C_FIRST   VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY    VARCHAR(20) NOT NULL,
C_D_ID    SMALLINT   NOT NULL,
C_W_ID    INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN CST_004
INDEX IN CSTI_004
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 5401 ENDING AT 7200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
```

connect reset;

connect to TPCC in share mode;

DROP TABLE CUSTOMER5;

CREATE TABLE CUSTOMER5

```
(
C_ID      INTEGER    NOT NULL,
C_STATE   CHAR(2)    NOT NULL,
C_ZIP     CHAR(9)    NOT NULL,
C_PHONE   CHAR(16)   NOT NULL,
C_SINCE   BIGINT     NOT NULL,
C_CREDIT_LIM BIGINT  NOT NULL,
C_MIDDLE  CHAR(2)    NOT NULL,
C_CREDIT  CHAR(2)    NOT NULL,
C_DISCOUNT INTEGER  NOT NULL,
C_DATA    VARCHAR(500) NOT NULL,
C_LAST    VARCHAR(16) NOT NULL,
C_FIRST   VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY    VARCHAR(20) NOT NULL,
C_D_ID    SMALLINT   NOT NULL,
C_W_ID    INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN CST_005
INDEX IN CSTI_005
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 7201 ENDING AT 9000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER6;
CREATE TABLE CUSTOMER6
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT  NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT INTEGER  NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT  NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN CST_006
INDEX IN CSTI_006
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 9001 ENDING AT 10800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER7;
CREATE TABLE CUSTOMER7
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT  NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT INTEGER  NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT  NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN CST_007
INDEX IN CSTI_007
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 10801 ENDING AT 12600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

)
ALLOW OVERFLOW;
connect reset;

```

crtb_item.ddl

```

connect to TPCC in share mode;
DROP TABLE ITEM;
CREATE TABLE ITEM
(
  I_NAME     CHAR(24)  NOT NULL,
  I_PRICE    INTEGER   NOT NULL,
  I_DATA     VARCHAR(50) NOT NULL,
  I_IM_ID    INTEGER   NOT NULL,
  I_ID       INTEGER   NOT NULL
)
IN ITM
INDEX IN ITM
ORGANIZE BY KEY SEQUENCE (
  I_ID STARTING FROM 1 ENDING AT 10000
)
ALLOW OVERFLOW;
ALTER TABLE ITEM LOCKSIZE TABLE;
connect reset;

```

crtb_district.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT;
CREATE TABLE DISTRICT
(
  D_NEXT_O_ID INTEGER  NOT NULL,
  D_TAX        INTEGER  NOT NULL,
  D_YTD        BIGINT   NOT NULL,
  D_NAME       CHAR(10) NOT NULL,
  D_STREET_1   CHAR(20) NOT NULL,
  D_STREET_2   CHAR(20) NOT NULL,
  D_CITY       CHAR(20) NOT NULL,
  D_STATE      CHAR(2)  NOT NULL,
  D_ZIP        CHAR(9)  NOT NULL,
  D_ID         SMALLINT NOT NULL,
  D_W_ID       INTEGER  NOT NULL
)
IN DIS
INDEX IN DIS
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 1 ENDING AT 12600
)
ALLOW OVERFLOW;
connect reset;

```

crtb_orders.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS;
CREATE TABLE ORDERS
(
  O_C_ID    INTEGER  NOT NULL,
  O_ENTRY_D BIGINT   NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID      INTEGER  NOT NULL,
  O_W_ID    INTEGER  NOT NULL,

```



```

        O_D_ID    SMALLINT NOT NULL
    )
    IN ORD
    INDEX IN ORD
    ORGANIZE BY KEY SEQUENCE (
        O_ID STARTING FROM 0 ENDING AT 3643,
        O_W_ID STARTING FROM 1 ENDING AT 12600,
        O_D_ID STARTING FROM 1 ENDING AT 10
    )
    ALLOW OVERFLOW;
connect reset;

```

crtb_order_line.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE1;
CREATE TABLE ORDER_LINE1
(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT     INTEGER NOT NULL,
    OL_I_ID       INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY   SMALLINT NOT NULL,
    OL_DIST_INFO  CHAR(24) NOT NULL,
    OL_O_ID       INTEGER NOT NULL,
    OL_D_ID       SMALLINT NOT NULL,
    OL_W_ID       INTEGER NOT NULL,
    OL_NUMBER     SMALLINT NOT NULL
)
IN OLN_001
INDEX IN OLN_001
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 1 ENDING AT 1800,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3643,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE2;
CREATE TABLE ORDER_LINE2
(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT     INTEGER NOT NULL,
    OL_I_ID       INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY   SMALLINT NOT NULL,
    OL_DIST_INFO  CHAR(24) NOT NULL,
    OL_O_ID       INTEGER NOT NULL,
    OL_D_ID       SMALLINT NOT NULL,
    OL_W_ID       INTEGER NOT NULL,
    OL_NUMBER     SMALLINT NOT NULL
)
IN OLN_002
INDEX IN OLN_002
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 1801 ENDING AT 3600,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3643,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE3;
CREATE TABLE ORDER_LINE3

```

```

(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT     INTEGER NOT NULL,
    OL_I_ID       INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY   SMALLINT NOT NULL,
    OL_DIST_INFO  CHAR(24) NOT NULL,
    OL_O_ID       INTEGER NOT NULL,
    OL_D_ID       SMALLINT NOT NULL,
    OL_W_ID       INTEGER NOT NULL,
    OL_NUMBER     SMALLINT NOT NULL
)
IN OLN_003
INDEX IN OLN_003
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 3601 ENDING AT 5400,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3643,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE4;
CREATE TABLE ORDER_LINE4
(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT     INTEGER NOT NULL,
    OL_I_ID       INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY   SMALLINT NOT NULL,
    OL_DIST_INFO  CHAR(24) NOT NULL,
    OL_O_ID       INTEGER NOT NULL,
    OL_D_ID       SMALLINT NOT NULL,
    OL_W_ID       INTEGER NOT NULL,
    OL_NUMBER     SMALLINT NOT NULL
)
IN OLN_004
INDEX IN OLN_004
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 5401 ENDING AT 7200,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3643,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE5;
CREATE TABLE ORDER_LINE5
(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT     INTEGER NOT NULL,
    OL_I_ID       INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY   SMALLINT NOT NULL,
    OL_DIST_INFO  CHAR(24) NOT NULL,
    OL_O_ID       INTEGER NOT NULL,
    OL_D_ID       SMALLINT NOT NULL,
    OL_W_ID       INTEGER NOT NULL,
    OL_NUMBER     SMALLINT NOT NULL
)
IN OLN_005
INDEX IN OLN_005
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 7201 ENDING AT 9000,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3643,

```

```

        OL_NUMBER STARTING FROM 1 ENDING AT 15
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE6;
CREATE TABLE ORDER_LINE6
(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_ID INTEGER NOT NULL,
    OL_D_ID SMALLINT NOT NULL,
    OL_W_ID INTEGER NOT NULL,
    OL_NUMBER SMALLINT NOT NULL
)
IN OLN_006
INDEX IN OLN_006
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 9001 ENDING AT 10800,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3643,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE7;
CREATE TABLE ORDER_LINE7
(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_ID INTEGER NOT NULL,
    OL_D_ID SMALLINT NOT NULL,
    OL_W_ID INTEGER NOT NULL,
    OL_NUMBER SMALLINT NOT NULL
)
IN OLN_007
INDEX IN OLN_007
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 10801 ENDING AT 12600,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3643,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;

```

crtb_new_ordera.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA1;
CREATE TABLE NEW_ORDERA1
(
    NO_O_ID INTEGER NOT NULL,
    NO_D_ID SMALLINT NOT NULL,
    NO_W_ID INTEGER NOT NULL
)

```

```

IN NEWA_001
INDEX IN NEWA_001
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 1 ENDING AT
1800,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3643
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA2;
CREATE TABLE NEW_ORDERA2
(
    NO_O_ID INTEGER NOT NULL,
    NO_D_ID SMALLINT NOT NULL,
    NO_W_ID INTEGER NOT NULL
)
IN NEWA_002
INDEX IN NEWA_002
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 1801 ENDING
AT 3600,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3643
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA3;
CREATE TABLE NEW_ORDERA3
(
    NO_O_ID INTEGER NOT NULL,
    NO_D_ID SMALLINT NOT NULL,
    NO_W_ID INTEGER NOT NULL
)
IN NEWA_003
INDEX IN NEWA_003
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 3601 ENDING
AT 5400,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3643
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA4;
CREATE TABLE NEW_ORDERA4
(
    NO_O_ID INTEGER NOT NULL,
    NO_D_ID SMALLINT NOT NULL,

```

```

        NO_W_ID    INTEGER    NOT NULL
    )
    IN NEWA_004
    INDEX IN NEWA_004
    ORGANIZE BY KEY SEQUENCE (
        NO_W_ID STARTING FROM 5401 ENDING
AT 7200,
        NO_D_ID STARTING FROM 1 ENDING AT 10,
        NO_O_ID STARTING FROM 1900 ENDING AT
3643
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA5;
CREATE TABLE NEW_ORDERA5
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN NEWA_005
INDEX IN NEWA_005
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 7201 ENDING
AT 9000,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3643
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA6;
CREATE TABLE NEW_ORDERA6
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN NEWA_006
INDEX IN NEWA_006
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 9001 ENDING
AT 10800,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3643
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA7;
CREATE TABLE NEW_ORDERA7
(

```

```

        NO_O_ID    INTEGER    NOT NULL,
        NO_D_ID    SMALLINT   NOT NULL,
        NO_W_ID    INTEGER    NOT NULL
    )
    IN NEWA_007
    INDEX IN NEWA_007
    ORGANIZE BY KEY SEQUENCE (
        NO_W_ID STARTING FROM 10801 ENDING
AT 12600,
        NO_D_ID STARTING FROM 1 ENDING AT 10,
        NO_O_ID STARTING FROM 1900 ENDING AT
3643
    )
    ALLOW OVERFLOW;
connect reset;

crtb_new_orderb.ddl

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB1;
CREATE TABLE NEW_ORDERB1
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN NEWB_001
INDEX IN NEWB_001
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 1 ENDING AT 1800,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3644 ENDING AT 5387
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB2;
CREATE TABLE NEW_ORDERB2
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN NEWB_002
INDEX IN NEWB_002
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 1801 ENDING AT 3600,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3644 ENDING AT 5387
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB3;
CREATE TABLE NEW_ORDERB3
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN NEWB_003
INDEX IN NEWB_003
ORGANIZE BY KEY SEQUENCE (

```

```

NO_W_ID STARTING FROM 3601 ENDING AT 5400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3644 ENDING AT 5387
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB4;
CREATE TABLE NEW_ORDERB4
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN NEWB_004
INDEX IN NEWB_004
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 5401 ENDING AT 7200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3644 ENDING AT 5387
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB5;
CREATE TABLE NEW_ORDERB5
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN NEWB_005
INDEX IN NEWB_005
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 7201 ENDING AT 9000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3644 ENDING AT 5387
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB6;
CREATE TABLE NEW_ORDERB6
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN NEWB_006
INDEX IN NEWB_006
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 9001 ENDING AT 10800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3644 ENDING AT 5387
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB7;
CREATE TABLE NEW_ORDERB7
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN NEWB_007
INDEX IN NEWB_007
ORGANIZE BY KEY SEQUENCE (

```

```

NO_W_ID STARTING FROM 10801 ENDING AT 12600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3644 ENDING AT 5387
)
ALLOW OVERFLOW;
connect reset;

crtb_stock.ddl
connect to TPCC in share mode;
DROP TABLE STOCK1;
CREATE TABLE STOCK1
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01    CHAR(24) NOT NULL,
S_DIST_02    CHAR(24) NOT NULL,
S_DIST_03    CHAR(24) NOT NULL,
S_DIST_04    CHAR(24) NOT NULL,
S_DIST_05    CHAR(24) NOT NULL,
S_DIST_06    CHAR(24) NOT NULL,
S_DIST_07    CHAR(24) NOT NULL,
S_DIST_08    CHAR(24) NOT NULL,
S_DIST_09    CHAR(24) NOT NULL,
S_DIST_10    CHAR(24) NOT NULL,
S_I_ID       INTEGER NOT NULL,
S_W_ID       INTEGER NOT NULL
)
IN STK_001
INDEX IN STK_001
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 1 ENDING AT 1800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK2;
CREATE TABLE STOCK2
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01    CHAR(24) NOT NULL,
S_DIST_02    CHAR(24) NOT NULL,
S_DIST_03    CHAR(24) NOT NULL,
S_DIST_04    CHAR(24) NOT NULL,
S_DIST_05    CHAR(24) NOT NULL,
S_DIST_06    CHAR(24) NOT NULL,
S_DIST_07    CHAR(24) NOT NULL,
S_DIST_08    CHAR(24) NOT NULL,
S_DIST_09    CHAR(24) NOT NULL,
S_DIST_10    CHAR(24) NOT NULL,
S_I_ID       INTEGER NOT NULL,
S_W_ID       INTEGER NOT NULL
)
IN STK_002
INDEX IN STK_002
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 1801 ENDING AT 3600
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK3;
CREATE TABLE STOCK3
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
IN STK_003
INDEX IN STK_003
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 3601 ENDING AT 5400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK4;
CREATE TABLE STOCK4
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
IN STK_004
INDEX IN STK_004
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 5401 ENDING AT 7200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK5;
CREATE TABLE STOCK5
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,

```

```

  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
IN STK_005
INDEX IN STK_005
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 7201 ENDING AT 9000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK6;
CREATE TABLE STOCK6
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
IN STK_006
INDEX IN STK_006
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 9001 ENDING AT 10800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK7;
CREATE TABLE STOCK7
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,

```

```

S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN STK_007
INDEX IN STK_007
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 10801 ENDING AT 12600
)
ALLOW OVERFLOW;

```

connect reset;

crtb_history.ddl

connect to TPCC in share mode;

DROP TABLE HISTORY;

CREATE TABLE HISTORY

```

(
H_C_ID INTEGER NOT NULL,
H_C_D_ID SMALLINT NOT NULL,
H_C_W_ID INTEGER NOT NULL,
H_D_ID SMALLINT NOT NULL,
H_W_ID INTEGER NOT NULL,
H_DATE BIGINT NOT NULL,
H_AMOUNT INTEGER NOT NULL,
H_DATA CHAR(24) NOT NULL
)
IN HST
INDEX IN HST;

```

ALTER TABLE HISTORY APPEND ON;

connect reset;

crtb_warehouse.ddl

connect to TPCC in share mode;

DROP TABLE WAREHOUSE;

CREATE TABLE WAREHOUSE

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN WAR
INDEX IN WAR
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 1 ENDING AT 12600
)
ALLOW OVERFLOW;

```

connect reset;

crvw_customer.ddl

connect to TPCC in share mode;

DROP VIEW CUSTOMER;

CREATE VIEW CUSTOMER

```

(C_ID,
C_STATE,
C_ZIP,
C_PHONE,
C_SINCE,
C_CREDIT_LIM,
C_MIDDLE,
C_CREDIT,
C_DISCOUNT,
C_DATA,
C_LAST,
C_FIRST,
C_STREET_1,
C_STREET_2,
C_CITY,
C_D_ID,
C_W_ID,
C_DELIVERY_CNT,
C_BALANCE,
C_YTD_PAYMENT,
C_PAYMENT_CNT
) AS SELECT * FROM CUSTOMER1 UNION ALL

```

SELECT * FROM CUSTOMER2 UNION ALL

SELECT * FROM CUSTOMER3 UNION ALL

SELECT * FROM CUSTOMER4 UNION ALL

SELECT * FROM CUSTOMER5 UNION ALL

SELECT * FROM CUSTOMER6 UNION ALL

SELECT * FROM CUSTOMER7

WITH ROW MOVEMENT;

COMMIT WORK;

connect reset;

crvw_new_order.ddl

connect to TPCC in share mode;

DROP VIEW NEW_ORDER;

CREATE VIEW NEW_ORDER

```

(NO_O_ID,
NO_D_ID,
NO_W_ID
) AS SELECT * FROM NEW_ORDERA1 UNION ALL

```

SELECT * FROM NEW_ORDERA2 UNION ALL

SELECT * FROM NEW_ORDERA3 UNION ALL

SELECT * FROM NEW_ORDERA4 UNION ALL

SELECT * FROM NEW_ORDERA5 UNION ALL

SELECT * FROM NEW_ORDERA6 UNION ALL

SELECT * FROM NEW_ORDERA7 UNION ALL

SELECT * FROM NEW_ORDERB1 UNION ALL

SELECT * FROM NEW_ORDERB2 UNION ALL

SELECT * FROM NEW_ORDERB3 UNION ALL

SELECT * FROM NEW_ORDERB4 UNION ALL

SELECT * FROM NEW_ORDERB5 UNION ALL

SELECT * FROM NEW_ORDERB6 UNION ALL

SELECT * FROM NEW_ORDERB7

WITH ROW MOVEMENT;

COMMIT WORK;

connect reset;

crvw_order_line.ddl

connect to TPCC in share mode;

DROP VIEW ORDER_LINE;

CREATE VIEW ORDER_LINE

```

(OL_DELIVERY_D,
OL_AMOUNT,
OL_I_ID,

```

```

        OL_SUPPLY_W_ID,
        OL_QUANTITY,
        OL_DIST_INFO,
        OL_O_ID,
        OL_D_ID,
        OL_W_ID,
        OL_NUMBER
    ) AS SELECT * FROM ORDER_LINE1 UNION ALL
SELECT * FROM ORDER_LINE2 UNION ALL
SELECT * FROM ORDER_LINE3 UNION ALL
SELECT * FROM ORDER_LINE4 UNION ALL
SELECT * FROM ORDER_LINE5 UNION ALL
SELECT * FROM ORDER_LINE6 UNION ALL
SELECT * FROM ORDER_LINE7
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

crvw_stock.ddl

```

connect to TPCC in share mode;
DROP VIEW STOCK;
CREATE VIEW STOCK
    (S_REMOTE_CNT,
     S_QUANTITY,
     S_ORDER_CNT,
     S_YTD,
     S_DATA,
     S_DIST_01,
     S_DIST_02,
     S_DIST_03,
     S_DIST_04,
     S_DIST_05,
     S_DIST_06,
     S_DIST_07,
     S_DIST_08,
     S_DIST_09,
     S_DIST_10,
     S_I_ID,
     S_W_ID
    ) AS SELECT * FROM STOCK1 UNION ALL
SELECT * FROM STOCK2 UNION ALL
SELECT * FROM STOCK3 UNION ALL
SELECT * FROM STOCK4 UNION ALL
SELECT * FROM STOCK5 UNION ALL
SELECT * FROM STOCK6 UNION ALL
SELECT * FROM STOCK7
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

gen_customer.bat

```

C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 1 1800 -f1
C:\flats\flat_001\customer_001_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 1801 3600 -f1
C:\flats\flat_002\customer_002_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 3601 5400 -f1
C:\flats\flat_003\customer_003_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 5401 7200 -f1
C:\flats\flat_004\customer_004_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 7201 9000 -f1
C:\flats\flat_005\customer_005_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 9001 10800 -f1
C:\flats\flat_006\customer_006_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 10801 12600 -f1
C:\flats\flat_007\customer_007_1.dat

```

gen_district.bat

```

C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 4 -r 1 12600 -f1
C:\flats\flat\district_1.dat

```

gen_history.bat

```

C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 8 -r 1 3150 -f1
T:\flats\flat\history_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 8 -r 3151 6300 -f1
T:\flats\flat\history_2.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 8 -r 6301 9450 -f1
T:\flats\flat\history_3.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 8 -r 9451 12600 -f1
T:\flats\flat\history_4.dat

```

gen_item.bat

```

C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 5 -f1
C:\flats\flat\item_1.dat

```

gen_new_order.bat

```

C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 1 1800 -f1
C:\flats\flat_001\neworder_001_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 1801 3600 -f1
C:\flats\flat_002\neworder_002_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 3601 5400 -f1
C:\flats\flat_003\neworder_003_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 5401 7200 -f1
C:\flats\flat_004\neworder_004_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 7201 9000 -f1
C:\flats\flat_005\neworder_005_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 9001 10800 -f1
C:\flats\flat_006\neworder_006_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 10801 12600 -f1
C:\flats\flat_007\neworder_007_1.dat

```

gen_orders.bat

```

C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 1 1800 -f1
T:\flats\flat\orders_1.dat -f2 C:\flats\flat_001\orderline_001_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 1801 3600 -f1
T:\flats\flat\orders_2.dat -f2 C:\flats\flat_002\orderline_002_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 3601 5400 -f1
T:\flats\flat\orders_3.dat -f2 C:\flats\flat_003\orderline_003_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 5401 7200 -f1
T:\flats\flat\orders_4.dat -f2 C:\flats\flat_004\orderline_004_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 7201 9000 -f1
T:\flats\flat\orders_5.dat -f2 C:\flats\flat_005\orderline_005_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 9001 10800 -f1
T:\flats\flat\orders_6.dat -f2 C:\flats\flat_006\orderline_006_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 10801 12600 -f1
T:\flats\flat\orders_7.dat -f2 C:\flats\flat_007\orderline_007_1.dat

```

gen_stock.bat

```

C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 1 1800 -f1
C:\flats\flat_001\stock_001_1.dat
C:\tpckit\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 1801 3600 -f1
C:\flats\flat_002\stock_002_1.dat

```

```
C:\tpcckit\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 3601 5400 -fl
C:\flats\flat_003\stock_003_1.dat
C:\tpcckit\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 5401 7200 -fl
C:\flats\flat_004\stock_004_1.dat
C:\tpcckit\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 7201 9000 -fl
C:\flats\flat_005\stock_005_1.dat
C:\tpcckit\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 9001 10800 -fl
C:\flats\flat_006\stock_006_1.dat
C:\tpcckit\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 10801 12600 -fl
C:\flats\flat_007\stock_007_1.dat
```

gen_warehouse.bat

```
C:\tpcckit\tpc-c.ibm\dbgen\gendata.exe -t 3 -r 1 12600 -fl
C:\flats\flat\warehouse_1.dat
```

load_customer_all.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_001\customer_001_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
54000000 INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_002\customer_002_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
54000000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_003\customer_003_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
54000000 INSERT INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_004\customer_004_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
54000000 INSERT INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_005\customer_005_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
54000000 INSERT INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_006\customer_006_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
54000000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_007\customer_007_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
54000000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
```

load_district_all.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat\district_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT;
COMMIT WORK;
CONNECT RESET;
```

load_history_all.ddl

```
connect to TPCC in share mode;
LOAD FROM T:\flats\flat\history_1.dat, T:\flats\flat\history_2.dat,
T:\flats\flat\history_3.dat, T:\flats\flat\history_4.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY
NONRECOVERABLE DATA BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
```

load_item_all.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat\item_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 1000 INSERT INTO
ITEM;
COMMIT WORK;
CONNECT RESET;
```

load_new_order_all.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_001\neworder_001_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_002\neworder_002_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_003\neworder_003_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_004\neworder_004_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```



```

IMPORT FROM C:\flats\flat_005\neworder_005_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_006\neworder_006_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_007\neworder_007_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA7;
COMMIT WORK;
CONNECT RESET;

```

load_order_line_all.ddl

```

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_001\orderline_001_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
594000000 INSERT INTO ORDER_LINE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_002\orderline_002_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
594000000 INSERT INTO ORDER_LINE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_003\orderline_003_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
594000000 INSERT INTO ORDER_LINE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_004\orderline_004_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
594000000 INSERT INTO ORDER_LINE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_005\orderline_005_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
594000000 INSERT INTO ORDER_LINE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_006\orderline_006_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
594000000 INSERT INTO ORDER_LINE6;
COMMIT WORK;
CONNECT RESET;

```

```

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_007\orderline_007_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
594000000 INSERT INTO ORDER_LINE7;
COMMIT WORK;
CONNECT RESET;

```

load_orders_all.ddl

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM T:\flats\flat\orders_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM T:\flats\flat\orders_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM T:\flats\flat\orders_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM T:\flats\flat\orders_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM T:\flats\flat\orders_5.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM T:\flats\flat\orders_6.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM T:\flats\flat\orders_7.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;

```

load_stock_all.ddl

```

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_001\stock_001_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 180000000
INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;

```

```

IMPORT FROM C:\flats\flat_002\stock_002_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 180000000
INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_003\stock_003_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 180000000
INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_004\stock_004_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 180000000
INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_005\stock_005_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 180000000
INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_006\stock_006_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 180000000
INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_007\stock_007_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 180000000
INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;

```

load_warehouse_all.ddl

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat\warehouse_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE;
COMMIT WORK;
CONNECT RESET;

```

rnst_customer.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.CUSTOMER1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.CUSTOMER2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.CUSTOMER3 AND INDEXES ALL;

```

```

COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.CUSTOMER4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.CUSTOMER5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.CUSTOMER6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.CUSTOMER7 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

rnst_district.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.DISTRICT AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

rnst_history.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.HISTORY AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

rnst_item.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.ITEM AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

rnst_new_ordera.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERA1 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERA2 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERA3 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERA4 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERA5 AND INDEXES
ALL;

```

COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERA6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERA7 AND INDEXES ALL;
COMMIT WORK;
connect reset;

rnst_new_orderb.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERB1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERB2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERB3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERB4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERB5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERB6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.NEW_ORDERB7 AND INDEXES ALL;
COMMIT WORK;
connect reset;

rnst_order_line.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.ORDER_LINE1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.ORDER_LINE2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.ORDER_LINE3 AND INDEXES ALL;
COMMIT WORK;

connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.ORDER_LINE4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.ORDER_LINE5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.ORDER_LINE6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.ORDER_LINE7 AND INDEXES ALL;
COMMIT WORK;
connect reset;

rnst_orders.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.ORDERS AND INDEXES ALL;
COMMIT WORK;
connect reset;

rnst_stock.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.STOCK1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.STOCK2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.STOCK3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.STOCK4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.STOCK5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.STOCK6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.STOCK7 AND INDEXES ALL;
COMMIT WORK;
connect reset;

rnst_warehouse.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE Administrator.WAREHOUSE AND INDEXES ALL;
COMMIT WORK;
connect reset;


```

if (strcmp(argv[i], "-t") == 0) {
    option = atoi(argv[i+1]);
    i++;
} else if (strcmp(argv[i], "-r") == 0) {
    ware_start = atoi(argv[i+1]);
    ware_end = atoi(argv[i+2]);
    i += 2;
} else if (strcmp(argv[i], "-d") == 0) {
    delim = argv[i+1];
    i++;
} else if ((strcmp(argv[i], "-f") == 0) ||
            (strcmp(argv[i], "-fl") == 0)) {
    outtype1 = IOH_FILE;
    outname1 = argv[i+1];
    i++;
} else if (strcmp(argv[i], "-f2") == 0) {
    outtype2 = IOH_FILE;
    outname2 = argv[i+1];
    i++;
} else if ((strcmp(argv[i], "-p") == 0) ||
            (strcmp(argv[i], "-p1") == 0)) {
    outtype1 = IOH_PIPE;
    outname1 = argv[i+1];
    i++;
} else if (strcmp(argv[i], "-p2") == 0) {
    outtype2 = IOH_PIPE;
    outname2 = argv[i+1];
    i++;
} else if (strcmp(argv[i], "-s") == 0) {
    ScalingReport();
    exit(0);
} else if (strcmp(argv[i], "-q") == 0) {
    quiet_mode = 1;
} else {
    fprintf(stderr, "gendata: Don't understand argument: %s\n", argv[i]);
    exit(-1);
}
}

/*
*****
*/
/* Validate Command Line Arguments */
/*
*****
*/

/* Validate Table Argument */
if (option < 3 || option > 11 || option == 10)
{
    fprintf(stderr, "gendata: Invalid table selected: %d\n", option);
    exit(-1);
}

/* Validate Delimiter Argument */
if (delim == NULL) {
    // default delimiter is used for IMPORT & LOAD, no changes necessary
    using_rctload = 0;
} else if (strlen(delim) == 1 && !isalnum(delim[0]) &&
            delim[0] != '.' && delim[0] != '%')
{
    // user-supplied delimiter used for rctload
    InitFormatStrings(delim[0]);
    using_rctload = 1;
} else {
    fprintf(stderr, "gendata: Invalid delimiter specified: %s\n", delim);
    exit(-1);
}

/* Validate File/Pipe Arguments */
if (option != 9 && outtype1 > 0 && outtype2 > 0)
{
    fprintf(stderr, "gendata: Specifying two output file/pipes allowed only when
generating\norders/orderline.\n");
    exit(-1);
}
if (option == 9 && ((outtype1 == 0) || (outtype2 == 0)))
{
    fprintf(stderr, "gendata: Must specify two output file/pipes when generating
orders/orderline.\n");
    exit(-1);
}
if (outtype1 == 0 || outname1 == NULL || strcmp(outname1, "") == 0)
{
    fprintf(stderr, "gendata: Invalid 1st output file/pipe specified.\n");
    exit(-1);
}
if (option == 9 && (outtype2 == 0 || outname2 == NULL ||
strcmp(outname2, "") == 0))
{
    fprintf(stderr, "gendata: Invalid 2nd output file/pipe specified.\n");
    exit(-1);
}
/* Ensure O/OL flat files are opened in append mode. This is required */
/* because we generate O/OL concurrently. See comments in genload.pl */
/* for further details on why this is necessary. */
if (option == 9)
{
    if (outtype1 == IOH_FILE) outtype1 = IOH_FILE_APPEND;
    if (outtype2 == IOH_FILE) outtype2 = IOH_FILE_APPEND;
}

/* Validate Range Arguments */
if (ware_start <= 0 || ware_start > WAREHOUSES) {
    fprintf(stderr, "gendata: Invalid range starting value: %d\n", ware_start);
    exit(-1);
}
if (ware_end <= 0 || ware_end > WAREHOUSES || ware_end < ware_start) {
    fprintf(stderr, "gendata: Invalid range ending value: %d\n", ware_end);
    exit(-1);
}
}

initialize_random();

/*
*****
*/
/* Generate Data */
/*
*****
*/
switch (option) {
case 3: /* WAREHOUSE */
    gen_ware_tbl();
    break;
case 4: /* DISTRICT */
    gen_dist_tbl();
    break;
case 5: /* ITEM */
    gen_item_tbl();
    break;
case 6: /* STOCK */
    gen_stock_tbl();
    break;
case 7: /* CUSTOMER */
    gen_cust_tbl();
}

```

```

break;
case 8: /* HISTORY */
    gen_hist_tbl();
    break;
case 9: /* ORDERS + ORDER_LINE */
    gen_ordr_tbl();
    break;
case 11: /* NEW_ORDER */
    gen_nu_ord_tbl();
    break;
case 2:
case 10:
default:
    fprintf(stderr, "Error: invalid option = %d \n", (option));
    break;
}
return 0;
}

```

```

/*-----*/
/* generate item table */
/*-----*/

```

```
void gen_item_tbl( void )
```

```

{
    sqlint32 item_num = 0 ;
    sqlint32 item_im_id ;
    char item_name[25] ;
    sqlint32 item_price ;
    char item_data[51] ;

```

```

IOH_NUM numBytes;
ioHandle hnd;
char Buffer[1024];

```

```
timestamp1 = current_time();
```

```

rc = GenericOpen(&hnd, outtype1, outname1);
if (rc != 0) { goto item_done; }

```

```

for(item_num = 1; item_num <= ITEMS; item_num++)
{

```

```

    /* create image id field */
    item_im_id = rand_integer( 1, 10000 ) ;
    /* create name field */
    create_random_a_string( item_name, 14, 24);
    /* create price field */
    item_price = rand_integer( 100, 10000 ) ;
    /* create ORIGINAL field */
    create_a_string_with_original( item_data, 26, 50, 10) ;

```

```

    numBytes = sprintf(Buffer, fmtItem,
        item_name,
        item_price,
        item_data,
        item_im_id,
        item_num);

```

```

    rc = GenericWrite(&hnd, Buffer, numBytes);
    if (rc != 0) { goto item_done; }

```

```
} /* end for... */
```

```
rc = GenericClose(&hnd);
```

```
item_done:
```

```
timestamp2 = current_time();
```

```

elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nITEM table generated in %8.2f seconds.\n\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nITEM table FAILED at (1 %d) after %8.2f
seconds.\n\n", item_num, elapsed);
    fflush(stderr);
}
}

```

```

/*-----*/
/* generate stock table */
/*-----*/

```

```
void gen_stock_tbl( void )
```

```

{
    sqlint32 ware_num = 0 ;
    sqlint32 stock_num = 0 ;
    sqlint32 stock_quant;
    sqlint32 s_ytd;
    sqlint32 s_order_cnt, s_remote_cnt;
    char stock_dist_01[25] ;
    char stock_dist_02[25] ;
    char stock_dist_03[25] ;
    char stock_dist_04[25] ;
    char stock_dist_05[25] ;
    char stock_dist_06[25] ;
    char stock_dist_07[25] ;
    char stock_dist_08[25] ;
    char stock_dist_09[25] ;
    char stock_dist_10[25] ;
    char stock_data[51] ;

```

```

IOH_NUM numBytes;
ioHandle hnd;
char Buffer[1024];

```

```
timestamp1 = current_time();
```

```

rc = GenericOpen(&hnd, outtype1, outname1);
if (rc != 0) { goto stock_done; }

```

```

for (stock_num = 1; stock_num <= STOCK_PER_WAREHOUSE;
stock_num++)
{

```

```

    {
        if (!quiet_mode && (stock_num%500 == 0))
        {
            fprintf(stdout, "STOCK for Item #%d\n", stock_num);
            fflush(stdout);
        }

```

```

    for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
    {

```

```

        stock_quant = rand_integer( 10, 100 ) ;
        create_random_a_string( stock_dist_01, 24, 24);
        create_random_a_string( stock_dist_02, 24, 24);
        create_random_a_string( stock_dist_03, 24, 24);
        create_random_a_string( stock_dist_04, 24, 24);
        create_random_a_string( stock_dist_05, 24, 24);
        create_random_a_string( stock_dist_06, 24, 24);
        create_random_a_string( stock_dist_07, 24, 24);
        create_random_a_string( stock_dist_08, 24, 24);
        create_random_a_string( stock_dist_09, 24, 24);
        create_random_a_string( stock_dist_10, 24, 24);

```

```

        /* create ORIGINAL field */
        create_a_string_with_original( stock_data, 26, 50, 10) ;
        s_ytd = s_order_cnt = s_remote_cnt = 0;

```

```

numBytes = sprintf(Buffer, fmtStock,
    s_remote_cnt,
    stock_quant,
    s_order_cnt,
    s_ytd,
    stock_data,
    stock_dist_01,
    stock_dist_02,
    stock_dist_03,
    stock_dist_04,
    stock_dist_05,
    stock_dist_06,
    stock_dist_07,
    stock_dist_08,
    stock_dist_09,
    stock_dist_10,
    stock_num,
    ware_num);

rc = GenericWrite(&hnd, Buffer, numBytes);
if (rc != 0) { goto stock_done; }

} /* end for... */
} /* end for... */

rc = GenericClose(&hnd);

stock_done:

timestamp2 = current_time();
elapse = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nSTOCK table generated in %8.2f seconds.\n\n", elapse);
    fflush(stdout);
} else {
    fprintf(stderr, "\nSTOCK table FAILED at (S %d W %d) after %8.2f
seconds.\n\n", stock_num, ware_num, elapse);
    fflush(stderr);
}
}

/*-----*/
/* generate warehouse table */
/*-----*/

void gen_ware_tbl( void )
{
    sqlint32 ware_num = 0 ;
    char ware_name[11] ;
    char ware_street_1[21] ;
    char ware_street_2[21] ;
    char ware_city[21] ;
    char ware_state[3] ;
    char ware_zip[10] ;
    sqlint32 ware_tax ;
    sqlint64 ware_YTD ;

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

    timestamp1 = current_time();

    rc = GenericOpen(&hnd, outtype1, outname1);
    if (rc != 0) { goto ware_done; }

    for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
    {
        if (!quiet_mode && ((ware_num % 500) == 0)) {

```

```

        fprintf(stdout, "Warehouse #%d\n", ware_num);
        fflush(stdout);
    }

    create_random_a_string( ware_name, 6, 10); /* create name */
    create_random_a_string( ware_street_1, 10, 20); /* create street 1 */
    create_random_a_string( ware_street_2, 10, 20); /* create street 2 */
    create_random_a_string( ware_city, 10, 20); /* create city */
    create_random_a_string( ware_state, 2, 2); /* create state */
    create_random_n_string( ware_zip, 4, 4); /* create zip */
    strcat(ware_zip, "11111");

    ware_tax = rand_integer(0, 2000);
    ware_YTD = 30000000;

    numBytes = sprintf(Buffer, fmtWare,
        ware_name,
        ware_street_1,
        ware_street_2,
        ware_city,
        ware_state,
        ware_zip,
        ware_tax,
        ware_YTD,
        ware_num);

    rc = GenericWrite(&hnd, Buffer, numBytes);
    if (rc != 0) { goto ware_done; }

} /* end for */

rc = GenericClose(&hnd);

ware_done:

timestamp2 = current_time();
elapse = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nWAREHOUSE table generated in %8.2f
seconds.\n\n", elapse);
    fflush(stdout);
} else {
    fprintf(stderr, "\nWAREHOUSE table FAILED at (W %d) after %8.2f
seconds.\n\n", ware_num, elapse);
    fflush(stderr);
}
}

/*-----*/
/* generate dist table */
/*-----*/

void gen_dist_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    char dist_name[11];
    char dist_street_1[21];
    char dist_street_2[21];
    char dist_city[21];
    char dist_state[3];
    char dist_zip[10];
    sqlint32 dist_tax;
    sqlint32 next_o_id;
    sqlint64 dist_YTD;

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

```

```

next_o_id = CUSTOMERS_PER_DISTRICT + 1;
timestamp1 = current_time();

rc = GenericOpen(&hnd, outtype1, outname1);
if (rc != 0) { goto dist_done; }

for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
{
    if (!quiet_mode) {
        fprintf(stdout, "DISTRICT for Warehouse #%d\n", ware_num);
        fflush(stdout);
    }
    for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE;
dist_num++)
    {
        create_random_a_string( dist_name, 6,10); /* create name */
        create_random_a_string( dist_street_1, 10,20); /* create street 1 */
        create_random_a_string( dist_street_2, 10,20); /* create street 2 */
        create_random_a_string( dist_city, 10,20); /* create city */
        create_random_a_string( dist_state, 2,2); /* create state */
        create_random_n_string( dist_zip, 4,4); /* create zip */
        strcat(dist_zip, "11111");
        dist_tax = rand_integer(0, 2000);
        dist_YTD = 3000000;

        numBytes = sprintf(Buffer, fmtDist,
            next_o_id,
            dist_tax,
            dist_YTD,
            dist_name,
            dist_street_1,
            dist_street_2,
            dist_city,
            dist_state,
            dist_zip,
            dist_num,
            ware_num);

        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto dist_done; }

    } /* end for... */
} /* end for... */

rc = GenericClose(&hnd);

dist_done:

timestamp2 = current_time();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nDISTRICT table generated in %8.2f seconds.\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nDISTRICT table FAILED at (W %d D %d) after %8.2f
seconds.\n\n", ware_num, dist_num, elapsed);
    fflush(stderr);
}
}
}

/*-----*/
/* generate customer table */
/*-----*/

void gen_cust_tbl( void )
{
    sqlint32 ware_num = 0;
    sqlint32 dist_num = 0;

```

```

    sqlint32 cust_num = 0;
    char cust_last[17];
    char cust_middle[3];
    char cust_first[17];
    char cust_street_1[21];
    char cust_street_2[21];
    char cust_city[21];
    char cust_state[3];
    char cust_zip[10];
    char cust_phone[17];
    char cust_credit[3];
    char cust_data[501];
    sqlint32 cust_discount;
    sqlint64 currtmstmp;
    sqlint64 cust_balance;
    sqlint64 cust_YTD_payment;
    sqlint64 cust_credit_lim;

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];
    int len, pos;

    timestamp1 = current_time();

    rc = GenericOpen(&hnd, outtype1, outname1);
    if (rc != 0) { goto cust_done; }

    strcpy(cust_middle, "OE");
    currtmstmp = time(NULL);

    for (cust_num = 1; cust_num <= CUSTOMERS_PER_DISTRICT;
cust_num++)
    {
        if (!quiet_mode) {
            fprintf(stdout, "CUSTOMER #%d:\n", cust_num);
            fflush(stdout);
        }

        for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
        {
            for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE;
dist_num++)
            {
                if (cust_num <= 1000) /* create last name */
                    create_random_last_name( cust_last, cust_num);
                else /* create last name */
                    create_random_last_name( cust_last, 0);
                create_random_a_string( cust_first, 8,16); /* create first name */
                create_random_a_string( cust_street_1, 10,20); /* create street 1 */
                create_random_a_string( cust_street_2, 10,20); /* create street 2 */
                create_random_a_string( cust_city, 10,20); /* create city */
                create_random_a_string( cust_state, 2,2); /* create state */
                create_random_n_string( cust_zip, 4,4); /* create zip */
                strcat(cust_zip, "11111");

                /* create phone number */
                create_random_n_string( cust_phone, 16,16);
                if ( rand_integer( 1, 100 ) <= 10 )
                    strcpy( cust_credit, "BC" );
                else
                    strcpy( cust_credit, "GC" );

                /* create discount rate */
                cust_discount = rand_integer(0, 5000);

                /* create customer data */
                create_random_a_string(cust_data, 300, 500);

```



```

/* pad customer data (only for non-rectload) */
if (using_rectload == 0) {
    for (pos=strlen(cust_data); pos<500; pos++)
        cust_data[pos] = ' ';
    cust_data[500] = '\0';
}

cust_credit_lim = 5000000;
cust_balance = -1000;
cust_YTD_payment = 1000;

if (cust_num == 1 && dist_num == 1 && ware_num == 1)
{
    sprintf(cust_first, "C_LAST_LOAD=%d", C_C_LAST_LOAD);
}

numBytes = sprintf(Buffer, fmtCust,
    cust_num,
    cust_state,
    cust_zip,
    cust_phone,
    currtmstamp,
    cust_credit_lim,
    cust_middle,
    cust_credit,
    cust_discount,
    cust_data,
    cust_last,
    cust_first,
    cust_street_1,
    cust_street_2,
    cust_city,
    dist_num,
    ware_num,
    0,
    cust_balance,
    cust_YTD_payment,
    1);

rc = GenericWrite(&hnd, Buffer, numBytes);
if (rc != 0) { goto cust_done; }

} /* end for district... */
} /* end for warehouse... */
} /* end for customer... */

rc = GenericClose(&hnd);

cust_done:

timestamp2 = current_time();
elapse = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nCUSTOMER table generated in %8.2f
seconds.\n\n", elapse);
    fflush(stdout);
} else {
    fprintf(stderr, "\nCUSTOMER table FAILED at (W %d D %d C %d) after
%8.2f seconds.\n\n", ware_num, dist_num, cust_num, elapse);
    fflush(stderr);
}
}

/*-----*/
/* generate hist table */
/*-----*/
void gen_hist_tbl( void )

```

```

{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 cust_num = 0 ;
    char hist_data[25] ;
    sqlint64 currtmstamp;

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

    timestamp1 = current_time();

    rc = GenericOpen(&hnd, outtype1, outname1);
    if (rc != 0) { goto hist_done; }

    currtmstamp = time(NULL);

    for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
    {
        if (!quiet_mode) {
            fprintf(stdout, "HISTORY for Warehouse #d:\n", ware_num);
            fflush(stdout);
        }
        for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE;
dist_num++)
        {
            for (cust_num = 1; cust_num <= CUSTOMERS_PER_DISTRICT;
cust_num++)
            {
                /* create history data */
                create_random_a_string( hist_data, 12,24) ;

                numBytes = sprintf(Buffer, fmtHist,
                    cust_num,
                    dist_num,
                    ware_num,
                    dist_num,
                    ware_num,
                    currtmstamp,
                    1000,
                    hist_data);

                rc = GenericWrite(&hnd, Buffer, numBytes);
                if (rc != 0) { goto hist_done; }

            } /* end for customer... */
        } /* end for district... */
    } /* end for warehouse... */

    rc = GenericClose(&hnd);

hist_done:

timestamp2 = current_time();
elapse = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nHISTORY table generated in %8.2f seconds.\n\n", elapse);
    fflush(stdout);
} else {
    fprintf(stderr, "\nHISTORY table FAILED at (W %d D %d C %d) after
%8.2f seconds.\n\n", ware_num, dist_num, cust_num, elapse);
    fflush(stderr);
}
}

/*-----*/
/* generate nu_ord table */
/*-----*/

```

```

/*-----*/
void gen_nu_ord_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 nu_ord_id = 0 ;
    int nu_ord_hi ;

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

    /* compute maximum and minimum
       order numbers for this
       district */
    nu_ord_hi = CUSTOMERS_PER_DISTRICT -
    NU_ORDERS_PER_DISTRICT + 1;
    if (nu_ord_hi < 0) {
        nu_ord_hi = CUSTOMERS_PER_DISTRICT -
    (CUSTOMERS_PER_DISTRICT / 3) + 1;
        fprintf(stderr, "\n**** WARNING **** NU_ORDERS_PER_DISTRICT is
    > CUSTOMERS_PER_DISTRICT\n");
        fprintf(stderr, "        Check the values in file lval.h\n");
        fprintf(stderr, "        Loading New-Order with 1/3 of
    CUSTOMERS_PER_DISTRICT\n");
    }

    timestamp1 = current_time();

    rc = GenericOpen(&hnd, outtype1, outname1);
    if (rc != 0) { goto neword_done; }

    for (nu_ord_id = nu_ord_hi;
        nu_ord_id <= CUSTOMERS_PER_DISTRICT;
        nu_ord_id++)
    {
        if (!quiet_mode) {
            fprintf(stdout, "NEW_ORDER for Customer #%d:\n", nu_ord_id);
            fflush(stdout);
        }
        for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
        {
            for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE;
                dist_num++)
            {
                numBytes = sprintf(Buffer, fmtNewOrd,
                    nu_ord_id,
                    dist_num,
                    ware_num);

                rc = GenericWrite(&hnd, Buffer, numBytes);
                if (rc != 0) { goto neword_done; }

            } /* end for... */
        } /* end for... */
    } /* end for... */

    rc = GenericClose(&hnd);

neword_done:

    timestamp2 = current_time();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {
        fprintf(stdout, "\nNEW_ORDER table generated in %8.2f
    seconds.\n\n", elapse);
        fflush(stdout);
    } else {

```

```

        fprintf(stderr, "\nNEW_ORDER table FAILED at (W %d D %d O %d) after
    %8.2f seconds.\n\n", ware_num, dist_num, nu_ord_id, elapse);
        fflush(stderr);
    }
}

/*-----*/
/* generate order and order_line tables */
/*-----*/
void gen_ordr_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 cust_num = 0 ;
    sqlint32 ord_num = 0 ;
    sqlint32 ord_r_carrier_id;
    sqlint32 ord_r_ol_cnt;
    sqlint32 oline_ol_num;
    sqlint32 oline_item_num;

    sqlint32 oline_amount;
    char oline_dist_info[25];
    sqlint64 nulltmstmp = 0;
    sqlint64 currtmstmp;

    IOH_NUM numBytes;
    ioHandle hnd1, hnd2;
    char Buffer[1024];

    oline_dist_info[24] = '\0';

    timestamp1 = current_time();

    rc1 = GenericOpen(&hnd1, outtype1, outname1);
    if (rc1 != 0) { goto ool_done; }
    rc2 = GenericOpen(&hnd2, outtype2, outname2);
    if (rc2 != 0) { goto ool_done; }

    currtmstmp = time(NULL);

    for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
    {
        if (!quiet_mode) {
            fprintf(stdout, "ORDERS & ORDER_LINE for Warehouse #%d\n",
                ware_num);
            fflush(stdout);
        }
        for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE;
            dist_num++)
        {
            if (!quiet_mode) {
                fprintf(stdout, "District #%d\t", dist_num);
                fflush(stdout);
            }
        }

        seed_1_3000();

        for (ord_num = 1; ord_num <= CUSTOMERS_PER_DISTRICT;
            ord_num++)
        {
            if (ord_num < 2101)
                ord_r_carrier_id = rand_integer( 1, 10 );
            else
                ord_r_carrier_id = 0;

            cust_num = random_1_3000();
            ord_r_ol_cnt =
            rand_integer(MIN_OL_PER_ORDER, MAX_OL_PER_ORDER);

```

```

numBytes = sprintf(Buffer, fmtOrdr,
    cust_num,
    currtmstmp,
    ord_r_carrier_id,
    ord_r_ol_cnt,
    1,
    ord_num,
    ware_num,
    dist_num);

rc1 = GenericWrite(&hnd1, Buffer, numBytes);
if (rc1 != 0) { goto ool_done; }

for (oline_ol_num = 1; oline_ol_num <= ord_r_ol_cnt; oline_ol_num++)
{
    oline_item_num = rand_integer(1, ITEMS);
    create_random_a_string(oline_dist_info, 24, 24);

    numBytes = sprintf(Buffer, fmtOLine,
        ((ord_num < 2101) ? currtmstmp : nulltmstmp),
        ((ord_num < 2101) ? 0 : rand_integer(1,999999)),
        oline_item_num,
        ware_num,
        5,
        oline_dist_info,
        ord_num,
        dist_num,
        ware_num,
        oline_ol_num);

    rc2 = GenericWrite(&hnd2, Buffer, numBytes);
    if (rc2 != 0) { goto ool_done; }

    } /* for order_line */
} /* for order */
} /* for dist */
} /* for ware */

rc1 = GenericClose(&hnd2);
rc2 = GenericClose(&hnd1);

ool_done:

timestamp2 = current_time();
elapsed = timestamp2 - timestamp1;
if (rc1 == 0 && rc2 == 0) {
    fprintf(stdout, "\nORDERS & ORDER_LINE table(s) generated in %8.2f
seconds.\n\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nORDERS & ORDER_LINE table(s) FAILED at (W %d D
%d O %d OL %d) after %8.2f seconds.\n\n", ware_num, dist_num, ord_num,
oline_ol_num, elapsed);
    fflush(stderr);
}
}

// This routine will initialize the printf format strings and replace the
// delimiter with the one provided. The pipe symbol is the default.
void InitFormatStrings(char delim)
{
    char *p;

    // Check if Using Default Delimiter
    if (delim == '|') return;

    // Replace Delimiters

```

```

while (p = strchr(fmtWare, '|')) { *p = delim; }
while (p = strchr(fmtDist, '|')) { *p = delim; }
while (p = strchr(fmtItem, '|')) { *p = delim; }
while (p = strchr(fmtStock, '|')) { *p = delim; }
while (p = strchr(fmtCust, '|')) { *p = delim; }
while (p = strchr(fmtHist, '|')) { *p = delim; }
while (p = strchr(fmtOrdr, '|')) { *p = delim; }
while (p = strchr(fmtOLine, '|')) { *p = delim; }
while (p = strchr(fmtNewOrd, '|')) { *p = delim; }
}

```

```

void ScalingReport(void)
{
    /* Print Scaling Values */
    fprintf(stdout, "Scaling Values in Use\n");
    fprintf(stdout, "-----\n");
    fprintf(stdout, "Warehouses: %d\n", WAREHOUSES);
    fprintf(stdout, "Districts/Warehouse: %d\n",
DISTRICTS_PER_WAREHOUSE);
    fprintf(stdout, "Customers/District: %d\n",
CUSTOMERS_PER_DISTRICT);
    fprintf(stdout, "Items: %d\n", ITEMS);
    fprintf(stdout, "Stock/Warehouse: %d\n", STOCK_PER_WAREHOUSE);
    fprintf(stdout, "Min Order Lines/Order: %d\n", MIN_OL_PER_ORDER);
    fprintf(stdout, "Max Order Lines/Order: %d\n", MAX_OL_PER_ORDER);
    fprintf(stdout, "New Orders/District: %d\n",
NU_ORDERS_PER_DISTRICT);
    fprintf(stdout, "-----\n");
}

```

dbgen\makefile

```

#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#####
#####
# Makefile - Build gendata tool
#
!include $(TPCC_ROOT)/Makefile.config
#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####
#####
INCLUDES = -I$(TPCC_SQLLIB)$(SLASH)include
-I$(TPCC_ROOT)$(SLASH)include

CFLAGS = $(INCLUDES) $(CFLAGS_OS) -DLINT_ARGS
-DSQLA_NOLINES \
-D$(DB2EDITION) -D$(DB2VERSION)
$(CFLAGS_DEBUG)

```

```

LD_FLAGS = $(LD_FLAGS_EXEC) $(LD_FLAGS_LIB)

#
#####
#####
# File Collections
#
#####
#####

OBJS =          tpcrnd$(OBJEXT) \
              $(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT)
OBJ_EEE =      $(TPCC_ROOT)/Src.Common/tpcc1wh$(OBJEXT)

EXEC =          gendata$(BINEXT)

#
#####
#####
# End-User Targets
#
#####
#####

all:            $(EXEC)

clean:          - $(ERASE) *$(OBJEXT) $(EXEC)

#
#####
#####
# Build Rules
#
#####
#####

.SUFFIXES:
.SUFFIXES:     $(OBJEXT).c

$(EXEC):
              $(LD_EXEC) $(LD_FLAGS) $(OBJS) *$(OBJEXT)
              $(LD_FLAGS_OUT)$@

#
#####
#####
# Dependencies
#
#####
#####

# Link Dependencies
gendata$(BINEXT): $(OBJS) gendata$(OBJEXT)

# Build Dependencies
# Source
gendata$(OBJEXT): gendata.c

# Headers
gendata.c: $(TPCC_ROOT)/include/tpcrnd.h $(TPCC_ROOT)/include/lval.h

dbgen\tpccrnd.c

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

```

```

** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****/

/*
 * tpcrnd.c - Random generation functions for TPC-C
 */

#include <windows.h>
#include <stdio.h>
#include <string.h>
#include "db2tpcc.h"
#include "tpccmisc.h"
#include "lval.h"

static char tbl_cust[CUSTOMERS_PER_DISTRICT];

static char alnum[] =
"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";

static char *last_name_parts[] =
{
  "BAR",
  "OUGHT",
  "ABLE",
  "PRI",
  "PRES",
  "ESE",
  "ANTI",
  "CALLY",
  "ATION",
  "EING"
};

/*
*****/
*****/
* seed_1_3000
*
*
*****/
*****/
*/

void seed_1_3000( void )
{
  int i;

  for (i = 0; i < CUSTOMERS_PER_DISTRICT; i++) {
    tbl_cust[i] = 0;
  }
}

/*
*****/
*****/
* random_1_3000

```

```

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

int random_1_3000( void )
{
    static int i;
    static int x;

    x = rand_integer(0, CUSTOMERS_PER_DISTRICT - 1);

    for (i = 0; i < CUSTOMERS_PER_DISTRICT; i++)
    {
        if (tbl_cust[x] == 0)
        {
            tbl_cust[x] = 1;
            return(x+1);
        } else {
            x++;
        }
        if (x == CUSTOMERS_PER_DISTRICT)
            x=0;
    }

    printf("\nfatal error in random_1_3000 \n");
    abort();
}

/*
*****
*****
* initialize_random
*****
*****
*/

void initialize_random(void)
{
    int t = current_time();

    srand(t);
    srandom(t);
}

/*
*****
*****
* create_random_a_string
*
* create a random alphanumeric string, of random length between lo and
* hi and place them in designated buffer. Routine returns the actual
* length.
*
* parameters
* -----
* lo end of acceptable length range
* hi end of acceptable length range
*
* output
* -----
* actual length
* random alphanumeric string
*

```

```

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

int create_random_a_string( char *out_buffer, int length_lo, int length_hi )
{
    int i, actual_length ;

    actual_length = rand_integer( length_lo, length_hi ) ;

    for (i = 0; i < actual_length; i++)
    {
        out_buffer[i] = alnum[rand_integer( 0, 61 )] ;
    }
    out_buffer[actual_length] = '\0' ;

    return (actual_length);
}

/*
*****
*****
* create_random_n_string
*
* create a random numeric string, of random length between lo and
* hi and place them in designated buffer. Routine returns the actual
* length.
*
* parameters
* -----
* lo end of acceptable length range
* hi end of acceptable length range
*
* output
* -----
* actual length
* random numeric string
*
*****
*****
*/

int create_random_n_string( char *out_buffer, int length_lo, int length_hi )
{
    int i, actual_length ;

    actual_length = rand_integer( length_lo, length_hi ) ;

    for (i = 0; i < actual_length; i++)
    {
        out_buffer[i] = (char)rand_integer( 48,57 ) ;
    }
    out_buffer[actual_length] = '\0' ;

    return (actual_length);
}

/*
*****
*****
* NUrnd_val
*
* create a non-uniform random numeric value of type integer, of random
* value between lo and hi. Number is NOT placed in BUFFER, and IS
* simply RETURNED.
*
* Routine RETURNS the VALUE.

```

```

*
* parameters
* -----
* lo end of acceptable value range
* hi end of acceptable value range
*
* output
* -----
* random integer value RETURNED
*
*****
*****
*/

int NUrand_val ( int A, int x, int y, int C )
{
    return((((rand_integer(0,A)|rand_integer(x,y))+C)%(y-x+1))+x);
}

/*
*****
*****
* rand_integer
*
* create a uniform random numeric value of type integer, of random
* value between lo and hi. Number is NOT placed in BUFFER, and IS
* simply RETURNED.
*
* Routine RETURNS the VALUE.
*
* parameters
* -----
* lo end of acceptable value range
* hi end of acceptable value range
*
* output
* -----
* random integer value RETURNED
*
*****
*****
*/

int rand_integer ( int val_lo, int val_hi )
{
    return((random()%(val_hi-val_lo+1))+val_lo);
}

/*
*****
*****
* create_a_string_with_original
*
* create a random alphanumeric string, of random length between lo and
* hi and place them in designated buffer. Routine returns the actual
* length.
*
* the word "ORIGINAL" is placed at a random location in the buffer at
* random, for a given percent of the records.
*
* percent_to_set must be an integer value from 0 to 100.
* if 0, no records will be set. If 100, all records will be set.
*
* CANNOT USE ON STRINGS OF LENGTH LESS THAN 8 ! LOWER
* LIMIT MUST BE > 8 !
*
* parameters
* -----

```

```

* lo end of acceptable length range
* hi end of acceptable length range
* percentage of records to set to ORIGINAL
*
* output
* -----
* actual length
* random alphanumeric string with the word "ORIGINAL" is placed at a
* random location
*
*****
*****
*/

int create_a_string_with_original( char *out_buffer, int length_lo,
                                int length_hi, int percent_to_set )
{
    int actual_length, start_pos ;

    actual_length = create_random_a_string( out_buffer, length_lo, length_hi ) ;

    if ( rand_integer( 1, 100 ) <= percent_to_set )
    {
        start_pos = rand_integer( 0, actual_length-8 ) ;
        strncpy(out_buffer+start_pos,"ORIGINAL",8) ;
    }

    return (actual_length);
}

/*
*****
*****
* create_random_last_name
*
* parameters:
* out_buffer - target buffer for the generated last name
*
* description:
* create_random_last_name generates a random number from 0 to 999
* inclusive. a random name is generated by associating a random string
* with each digit of the generated number. the three strings are
* concatenated to generate the name
*
*****
*****
*/

int create_random_last_name(char *out_buffer, int cust_num)
{
    int random_num;

    if (cust_num == 0)
        random_num = NUrand_val( A_C_LAST, 0, 999, C_C_LAST_LOAD );
    else
        random_num = cust_num - 1;

    strcpy(out_buffer, last_name_parts[random_num / 100]);
    random_num %= 100;
    strcat(out_buffer, last_name_parts[random_num / 10]);
    random_num %= 10;
    strcat(out_buffer, last_name_parts[random_num]);

    return(strlen(out_buffer));
}

```

dbgen\include\db2tpcc.h

```

/*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****
*****/

/*
 * db2tpcc.h - Macros and Miscellany
 */

#ifndef __DB2TPCC_H
#define __DB2TPCC_H

#include <sys/types.h>
typedef __int16 int16_t;
typedef __int32 int32_t;
typedef __int64 int64_t;

#include "lval.h"

/*
*****
***** */
/* Transaction Return Codes (s_transtatus) */
/*
*****
***** */

#define INVALID_ITEM 100
#define TRAN_OK 0
#define FATAL_SQLERROR -1

/*
*****
***** */
/* Definition of Unused and Bad Items */
/*
*****
***** */
/* Define unused item ID to be 0. This allows the SUT to determine the
 * number of items in the order as required by 2.4.1.3 and 2.4.2.2 since
 * the assumption that any item with OL_I_ID = 0 is unused will be true.
 * This in turn requires that the value used for an invalid item is
 * equal to ITEMS + 1.
 */
*****
***** */

#define INVALID_ITEM_ID (2 * ITEMS) + 1
#define UNUSED_ITEM_ID 0

#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES

/*****
*****
*****
** NURand Constants
** C_C_LAST_RUN and C_C_LAST_LOAD must adhere to clause 2.1.6.
**
*****/
/*****
*****
** Analysis indicates that a C_LAST delta of 85 is optimal.
**
*****/
#define C_C_LAST_RUN 88
#define C_C_LAST_LOAD 173
#define C_C_ID 319
#define C_OL_I_ID 3849
#define A_C_LAST 255
#define A_C_ID 1023
#define A_OL_I_ID 8191

/*****
*****
** Transaction Type Identifiers
**
*****/

#define CLIENT_SQL 0
#define NEWORD_SQL 1
#define PAYMENT_SQL 2
#define ORDSTAT_SQL 3
#define DELIVERY_SQL 4
#define STOCKLEV_SQL 5

#define SPGENERAL_PAD 3
#define SPGENERAL_ADJUST sizeof(int16_t)

struct in_neword_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
struct in_items_struct {
int32_t s_OL_I_ID;
int32_t s_OL_SUPPLY_W_ID;
int16_t s_OL_QUANTITY;
int16_t pad1[3];
} in_item[15];
int64_t s_O_ENTRY_D_time; /* init by SUT */
int32_t s_C_ID;
int32_t s_W_ID;
int16_t s_D_ID;
int16_t s_O_OL_CNT; /* init by SUT */
int16_t s_all_local;
int16_t duplicate_items;
};

struct out_neword_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
struct items_struct {
int32_t s_I_PRICE;
int32_t s_OL_AMOUNT;
int16_t s_S_QUANTITY;
int16_t pad2;
char s_I_NAME[25];
char s_brand_generic;
} item[15];
int64_t s_O_ENTRY_D_time;
int32_t s_W_TAX;
int32_t s_D_TAX;
int32_t s_C_DISCOUNT;
int32_t s_total_amount;
int32_t s_O_ID;
int16_t s_O_OL_CNT;
int16_t s_transtatus;
int16_t deadlocks;
char s_C_LAST[17];
char s_C_CREDIT[3];
};

```

```

struct in_payment_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_H_DATE_time; /* init by SUT */
    int64_t s_H_AMOUNT;
    int32_t s_W_ID;
    int32_t s_C_W_ID;
    int32_t s_C_ID;
    int16_t s_C_D_ID;
    int16_t s_D_ID;
    char s_C_LAST[17];
};

```

```

struct out_payment_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_H_DATE_time;
    int64_t s_C_SINCE_time;
    int64_t s_C_CREDIT LIM;
    int64_t s_C_BALANCE;
    int32_t s_C_DISCOUNT;
    int32_t s_C_ID;
    int16_t s_transtatus;
    int16_t deadlocks;
    char s_W_STREET_1[21];
    char s_W_STREET_2[21];
    char s_W_CITY[21];
    char s_W_STATE[3];
    char s_W_ZIP[10];
    char s_D_STREET_1[21];
    char s_D_STREET_2[21];
    char s_D_CITY[21];
    char s_D_STATE[3];
    char s_D_ZIP[10];
    char s_C_FIRST[17];
    char s_C_MIDDLE[3];
    char s_C_LAST[17];
    char s_C_STREET_1[21];
    char s_C_STREET_2[21];
    char s_C_CITY[21];
    char s_C_STATE[3];
    char s_C_ZIP[10];
    char s_C_PHONE[17];
    char s_C_CREDIT[3];
    char s_C_DATA[201];
};

```

```

struct in_ordstat_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_C_ID;
    int32_t s_W_ID;
    int16_t s_D_ID;
    int16_t pad1[3];
    char s_C_LAST[17];
};

```

```

struct out_ordstat_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_C_BALANCE;
    int64_t s_O_ENTRY_D_time;
    int32_t s_C_ID;
    int32_t s_O_ID;
    int16_t s_O_CARRIER_ID;
    int16_t s_ol_cnt;
    int16_t pad1[2];
};

```

```

struct oitems_struct {
    int64_t s_OL_DELIVERY_D_time;
    int32_t s_OL_AMOUNT;
    int32_t s_OL_I_ID;
    int32_t s_OL_SUPPLY_W_ID;
    int16_t s_OL_QUANTITY;
    int16_t pad2;
} item[15];
int16_t s_transtatus;
int16_t deadlocks;
char s_C_FIRST[17];
char s_C_MIDDLE[3];
char s_C_LAST[17];
};

```

```

struct in_delivery_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_O_DELIVERY_D_time; /* init by SUT */
    int32_t s_W_ID;
    int16_t s_O_CARRIER_ID;
};

```

```

struct out_delivery_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_O_ID[10];
    int16_t s_transtatus;
    int16_t deadlocks;
};

```

```

struct in_stocklev_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_threshold;
    int32_t s_W_ID;
    int16_t s_D_ID;
};

```

```

struct out_stocklev_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_low_stock;
    int16_t s_transtatus;
    int16_t deadlocks;
};

```

```

/*
*****
***** */
/* Transaction Prototypes */
/*
*****
***** */

```

```

#ifdef __cplusplus
extern "C" {
#endif

extern int neword_sql(struct in_neword_struct*, struct out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct out_stocklev_struct*);

```

```

#ifdef __cplusplus
}
#endif

```



```

/*
***** */
/* DB2 Connect/Disconnect & Thread Context Wrappers */
/*
***** */

#ifdef __cplusplus
extern "C" {
#endif

extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*, char*, char*);
extern int disconnect_from_TM(void);

extern int create_context(void);
extern int destroy_context(void);
extern int get_context(void**);
extern int attach_context(void*);
extern int detach_context(void*);

#ifdef __cplusplus
}
#endif

#endif // __DB2TPCC_H

```

dbgen\include\val.h

```

#ifdef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 12000
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900
#endif // __LVAL_H

```

dbgen\include\platform.h

```

/*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****
******/

/*
* platform.h - Platform Isolation Layer
*/

#ifdef __PLATFORM_H
#define __PLATFORM_H

```

```

/*
***** */
/* Generic Macros */
/*
***** */
#define GEN_ERRCODE GetLastError()

/*
***** */
/* Windows I/O Macros */
/*
***** */
#ifdef INVALID_HANDLE_VALUE
#define INVALID_HANDLE_VALUE ((DWORD)-1)
#endif

#ifdef INVALID_SET_FILE_POINTER
#define INVALID_SET_FILE_POINTER ((DWORD)-1)
#endif

#define IOH_INIT(hnd, type, name) \
    hnd->fd = INVALID_HANDLE_VALUE; \
    hnd->type = type; \
    hnd->name = name;

#define IOH_CREATE(hnd) \
    if (hnd->type == IOH_PIPE) { \
        DWORD timeout = 1000; \
        hnd->fd = CreateNamedPipe(hnd->name, PIPE_ACCESS_OUTBOUND, \
            PIPE_TYPE_BYTE | PIPE_READMODE_BYTE | PIPE_WAIT, \
            1, 0, 0, timeout, NULL); \
        rc = (hnd->fd == INVALID_HANDLE_VALUE) ? -1 : 0; \
    } else { \
        rc = 0; \
    }

#define IOH_OPEN(hnd) \
    if (hnd->type == IOH_PIPE) { \
        rc = (ConnectNamedPipe(hnd->fd, NULL) != 0) ? 0 : -1; \
    } else { \
        hnd->fd = CreateFile(hnd->name, GENERIC_WRITE, \
            FILE_SHARE_WRITE, \
            NULL, OPEN_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL); \
        rc = (hnd->fd == INVALID_HANDLE_VALUE) ? -1 : 0; \
        if (rc == 0 && hnd->type == IOH_FILE_APPEND) { \
            rc = SetFilePointer(hnd->fd, 0, 0, FILE_END); \
            if (rc == INVALID_SET_FILE_POINTER) { \
                rc = (GetLastError() == NO_ERROR) ? 0 : -1; \
            } else { \
                rc = 0; \
            } \
        } \
    }

#define IOH_WRITE(hnd, buff, num, num2) \
    rc = (WriteFile(hnd->fd, buff, num, (LPDWORD)&num2, NULL) != 0) ? 0 : \
    -1;

#define IOH_FLUSH(hnd) \
    if (hnd->type == IOH_PIPE) { \

```

```

    rc = (FlushFileBuffers(hnd->fd) != 0) ? 0 : -1;
} else {
    rc = 0;
}

#define IOH_DELETE(hnd)          rc = 0;

#define IOH_CLOSE(hnd)          \
if (hnd->type == IOH_PIPE) {   \
    rc = (DisconnectNamedPipe(hnd->fd) != 0) ? 0 : -1; \
    IOH_ERRMSG(hnd, "disconnecting"); \
    rc = (CloseHandle(hnd->fd) != 0) ? 0 : -1; \
}

typedef DWORD IOH_NUM;
typedef HANDLE IOH_HND;

/*****
*****/
/* Windows Semaphore Macros */
/*****
*****/
#define SEM_HANDLE HANDLE

#define SEM_INIT(hnd, x, name) \
hnd = CreateSemaphore(NULL, x, 1, NULL); \
if (hnd == NULL) \
    API_ERROR(__LINE__, "CreateSemaphore", (rc=GEN_ERRCODE));

#define SEM_WAIT(hnd) \
if ((rc=WaitForSingleObject(hnd, INFINITE)) == WAIT_FAILED) \
    API_ERROR(__LINE__, "WaitForSingleObject", (rc=GEN_ERRCODE));

#define SEM_FREE(hnd) \
ReleaseSemaphore(hnd, 1, NULL)

#define SEM_DESTROY(hnd) \
if ((rc=CloseHandle(hnd)) == 0) \
    API_ERROR(__LINE__, "CloseHandle", (rc=GEN_ERRCODE));

/*
*****/
/* Common I/O Macros and Definitions */
/*
*****/
#define IOH_FILE 1
#define IOH_PIPE 2
#define IOH_FILE_APPEND 3

#define IOH_ERRMSG(hnd, msg) \
if (rc != 0) { \
    fprintf(stderr, "Error %d %s fd %d (%d, %s)\n", GEN_ERRCODE, msg, \
        hnd->fd, hnd->type, hnd->name); \
    return rc; \
}

struct _ioh {
    IOH_HND fd;
    int type;
    char *name;
};

typedef struct _ioh ioHandle;

/*
*****/
/* Generic I/O Routine Prototypes */
/*
*****/
int GenericOpen(ioHandle *hnd, int type, char *name);
int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes);
int GenericClose(ioHandle *hnd);

/*
*****/
/* Generic I/O Routines */
/*
*****/
int GenericOpen(ioHandle *hnd, int type, char *name)
{
    int rc = 0;

    IOH_INIT(hnd, type, name)

    IOH_CREATE(hnd)
    IOH_ERRMSG(hnd, "creating")

    IOH_OPEN(hnd)
    IOH_ERRMSG(hnd, "opening")

    return rc;
}

int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes)
{
    int rc = 0;
    int numBytesWritten = -1;

    IOH_WRITE(hnd, Buffer, numBytes, numBytesWritten)
    IOH_ERRMSG(hnd, "writing")
    if (numBytes != numBytesWritten) {
        fprintf(stderr, "Truncated data writing to fd %d (%d, %s)\n", hnd->fd,
            hnd->type, hnd->name);
        rc = -1;
    }
    return rc;
}

int GenericClose(ioHandle *hnd)
{
    int rc = 0;

    IOH_FLUSH(hnd)
    IOH_ERRMSG(hnd, "flushing")

    IOH_CLOSE(hnd)
    IOH_ERRMSG(hnd, "closing")

    IOH_DELETE(hnd)
    IOH_ERRMSG(hnd, "deleting")

    return rc;
}

#endif // __PLATFORM_H

```

dbgen\include\tpccrnd.h

```
/*
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****
*****/

/*
* tpccrnd.h - Random generation functions for TPC-C
*/

#ifdef __TPCCRND_H
#define __TPCCRND_H

void initialize_random(void);
int rand_integer( int val_lo, int val_hi );
int NUrand_val( int A, int val_lo, int val_hi, int C );

void seed_1_3000( void );
int random_1_3000( void );

int create_random_a_string( char *out_buffer,
                           int length_lo,
                           int length_hi );
int create_random_n_string( char *out_buffer,
                           int length_lo,
                           int length_hi );
int create_a_string_with_original( char *out_buffer,
                                  int length_lo,
                                  int length_hi,
                                  int percent_to_set );
int create_random_last_name(char *out_buffer, int cust_num);

#endif // __TPCCRND_H
```

dbgen\makefile.config

```
#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#####
#####

#
# Makefile.config - NT/Win2000 Makefile Configuration
```

```
#
# Make Configuration (MSVC)
MAKE=nmake.exe

# Compiler Configuration (MSVC).
# CFLAGS_DEBUG may be set to "-Zi -Od", "-DDEBUGIT" "-Zi -Od
-DDEBUGIT" or left blank
CC=cl.exe
CFLAGS_OS=-DSQLWINT -MT -GS- -DWIN64 -J -Zp8
-DREG_KIT_METHOD
CFLAGS_OUT=/Fo
CFLAGS_DEBUG=

# Linker Configuration (MSVC)
LD_EXEC=link.exe
LD_STORP=link.exe
LDFLAGS_EXEC=
LDFLAGS_SHLIB=/DLL
LDFLAGS_STORP=$(LDFLAGS_SHLIB) /DEF:rpctpcc.def
LDFLAGS_LIB=/LIBPATH:$(TPCC_SQLLIB)\lib
/LIBPATH:"C:\MsSDKx64\Lib\AMD64" db2api.lib WinMM.lib
LDFLAGS_OUT=/OUT:

# Library Configuration
AR=lib.exe
ARFLAGS=
ARFLAGS_LIB=
ARFLAGS_OUT=/OUT:

# OS Commands
ERASE=del /F
ERASEDIR=rmdir /S
MOVE=MOVE
COPY=COPY

# OS File Extensions & Path Separator
OBJEXT=.obj
LIBEXT=.lib
SHLIBEXT=.dll
BINEXT=.exe
SLASH=\\
CMDSEP=&
```

dbgen\Src.Common\makefile

```
#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#####
#####

#
# Makefile - Makefile for Src.Common
#

!include $(TPCC_ROOT)/Makefile.config
```

```

#
#####
#####
# Preprocessor, Compiler and LInker Flags
#
#####
#####

BND_OPTS =      GRANT PUBLIC \
                 MESSAGES $*.bnd.msg
PRP_OPTS =      BINDFILE \
                 OPTLEVEL 1 \
                 ISOLATION RR \
                 MESSAGES $*.prep.msg \
                 LEVEL $(TPCC_VERSION) \
                 NOLINEMACRO

INCLUDES =      -I$(TPCC_SQLLIB)$(SLASH)include
                -I$(TPCC_ROOT)$(SLASH)include

CFLAGS =        $(CFLAGS_OS) $(CFLAGS_DEBUG) $(INCLUDES) \
                -DSQLA_NOLINES -D$(DB2EDITION)
                -D$(DB2VERSION) \
                -D$(TPCC_SPTYPE)

UTIL_OBJ =      tpcmisc$(OBJEXT) tpcdbg$(OBJEXT)
                tpcctx$(OBJEXT)

#
#####
#####
# User Targets
#
#####
#####

all:            connect $(UTIL_OBJ) disconnect

clean:
               - $(ERASE) *$(OBJEXT) *.bnd *.msg tpcctx.c

#
#####
#####
# Helper Targets
#
#####
#####

connect:
               - db2 connect to $(TPCC_DBNAME)

disconnect:
               - db2 connect reset
               - db2 terminate

rebind:
               db2 bind tpcctx.bnd $(BND_OPTS)

#
#####
#####
# Build Rules
#
#####
#####

.SUFFIXES:

```

```

.SUFFIXES: $(OBJEXT) .c .sqc

.sqc.c:
        @echo "Prepping $*.sqc"
        -db2 prep $*.sqc $(PRP_OPTS)
        @echo "Binding $*.bnd"
        db2 bind $*.bnd $(BND_OPTS)

#
#####
#####
# Dependencies
#
#####
#####

# Source
tpcdbg$(OBJEXT): tpcdbg.c
tpcctx$(OBJEXT): tpcctx.c
tpcmisc$(OBJEXT): tpcmisc.c

# Headers
tpcdbg.c: $(TPCC_ROOT)/include/db2tpcc.h

dbgen\Src.Common\tpcmisc.c

/*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****/

/*
 * tpcmisc.c - Miscellaneous routines
 */

#include <windows.h>

#define RAND_A 16807
#define RAND_M 2147483647
#define RAND_M1 2147483646
#define RAND_MD 2147483647.0
#define RAND_Q 127773
#define RAND_R 2836

static int seed = 1;
static int seedflag = 0;

void srandom(int);
int random(void);
double current_time_ms(void);
double current_time(void);

void srandom (int initial_seed)
{
    seed = initial_seed;
    if ((seed < 1) || (seed > RAND_M1)) seed = 1;
}

```

```

int random (void)
{
    int lo;
    int hi;
    int test;

    hi = seed / RAND_Q;
    lo = seed % RAND_Q;
    test = RAND_A * lo - RAND_R * hi;
    if (test > 0) seed = test;
    else seed = test + RAND_M;

    return (seed);
}

/* Current time in SECONDS, precision SECONDS */
double current_time(void)
{
    /* truncate fractional seconds -> seconds */
    return (double)((int)(current_time_ms()));
}

/* Current time in SECONDS, precision MILLISECONDS */
double current_time_ms(void)
{
    /* GetCurrentTime() returns ms */
    /* convert to fractional seconds */
    return (GetCurrentTime() / 1000);
}

```

dbgen\tpccenv.bat

```

@REM
*****
*****
@REM Licensed Materials - Property of IBM
@REM
@REM Governed under the terms of the International
@REM License Agreement for Non-Warranted Sample Code.
@REM
@REM (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
@REM All Rights Reserved.
@REM
@REM US Government Users Restricted Rights - Use, duplication or
@REM disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
@REM
*****
*****
@REM
@REM tpccenv.bat - Windows Environment Setup
@REM

@REM The Kit Version
set TPCC_VERSION=CK041012

@REM The DB2 Instance Name (for DB2)
set DB2INSTANCE=DB2

@REM The OS being used (i.e. "UNIX", "WINDOWS")
set PLATFORM=WINDOWS

@REM The type of make command and slash used by the OS
@REM (i.e. UNIX - "/", WINDOWS - "\")
@REM These are referenced all over the kit.
set SLASH=\
set MAKE=nmake

```

```

set TPCC_SPTYPE=SPGENERAL

set DB2VERSION=v8

@REM The schema name is typically the SQL authorization ID (or username).
@REM This is required for runstats and EEE.
set TPCC_SCHEMA=%USERNAME%

@REM DB2 EE/EEE Configuration
set DB2EDITION=EE
set DB2NODE=0
set DB2NODES=1

@REM TPCC General Configuration
set HOME=C
set TPCC_DBNAME=TPCC
set TPCC_ROOT=c:\tpckit\tpc-c.ibm
set TPCC_SQLLIB=c:\SQLLIB
set TPCC_RUNDATA=c:\tpckit\tpccdata

@REM TPCC Debug Configuration
set TPCC_DEBUGDIR=c:\temp

@REM Specifies where stored procedures should be placed and if they should
@REM be fenced.
set TPCC_SPDIR=%TPCC_SQLLIB%\function
set TPCC_FENCED=NO

```

Appendix C: Tunable Parameters

IBM DB2 UDB

Database Manager Configuration

Database Manager Configuration

Node type = Database Server with local and remote clients

Database manager configuration release level = 0x0a00

Maximum total of files open (MAXTOTFILOP) = 16000

CPU speed (millisec/instruction) (CPUSPEED) = 3.306409e-007

Max number of concurrently active databases (NUMMDB) = 1

Data Links support (DATALINKS) = NO

Federated Database System Support (FEDERATED) = NO

Transaction processor monitor name (TP_MON_NAME) =

Default charge-back account (DFT_ACCOUNT_STR) =

Java Development Kit installation path (JDK_PATH) =
C:\SQLLIB\java\jdk

Diagnostic error capture level (DIAGLEVEL) = 1

Notify Level (NOTIFYLEVEL) = 1

Diagnostic data directory path (DIAGPATH) =

Default database monitor switches

Buffer pool (DFT_MON_BUFPOOL) = OFF

Lock (DFT_MON_LOCK) = OFF

Sort (DFT_MON_SORT) = OFF

Statement (DFT_MON_STMT) = OFF

Table (DFT_MON_TABLE) = OFF

Timestamp (DFT_MON_TIMESTAMP) = OFF

Unit of work (DFT_MON_UOW) = OFF

Monitor health of instance and databases (HEALTH_MON) = OFF

SYSADM group name (SYSADM_GROUP) =

SYSCTRL group name (SYSCTRL_GROUP) =

SYSMAINT group name (SYSMAINT_GROUP) =

SYSMON group name (SYSMON_GROUP) =

Client Userid-Password Plugin (CLNT_PW_PLUGIN) =

Client Kerberos Plugin (CLNT_KRB_PLUGIN) = IBMkrb5

Group Plugin (GROUP_PLUGIN) =

GSS Plugin for Local Authorization (LOCAL_GSSPLUGIN) =

Server Plugin Mode (SRV_PLUGIN_MODE) = UNFENCED

Server List of GSS Plugins (SRVCON_GSSPLUGIN_LIST) =

Server Userid-Password Plugin (SRVCON_PW_PLUGIN) =

Server Connection Authentication (SRVCON_AUTH) =
NOT_SPECIFIED

Database manager authentication (AUTHENTICATION) = CLIENT

Cataloging allowed without authority (CATALOG_NOAUTH) = NO

Trust all clients (TRUST_ALLCLNTS) = YES

Trusted client authentication (TRUST_CLNTAUTH) = CLIENT

Bypass federated authentication (FED_NOAUTH) = NO

Default database path (DFTDBPATH) = C:

Database monitor heap size (4KB) (MON_HEAP_SZ) = 4096

Java Virtual Machine heap size (4KB) (JAVA_HEAP_SZ) = 1024

Audit buffer size (4KB) (AUDIT_BUF_SZ) = 0

Size of instance shared memory (4KB) (INSTANCE_MEMORY) =
AUTOMATIC

Backup buffer default size (4KB) (BACKBUFSZ) = 1024

Restore buffer default size (4KB) (RESTBUFSZ) = 1024

Agent stack size (AGENT_STACK_SZ) = 16

Minimum committed private memory (4KB) (MIN_PRIV_MEM) = 32

Private memory threshold (4KB) (PRIV_MEM_THRESH) = 20000

Sort heap threshold (4KB) (SHEAPTHRES) = 10000

Directory cache support (DIR_CACHE) = YES

Application support layer heap size (4KB) (ASLHEAPSZ) = 15

Max requester I/O block size (bytes) (RQRIOBLK) = 4096

DOS requester I/O block size (bytes) (DOS_RQRIOBLK) = 4096

Query heap size (4KB) (QUERY_HEAP_SZ) = 1000

Workload impact by throttled utilities (UTIL_IMPACT_LIM) = 10

Priority of agents (AGENTPRI) = SYSTEM

Max number of existing agents (MAXAGENTS) = 375

Agent pool size (NUM_POOLAGENTS) = 300

Initial number of agents in pool (NUM_INITAGENTS) = 0

Max number of coordinating agents (MAX_COORDAGENTS) =
MAXAGENTS

Max no. of concurrent coordinating agents (MAXCAGENTS) =
MAX_COORDAGENTS

Max number of client connections (MAX_CONNECTIONS) =
MAX_COORDAGENTS

Keep fenced process (KEEPFENCED) = YES

Number of pooled fenced processes (FENCED_POOL) =
MAX_COORDAGENTS

Initial number of fenced processes (NUM_INITFENCED) = 0

Index re-creation time and redo index build (INDEXREC) = RESTART

Transaction manager database name (TM_DATABASE) = 1ST_CONN

Transaction resync interval (sec) (RESYNC_INTERVAL) = 180

SPM name (SPM_NAME) =

SPM log size (SPM_LOG_FILE_SZ) = 256

SPM resync agent limit (SPM_MAX_RESYNC) = 20

SPM log path (SPM_LOG_PATH) =

NetBIOS Workstation name (NNAME) =

TCP/IP Service name (SVCENAME) = db2c_DB2

Discovery mode (DISCOVER) = SEARCH

Discover server instance (DISCOVER_INST) = ENABLE

Maximum query degree of parallelism (MAX_QUERYDEGREE) = ANY

Enable intra-partition parallelism (INTRA_PARALLEL) = NO

No. of int. communication buffers(4KB)(FCM_NUM_BUFFERS) = 1024

Number of FCM request blocks (FCM_NUM_RQB) = AUTOMATIC

Database Configuration

Database Configuration for Database tpcc

Database configuration release level = 0x0a00

Database release level = 0x0a00

Database territory = US

Database code page = 1252

Database code set = IBM-1252

Database country/region code = 1

Database collating sequence = BINARY

Alternate collating sequence (ALT_COLLATE) =

Database page size = 4096

Dynamic SQL Query management (DYN_QUERY_MGMT) = DISABLE

Discovery support for this database (DISCOVER_DB) = ENABLE

Default query optimization class (DFT_QUERYOPT) = 5

Degree of parallelism (DFT_DEGREE) = 1

Continue upon arithmetic exceptions (DFT_SQLMATHWARN) = NO

Default refresh age (DFT_REFRESH_AGE) = 0

Default maintained table types for opt (DFT_MTTB_TYPES) = SYSTEM

Number of frequent values retained (NUM_FREQVALUES) = 10

Number of quantiles retained (NUM_QUANTILES) = 20

Backup pending = NO

Database is consistent = YES

Rollforward pending = NO

Restore pending = NO

Multi-page file allocation enabled = YES

Log retain for recovery status = RECOVERY

User exit for logging status = NO

Data Links Token Expiry Interval (sec) (DL_EXPINT) = 60

Data Links Write Token Init Expiry Intvl(DL_WT_IEXPINT) = 60

Data Links Number of Copies (DL_NUM_COPIES) = 1

Data Links Time after Drop (days) (DL_TIME_DROP) = 1

Data Links Token in Uppercase (DL_UPPER) = NO

Data Links Token Algorithm (DL_TOKEN) = MACO

Database heap (4KB) (DBHEAP) = 8192

Size of database shared memory (4KB) (DATABASE_MEMORY) = AUTOMATIC

Catalog cache size (4KB) (CATALOGCACHE_SZ) = (MAXAPPLS*4)

Log buffer size (4KB) (LOGBUFSZ) = 3000

Utilities heap size (4KB) (UTIL_HEAP_SZ) = 200000

Buffer pool size (pages) (BUFFPAGE) = 250

Extended storage segments size (4KB) (ESTORE_SEG_SZ) = 16000

Number of extended storage segments (NUM_ESTORE_SEGS) = 0

Max storage for lock list (4KB) (LOCKLIST) = 5000

Max size of appl. group mem set (4KB) (APPGROUP_MEM_SZ) = 30000

Percent of mem for appl. group heap (GROUPHEAP_RATIO) = 70

Max appl. control heap size (4KB) (APP_CTL_HEAP_SZ) = 128

Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = (SHEAPTHRES)

Sort list heap (4KB) (SORTHEAP) = 16

SQL statement heap (4KB) (STMTHEAP) = 16384

Default application heap (4KB) (APPLHEAPSZ) = 328

Package cache size (4KB) (PCKCACHESZ) = 1000

Statistics heap size (4KB) (STAT_HEAP_SZ) = 4384

Interval for checking deadlock (ms) (DLCHKTIME) = 3000

Percent. of lock lists per application (MAXLOCKS) = 100

Lock timeout (sec) (LOCKTIMEOUT) = -1

Changed pages threshold (CHNGPGS_THRESH) = 99

Number of asynchronous page cleaners (NUM_IOCLEANERS) = 4

Number of I/O servers (NUM_IOSERVERS) = 1

Index sort flag (INDEXSORT) = YES

Sequential detect flag (SEQDETECT) = NO

Default prefetch size (pages) (DFT_PREFETCH_SZ) = AUTOMATIC

Track modified pages (TRACKMOD) = OFF
 Default number of containers = 1
 Default tablespace extentsize (pages) (DFT_EXTENT_SZ) = 32
 Max number of active applications (MAXAPPLS) = 375
 Average number of active applications (AVG_APPLS) = 1
 Max DB files open per application (MAXFILOP) = 800
 Log file size (4KB) (LOGFILSIZ) = 256000
 Number of primary log files (LOGPRIMARY) = 31
 Number of secondary log files (LOGSECOND) = 0
 Changed path to log files (NEWLOGPATH) =
 Path to log files = \\.\L:
 Overflow log path (OVERFLOWLOGPATH) =
 Mirror log path (MIRRORLOGPATH) =
 First active log file = S0000002.LOG
 Block log on disk full (BLK_LOG_DSK_FUL) = NO
 Percent of max active log space by transaction(MAX_LOG) = 0
 Num. of active log files for 1 active UOW(NUM_LOG_SPAN) = 0
 Group commit count (MINCOMMIT) = 1
 Percent log file reclaimed before soft chkpt (SOFTMAX) = 900
 Log retain for recovery enabled (LOGRETAIN) = RECOVERY
 User exit for logging enabled (USEREXIT) = OFF
 HADR database role = STANDARD
 HADR local host name (HADR_LOCAL_HOST) =
 HADR local service name (HADR_LOCAL_SVC) =
 HADR remote host name (HADR_REMOTE_HOST) =
 HADR remote service name (HADR_REMOTE_SVC) =
 HADR instance name of remote server (HADR_REMOTE_INST) =
 HADR timeout value (HADR_TIMEOUT) = 120
 HADR log write synchronization mode (HADR_SYNCMODE) = NEARSYNC
 First log archive method (LOGARCHMETH1) = LOGRETAIN
 Options for logarchmeth1 (LOGARCHOPT1) =
 Second log archive method (LOGARCHMETH2) = OFF
 Options for logarchmeth2 (LOGARCHOPT2) =
 Failover log archive path (FAILARCHPATH) =
 Number of log archive retries on error (NUMARCHRETRY) = 5
 Log archive retry Delay (secs) (ARCHRETRYDELAY) = 20
 Vendor options (VENDOROPT) =
 Auto restart enabled (AUTORESTART) = ON

Index re-creation time and redo index build (INDEXREC) = SYSTEM (RESTART)

Log pages during index build (LOGINDEXBUILD) = OFF
 Default number of loadrec sessions (DFT_LOADREC_SES) = 1
 Number of database backups to retain (NUM_DB_BACKUPS) = 12
 Recovery history retention (days) (REC_HIS_RETENTN) = 366
 TSM management class (TSM_MGMTCLASS) =
 TSM node name (TSM_NODENAME) =
 TSM owner (TSM_OWNER) =
 TSM password (TSM_PASSWORD) =
 Automatic maintenance (AUTO_MAINT) = OFF
 Automatic database backup (AUTO_DB_BACKUP) = OFF
 Automatic table maintenance (AUTO_TBL_MAINT) = OFF
 Automatic runstats (AUTO_RUNSTATS) = OFF
 Automatic statistics profiling (AUTO_STATS_PROF) = OFF
 Automatic profile updates (AUTO_PROF_UPD) = OFF
 Automatic reorganization (AUTO_REORG) = OFF

DB2set Parameters

[e] DB2PATH=C:\SQLLIB
 [i] DB2_SELUDI_COMM_BUFFER=Y
 [i] DB2_USE_ALTERNATE_PAGE_CLEANING=YES
 [i] DB2_MAX_NON_TABLE_LOCKS=500
 [i] DB2_LGPAGE_BP=ON
 [i] DB2_TRUSTED_BINDIN=ON
 [i] DB2_KEEPTABLELOCK=ON
 [i] DB2_NO_FORK_CHECK=ON
 [i] DB2_APM_PERFORMANCE=ALL
 [i] DB2_ENABLE_BUFPPD=OFF
 [i] DB2_PINNED_BP=ON
 [i] DB2_SELECTIVITY=ON
 [i] DB2ASSUMEUPDATE=ON
 [i] DB2CHECKCLIENTINTERVAL=0
 [i] DB2_HASH_JOIN=OFF
 [i] DB2_COLLECT_TS_REC_INFO=false
 [i] DB2COMM=tcPIP
 [i] DB2CHKPTR=OFF
 [g] DB2_EXTSECURITY=YES

[g] DB2SYSTEM=DB2SERV1
 [g] DB2PATH=C:\SQLLIB
 [g] DB2ADMINSERVER=DB2DAS00

Microsoft Windows Server 2003 Enterprise x64 Edition

Server Configuration Parameters

Server Configuration Parameters

Microsoft Windows Server 2003 Enterprise x64 Edition Configuration

The following services were set as manual on the server:

- Automatic Updates
- Computer Browser
- Cryptographic Services
- DHCP Client
- Distributed Link Tracking Client
- DNS Client
- Help and Support
- IPSEC service
- Print Spooler
- Remote Registry
- System Event Notification
- Task Scheduler
- Windows Time
- Wireless Configuration

Boot.ini

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows Server 2003
Enterprise x64 Edition - maxmem=65536" /noexecute=optout /fastdetect
/maxmem=65536
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows Server 2003
Enterprise x64 Edition" /noexecute=optout /fastdetect
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows Server 2003
Enterprise x64 Edition - DEBUG" /noexecute=optout /fastdetect
/baudrate=115200 /debugport=com1
```

System Information Report

System Information report written at: 03/22/05 08:14:28

System Name: DB2SERV1

[System Summary]

| Item | Value |
|---------------------|--|
| OS Name | Microsoft(R) Windows(R) Server 2003 Enterprise x64 Edition |
| Version | 5.2.3790 Service Pack 1, v.1289 Build 3790 |
| OS Manufacturer | Microsoft Corporation |
| System Name | DB2SERV1 |
| System Manufacturer | IBM |
| System Model | eserver xSeries 366-[88634RZ]- |
| System Type | x64-based PC |
| Processor | EM64T Family 15 Model 4 Stepping 1 GenuineIntel ~3670 Mhz |
| Processor | EM64T Family 15 Model 4 Stepping 1 GenuineIntel ~3670 Mhz |
| Processor | EM64T Family 15 Model 4 Stepping 1 GenuineIntel ~3670 Mhz |
| Processor | EM64T Family 15 Model 4 Stepping 1 GenuineIntel ~3670 Mhz |
| Processor | EM64T Family 15 Model 4 Stepping 1 GenuineIntel ~3670 Mhz |
| Processor | EM64T Family 15 Model 4 Stepping 1 GenuineIntel ~3670 Mhz |
| Processor | EM64T Family 15 Model 4 Stepping 1 GenuineIntel ~3670 Mhz |

```
Processor EM64T Family 15 Model 4 Stepping 1 GenuineIntel ~3670 Mhz
BIOS Version/Date IBM -[ZUE123AUS-1.01]-, 3/6/2005
SMBIOS Version 2.3
Windows Directory C:\WINDOWS
System Directory C:\WINDOWS\system32
Boot Device \Device\HarddiskVolume232
```

```
Locale United States
Hardware Abstraction Layer Version = "5.2.3790.1289
(srv03_sp1_rc1.041202-1618)"
User NameDB2SERV1\Administrator
Time ZoneEastern Standard Time
Total Physical Memory 63,486.91 MB
Available Physical Memory 60.25 GB
Total Virtual Memory62.45 GB
Available Virtual Memory 62.27 GB
Page File Space 2.00 GB
Page File C:\pagefile.sys
```

[Hardware Resources]

[Conflicts/Sharing]

| Resource | Device | Status |
|--------------------------------|---------------------------------|--------|
| I/O Port 0x00000000-0x00001FFF | PCI bus | |
| I/O Port 0x00000000-0x00001FFF | Direct memory access controller | |

| IRQ | Device | Status |
|--------|--|--------|
| IRQ 20 | NEC PCI to USB Open Host Controller | |
| IRQ 20 | NEC PCI to USB Open Host Controller | |
| IRQ 20 | Standard Enhanced PCI to USB Host Controller | |

| Memory Address | Device | Status |
|-----------------------|--|--------|
| 0xF0900000-0xF09FFFFF | PCI bus | |
| 0xF0900000-0xF09FFFFF | Radeon 7000 / RADEON VE Family (Microsoft Corporation) | |

| I/O Port | Device | Status |
|-----------------------|------------------------------|--------|
| 0x00002200-0x000023FF | PCI bus | |
| 0x00002200-0x000023FF | QLogic Fibre Channel Adapter | |

| I/O Port | Device | Status |
|-----------------------|------------------------------|--------|
| 0x00002000-0x000021FF | PCI bus | |
| 0x00002000-0x000021FF | QLogic Fibre Channel Adapter | |

| Memory Address | Device | Status |
|-----------------------|--------------|--------|
| 0xF0E00000-0xF0EFFFFF | PCI bus | |
| 0xF0E00000-0xF0EFFFFF | PCI standard | |
| PCI-to-PCI bridge | | |

| Memory Address | Device | Status |
|-----------------------|--|--------|
| 0xF0A00000-0xF0AFFFFF | PCI bus | |
| 0xF0A00000-0xF0AFFFFF | Broadcom NetXtreme Gigabit Ethernet #2 | |

| I/O Port | Device | Status |
|-----------------------|------------------------------|--------|
| 0x00002400-0x000025FF | PCI bus | |
| 0x00002400-0x000025FF | QLogic Fibre Channel Adapter | |

| Memory Address | Device | Status |
|------------------------|--|--------|
| 0xE0000000-0xE7FFFFFFF | PCI bus | |
| 0xE0000000-0xE7FFFFFFF | Radeon 7000 / RADEON VE Family (Microsoft Corporation) | |

| Memory Address | Device | Status |
|------------------|--|--------|
| 0xA0000-0xBFFFFF | PCI bus | |
| 0xA0000-0xBFFFFF | Radeon 7000 / RADEON VE Family (Microsoft Corporation) | |

[DMA]

| Resource | Device | Status |
|-----------|---------------------------------|--------|
| Channel 4 | Direct memory access controller | OK |

[Forced Hardware]

Device PNP Device ID

[I/O]

| Resource | Device | Status |
|-------------------------|--|--------|
| 0x00000000-0x00001FFF | PCI bus | OK |
| 0x00000000-0x00001FFF | Direct memory access controller | OK |
| 0x00001800-0x000018FF | Radeon 7000 / RADEON VE Family | OK |
| (Microsoft Corporation) | | |
| 0x000003B0-0x000003BB | Radeon 7000 / RADEON VE Family | OK |
| (Microsoft Corporation) | | |
| 0x000003C0-0x000003DF | Radeon 7000 / RADEON VE Family | OK |
| (Microsoft Corporation) | | |
| 0x00000700-0x0000070F | Standard Dual Channel PCI IDE Controller | 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 |
| 0x00000060-0x00000060 | Standard 101/102-Key or Microsoft Natural | OK |
| PS/2 Keyboard | | |
| 0x00000064-0x00000064 | Standard 101/102-Key or Microsoft Natural | OK |
| PS/2 Keyboard | | |
| 0x000003F8-0x000003FF | Communications Port (COM1) | OK |
| 0x000002F8-0x000002FF | Communications Port (COM2) | OK |
| 0x00000020-0x00000021 | Advanced programmable interrupt controller | OK |
| 0x000000A0-0x000000A1 | Advanced programmable interrupt controller | OK |
| 0x00000080-0x0000008F | Direct memory access controller | OK |
| 0x000000C0-0x000000DF | Direct memory access controller | OK |
| 0x00000040-0x00000043 | System timer | OK |
| 0x00000070-0x00000073 | System CMOS/real time clock | OK |
| 0x00000061-0x00000061 | System speaker | OK |
| 0x000000F0-0x000000FF | Numeric data processor | OK |
| 0x0000002E-0x0000002F | Motherboard resources | OK |
| 0x0000004E-0x0000004F | Motherboard resources | OK |
| 0x00000052-0x00000053 | Motherboard resources | OK |
| 0x00000092-0x00000092 | Motherboard resources | OK |
| 0x00000094-0x0000009F | Motherboard resources | OK |
| 0x000000A8-0x000000A9 | Motherboard resources | OK |
| 0x00000400-0x0000047F | Motherboard resources | OK |
| 0x00000480-0x000004FF | Motherboard resources | OK |
| 0x00000500-0x0000055F | Motherboard resources | OK |
| 0x00000600-0x00000600 | Motherboard resources | OK |
| 0x00000800-0x00000803 | Motherboard resources | OK |
| 0x00000C00-0x00000CDF | Motherboard resources | OK |
| 0x00000F50-0x00000F5F | Motherboard resources | OK |
| 0x00002000-0x000021FF | PCI bus | OK |
| 0x00002000-0x000021FF | QLogic Fibre Channel Adapter | OK |
| 0x00002200-0x000023FF | PCI bus | OK |
| 0x00002200-0x000023FF | QLogic Fibre Channel Adapter | OK |
| 0x00003000-0x00003FFF | PCI bus | OK |
| 0x00002400-0x000025FF | PCI bus | OK |
| 0x00002400-0x000025FF | QLogic Fibre Channel Adapter | OK |
| 0x00004000-0x00004FFF | PCI bus | OK |

[IRQs]

| Resource | Device | Status |
|----------|--|--------|
| IRQ 9 | Microsoft ACPI-Compliant System | OK |
| IRQ 16 | Radeon 7000 / RADEON VE Family (Microsoft Corporation) | OK |
| IRQ 20 | NEC PCI to USB Open Host Controller | OK |
| IRQ 20 | NEC PCI to USB Open Host Controller | OK |
| IRQ 20 | Standard Enhanced PCI to USB Host Controller | OK |
| IRQ 14 | Primary IDE Channel | 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 3 | Communications Port (COM2) | OK |
| IRQ 0 | System timer | OK |
| IRQ 8 | System CMOS/real time clock | OK |
| IRQ 13 | Numeric data processor | OK |
| IRQ 24 | Broadcom NetXtreme Gigabit Ethernet #2 | OK |
| IRQ 28 | Broadcom NetXtreme Gigabit Ethernet | OK |
| IRQ 18 | QLogic Fibre Channel Adapter | OK |
| IRQ 19 | QLogic Fibre Channel Adapter | OK |
| IRQ 52 | QLogic Fibre Channel Adapter | OK |
| IRQ 54 | IBM ServeRAID 6M Controller | OK |

[Memory]

| Resource | Device | Status |
|-------------------------|--|--------|
| 0xA0000-0xBFFFF | PCI bus | OK |
| 0xA0000-0xBFFFF | Radeon 7000 / RADEON VE Family (Microsoft Corporation) | OK |
| 0xE0000000-0xE7FFFFFF | PCI bus | OK |
| 0xE0000000-0xE7FFFFFF | Radeon 7000 / RADEON VE Family (Microsoft Corporation) | OK |
| 0xF0900000-0xF09FFFFF | PCI bus | OK |
| 0xF0900000-0xF09FFFFF | Radeon 7000 / RADEON VE Family (Microsoft Corporation) | OK |
| 0xF0910000-0xF0910FFF | NEC PCI to USB Open Host Controller | OK |
| 0xF0911000-0xF0911FFF | NEC PCI to USB Open Host Controller | OK |
| 0xF0912000-0xF09120FF | Standard Enhanced PCI to USB Host Controller | OK |
| 0x0400-0x04FF | System board | OK |
| 0x10000-0x7FFFFFFF | Memory Module | OK |
| 0xFA00000-0xFAFFFFFF | PCI bus | OK |
| 0xF0A00000-0xF0AFFFFFFF | Broadcom NetXtreme Gigabit Ethernet #2 | OK |
| 0xF0A10000-0xF0A1FFFF | Broadcom NetXtreme Gigabit Ethernet | OK |
| 0xF0B00000-0xF0BFFFFFFF | PCI bus | OK |
| 0xF0B20000-0xF0B20FFF | QLogic Fibre Channel Adapter | OK |
| 0xF0C00000-0xF0CFFFFFFF | PCI bus | OK |
| 0xF0C20000-0xF0C20FFF | QLogic Fibre Channel Adapter | OK |
| 0xE8000000-0xE83FFFFFFF | PCI bus | OK |
| 0xF0000000-0xF03FFFFFFF | PCI bus | OK |
| 0xF0D00000-0xF0DFFFFFFF | PCI bus | OK |
| 0xF0D20000-0xF0D20FFF | QLogic Fibre Channel Adapter | OK |
| 0xE8400000-0xE87FFFFFFF | PCI bus | OK |
| 0xF0400000-0xF07FFFFFFF | PCI bus | OK |
| 0xF0E00000-0xF0EFFFFFFF | PCI bus | OK |
| 0xF0E00000-0xF0EFFFFFFF | PCI standard PCI-to-PCI bridge | OK |
| 0xF0E80000-0xF0E80FFF | IBM ServeRAID 6M Controller | OK |

[Components]

[Multimedia]

[Audio Codecs]

| CODEC | Manufacturer | Description | Status | File |
|----------------------------------|----------------------------------|---------------|-----------------------|------|
| Version | Size | Creation Date | | |
| c:\windows\system32\imaadp32.acm | | | Microsoft Corporation | |
| OK | C:\WINDOWS\system32\IMAADP32.ACM | 5.2.3790.1289 | | |

```
(srv03_sp1_rc1.041202-1618) 23.50 KB (24,064 bytes) 12/3/2004
7:00 AM
c:\windows\system32\msg711.acm Microsoft Corporation
OK C:\WINDOWS\system32\MSG711.ACM 5.2.3790.1289
(srv03_sp1_rc1.041202-1618) 13.50 KB (13,824 bytes) 12/3/2004
7:00 AM
c:\windows\system32\msgsm32.acm Microsoft Corporation
OK C:\WINDOWS\system32\MSGSM32.ACM 5.2.3790.1289
(srv03_sp1_rc1.041202-1618) 34.00 KB (34,816 bytes) 12/3/2004
7:00 AM
c:\windows\system32\tssoft32.acm DSP GROUP, INC.
OK C:\WINDOWS\system32\TSSOFT32.ACM 1.01 13.00 KB
(13,312 bytes) 12/3/2004 7:00 AM
c:\windows\system32\msadp32.acm Microsoft Corporation
OK C:\WINDOWS\system32\MSADP32.ACM 5.2.3790.1289
(srv03_sp1_rc1.041202-1618) 23.00 KB (23,552 bytes) 12/3/2004
7:00 AM
```

[Video Codecs]

| CODEC | Manufacturer | Description | Status | File |
|----------------------------------|----------------------------------|---------------|--------|------|
| Version | Size | Creation Date | | |
| c:\windows\system32\msrle32.dll | Microsoft Corporation | | | |
| OK | C:\WINDOWS\system32\MSRLE32.DLL | 5.2.3790.1289 | | |
| (srv03_sp1_rc1.041202-1618) | 15.50 KB (15,872 bytes) | 12/3/2004 | | |
| 7:00 AM | | | | |
| c:\windows\system32\tsbyuv.dll | Microsoft Corporation | | | |
| OK | C:\WINDOWS\system32\TSBYUV.DLL | 5.2.3790.1289 | | |
| (srv03_sp1_rc1.041202-1618) | 12.50 KB (12,800 bytes) | 12/2/2004 | | |
| 2:09 PM | | | | |
| c:\windows\system32\iyuv_32.dll | Microsoft Corporation | | | |
| OK | C:\WINDOWS\system32\IYUV_32.DLL | 5.2.3790.1289 | | |
| (srv03_sp1_rc1.041202-1618) | 52.50 KB (53,760 bytes) | 12/2/2004 | | |
| 1:52 PM | | | | |
| c:\windows\system32\msvidc32.dll | Microsoft Corporation | | | |
| OK | C:\WINDOWS\system32\MSVIDC32.DLL | 5.2.3790.1289 | | |
| (srv03_sp1_rc1.041202-1618) | 42.00 KB (43,008 bytes) | 12/3/2004 | | |
| 7:00 AM | | | | |
| c:\windows\system32\msyuv.dll | Microsoft Corporation | | | |
| OK | C:\WINDOWS\system32\MSYUV.DLL | 5.2.3790.1289 | | |
| (srv03_sp1_rc1.041202-1618) | 21.50 KB (22,016 bytes) | 12/2/2004 | | |
| 1:53 PM | | | | |

[CD-ROM]

| Item | Value |
|---|--|
| Drive | D: |
| Description | CD-ROM Drive |
| Media Loaded | No |
| Media Type | CD-ROM |
| Name | MATSHITA DVD-ROM SR-8178 |
| Manufacturer | (Standard CD-ROM drives) |
| Status | OK |
| Transfer Rate | Not Available |
| SCSI Target ID | 0 |
| PNP Device ID | |
| IDE\CDROM\MATSHITA_DVD-ROM_SR-8178 | PJ22 |
| \5&A8D2D22&0&0.0.0 | |
| Driver | c:\windows\system32\drivers\cdrom.sys (5.2.3790.1289 |
| (srv03_sp1_rc1.041202-1618), 75.50 KB (77,312 bytes), 12/3/2004 | 7:00 AM) |

[Sound Device]

| Item | Value |
|------|-------|
| Item | Value |

[Display]

| Item | Value |
|------|-------|
| Item | Value |

```
Name Radeon 7000 / RADEON VE Family (Microsoft Corporation)
PNP Device ID
PCI\VEN_1002&DEV_5159&SUBSYS_02C81014&REV_003&267A616A&
0&08
Adapter Type RADEON 7000 (0x5159), ATI Technologies Inc.
compatible
Adapter Description Radeon 7000 / RADEON VE Family (Microsoft
Corporation)
Adapter RAM 16.00 MB (16,777,216 bytes)
Installed Drivers ati2dvag.dll
Driver Version 6.14.10.6490
INF File atiixpag.inf (ati2mtag_RV100 section)
Color Planes 1
Color Table Entries 4294967296
Resolution 1024 x 768 x 60 hertz
Bits/Pixel 32
Memory Address 0xE0000000-0xE7FFFFFFF
I/O Port 0x00001800-0x000018FF
Memory Address 0xF0900000-0xF09FFFFFFF
IRQ Channel IRQ 16
I/O Port 0x000003B0-0x000003BB
I/O Port 0x000003C0-0x000003DF
Memory Address 0xA0000-0xBFFFFF
Driver c:\windows\system32\drivers\ati2mtag.sys (6.14.10.6490, 1.20 MB
(1,260,032 bytes), 2/3/2005 6:07 AM)
```

[Infrared]

| Item | Value |
|------|-------|
| 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&13245C1&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\i8042prt.sys (5.2.3790.1289 |
| (srv03_sp1_rc1.041202-1618), 91.00 KB (93,184 bytes), 12/3/2004 | 7:00 AM) |

[Pointing Device]

| Item | Value |
|---|---|
| Hardware Type | PS/2 Compatible Mouse |
| Number of Buttons | 5 |
| Status | OK |
| PNP Device ID | ACPI\PNP0F13\4&13245C1&0 |
| Power Management Supported | No |
| Double Click Threshold | 6 |
| Handedness | Right Handed Operation |
| IRQ Channel | IRQ 12 |
| Driver | c:\windows\system32\drivers\i8042prt.sys (5.2.3790.1289 |
| (srv03_sp1_rc1.041202-1618), 91.00 KB (93,184 bytes), 12/3/2004 | 7:00 AM) |

[Modem]

| Item | Value |
|------|-------|
| Item | Value |

[Network]

[Adapter]

Item Value
Name [00000001] Broadcom NetXtreme Gigabit Ethernet
Adapter Type Ethernet 802.3
Product Type Broadcom NetXtreme Gigabit Ethernet
Installed Yes
PNP Device ID PCI\VEN_14E4&DEV_1648&SUBSYS_02E71014&REV_10\3&13C0B0C5&0&09
Last Reset 3/22/2005 8:01 AM
Index 1
Service Name b57nd
IP Address 192.168.122.200
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:0D:60:98:00:57
Memory Address 0xF0A10000-0xF0A1FFFF
IRQ Channel IRQ 28
Driver c:\windows\system32\drivers\b57amd64.sys (7.80.0.0 built by: WinDDK, 254.50 KB (260,608 bytes), 2/3/2005 6:08 AM)

Name [00000002] Broadcom NetXtreme Gigabit Ethernet
Adapter Type Ethernet 802.3
Product Type Broadcom NetXtreme Gigabit Ethernet
Installed Yes
PNP Device ID PCI\VEN_14E4&DEV_1648&SUBSYS_02E71014&REV_10\3&13C0B0C5&0&08
Last Reset 3/22/2005 8:01 AM
Index 2
Service Name b57nd
IP Address 192.168.50.201
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:0D:60:98:00:56
Memory Address 0xF0A00000-0xF0AFFFFF
IRQ Channel IRQ 24
Driver c:\windows\system32\drivers\b57amd64.sys (7.80.0.0 built by: WinDDK, 254.50 KB (260,608 bytes), 2/3/2005 6:08 AM)

Name [00000003] RAS Async Adapter
Adapter Type Not Available
Product Type RAS Async Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 3/22/2005 8:01 AM
Index 3
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 [00000004] WAN Miniport (L2TP)
Adapter Type Not Available
Product Type WAN Miniport (L2TP)
Installed Yes
PNP Device ID ROOT\MS_L2TPMINIPORT\0000
Last Reset 3/22/2005 8:01 AM
Index 4
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000005] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPTP)
Installed Yes
PNP Device ID ROOT\MS_PPTPMINIPORT\0000
Last Reset 3/22/2005 8:01 AM
Index 5
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30

Name [00000006] WAN Miniport (PPPOE)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPPOE)
Installed Yes
PNP Device ID ROOT\MS_PPPOEMINIPORT\0000
Last Reset 3/22/2005 8:01 AM
Index 6
Service Name RasPppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30

Name [00000007] Direct Parallel
Adapter Type Not Available
Product Type Direct Parallel
Installed Yes
PNP Device ID ROOT\MS_PTMINIPORT\0000
Last Reset 3/22/2005 8:01 AM
Index 7
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000008] WAN Miniport (IP)
 Adapter Type Not Available
 Product Type WAN Miniport (IP)
 Installed Yes
 PNP Device ID ROOT\MS_NDISWANIP\0000
 Last Reset 3/22/2005 8:01 AM
 Index 8
 Service Name NdisWan
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available

[Protocol]

Item Value
 Name MSADF 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 MSADF Tcpip [UDP/IP]
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 16 bytes
 Maximum Message Size 63.93 KB (65,467 bytes)
 Message Oriented Yes
 Minimum Address Size 16 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting Yes
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting Yes

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 23.50 KB (24,064 bytes)
 Version 5.2.3790.1289 (srv03_sp1_rc1.041202-1618)

[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 19
 XOffXMit Threshold 512
 XOn Character 17
 XOnXMit Threshold 2048
 XOnXOff InFlow Control 0
 XOnXOff OutFlow Control 0
 I/O Port 0x000003F8-0x000003FF
 IRQ Channel IRQ 4
 Driver c:\windows\system32\drivers\serial.sys (5.2.3790.1289
 (srv03_sp1_rc1.041202-1618), 118.50 KB (121,344 bytes), 12/3/2004 7:00
 AM)

Name Communications Port (COM2)
 Status OK
 PNP Device ID ACPI\PNP0501\2
 Maximum Input Buffer Size 0
 Maximum Output Buffer Size No
 Settable Baud Rate Yes
 Settable Data Bits Yes
 Settable Flow Control Yes
 Settable Parity Yes
 Settable Parity Check Yes
 Settable Stop Bits Yes
 Settable RLSD Yes
 Supports RLSD Yes
 Supports 16 Bit Mode No
 Supports Special Characters No
 Baud Rate 9600
 Bits/Byte 8
 Stop Bits 1
 Parity None
 Busy No
 Abort Read/Write on Error No
 Binary Mode Enabled Yes
 Continue XMit on XOff No
 CTS Outflow Control No
 Discard NULL Bytes No
 DSR Outflow Control 0
 DSR Sensitivity 0
 DTR Flow Control Type Enable
 EOF Character 0
 Error Replace Character 0
 Error Replacement Enabled No
 Event Character 0
 Parity Check Enabled No
 RTS Flow Control Type Enable
 XOff Character 19
 XOffXMit Threshold 512
 XOn Character 17
 XOnXMit Threshold 2048
 XOnXOff InFlow Control 0
 XOnXOff OutFlow Control 0
 I/O Port 0x000002F8-0x000002FF
 IRQ Channel IRQ 3
 Driver c:\windows\system32\drivers\serial.sys (5.2.3790.1289
 (srv03_sp1_rc1.041202-1618), 118.50 KB (121,344 bytes), 12/3/2004 7:00
 AM)

[Parallel]

Item Value

[Storage]

[Drives]

| Item | Value |
|----------------------|---------------------------------|
| Drive C: | Local Fixed Disk |
| Description | Local Fixed Disk |
| Compressed | No |
| File System | NTFS |
| Size | 33.90 GB (36,398,149,632 bytes) |
| Free Space | 26.25 GB (28,185,587,712 bytes) |
| Volume Name | |
| Volume Serial Number | 68B790E7 |
| Drive D: | CD-ROM Disc |
| Description | CD-ROM Disc |
| Drive L: | Local Fixed Disk |
| 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 Y: | Network Connection |
| Description | Network Connection |
| Provider Name | \\192.168.50.224\c\$ |
| Drive Z: | Network Connection |
| Description | Network Connection |
| Provider Name | \\fsserv\ddrive |

[Disks]

| Item | Value |
|---------------------------|-------------------------------------|
| Description | Disk drive |
| Manufacturer | (Standard disk drives) |
| Model | IBM 1742-900 SCSI Disk Device |
| Bytes/Sector | 512 |
| Media Loaded | Yes |
| Media Type | Fixed hard disk |
| Partitions | 19 |
| SCSI Bus | 0 |
| SCSI Logical Unit | 0 |
| SCSI Port | 3 |
| SCSI Target ID | 0 |
| Sectors/Track | 63 |
| Size | 935.23 GB (1,004,199,759,360 bytes) |
| Total Cylinders | 122,087 |
| Total Sectors | 1,961,327,655 |
| Total Tracks | 31,132,185 |
| Tracks/Cylinder | 255 |
| Partition | Disk #4, Partition #0 |
| Partition Size | 935.23 GB (1,004,191,534,080 bytes) |
| Partition Starting Offset | 8,225,280 bytes |
| Description | Disk drive |
| Manufacturer | (Standard disk drives) |
| Model | IBM 1742-900 SCSI Disk Device |
| Bytes/Sector | 512 |
| Media Loaded | Yes |
| Media Type | Fixed hard disk |
| Partitions | 19 |
| SCSI Bus | 0 |
| SCSI Logical Unit | 1 |

SCSI Port 3
 SCSI Target ID 0
 Sectors/Track 63
 Size 935.23 GB (1,004,199,759,360 bytes)
 Total Cylinders 122,087
 Total Sectors 1,961,327,655
 Total Tracks 31,132,185
 Tracks/Cylinder 255
 Partition Disk #5, Partition #0
 Partition Size 935.23 GB (1,004,191,534,080 bytes)
 Partition Starting Offset 8,225,280 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1742-900 SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk

Partitions 20
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 3
 SCSI Target ID 1
 Sectors/Track 63
 Size 935.23 GB (1,004,199,759,360 bytes)
 Total Cylinders 122,087
 Total Sectors 1,961,327,655
 Total Tracks 31,132,185
 Tracks/Cylinder 255
 Partition Disk #6, Partition #0
 Partition Size 935.23 GB (1,004,191,534,080 bytes)
 Partition Starting Offset 8,225,280 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1742-900 SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk

Partitions 19
 SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 3
 SCSI Target ID 1
 Sectors/Track 63
 Size 935.23 GB (1,004,199,759,360 bytes)
 Total Cylinders 122,087
 Total Sectors 1,961,327,655
 Total Tracks 31,132,185
 Tracks/Cylinder 255
 Partition Disk #7, Partition #0
 Partition Size 935.23 GB (1,004,191,534,080 bytes)
 Partition Starting Offset 8,225,280 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1742-900 SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk

Partitions 20
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 4
 SCSI Target ID 0
 Sectors/Track 63
 Size 935.23 GB (1,004,199,759,360 bytes)
 Total Cylinders 122,087

Total Sectors 1,961,327,655
 Total Tracks 31,132,185
 Tracks/Cylinder 255
 Partition Disk #8, Partition #0
 Partition Size 935.23 GB (1,004,191,534,080 bytes)
 Partition Starting Offset 8,225,280 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1742-900 SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk

Partitions 19
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 4
 SCSI Target ID 0
 Sectors/Track 63
 Size 935.23 GB (1,004,199,759,360 bytes)
 Total Cylinders 122,087
 Total Sectors 1,961,327,655
 Total Tracks 31,132,185
 Tracks/Cylinder 255
 Partition Disk #9, Partition #0
 Partition Size 935.23 GB (1,004,191,534,080 bytes)
 Partition Starting Offset 8,225,280 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1742-900 SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk

Partitions 20
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 4
 SCSI Target ID 1
 Sectors/Track 63
 Size 935.23 GB (1,004,199,759,360 bytes)
 Total Cylinders 122,087
 Total Sectors 1,961,327,655
 Total Tracks 31,132,185
 Tracks/Cylinder 255
 Partition Disk #10, Partition #0
 Partition Size 935.23 GB (1,004,191,534,080 bytes)
 Partition Starting Offset 8,225,280 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1742-900 SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk

Partitions 19
 SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 4
 SCSI Target ID 1
 Sectors/Track 63
 Size 935.23 GB (1,004,199,759,360 bytes)
 Total Cylinders 122,087
 Total Sectors 1,961,327,655
 Total Tracks 31,132,185
 Tracks/Cylinder 255
 Partition Disk #11, Partition #0
 Partition Size 935.23 GB (1,004,191,534,080 bytes)

Partition Starting Offset 8,225,280 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1742-900 SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 19
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 2
 SCSI Target ID 0
 Sectors/Track 63
 Size 935.23 GB (1,004,199,759,360 bytes)
 Total Cylinders 122,087
 Total Sectors 1,961,327,655
 Total Tracks 31,132,185
 Tracks/Cylinder 255
 Partition Disk #0, Partition #0
 Partition Size 935.23 GB (1,004,191,534,080 bytes)
 Partition Starting Offset 8,225,280 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1742-900 SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 19
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 2
 SCSI Target ID 0
 Sectors/Track 63
 Size 935.23 GB (1,004,199,759,360 bytes)
 Total Cylinders 122,087
 Total Sectors 1,961,327,655
 Total Tracks 31,132,185
 Tracks/Cylinder 255
 Partition Disk #1, Partition #0
 Partition Size 935.23 GB (1,004,191,534,080 bytes)
 Partition Starting Offset 8,225,280 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1742-900 SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 19
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 2
 SCSI Target ID 1
 Sectors/Track 63
 Size 935.23 GB (1,004,199,759,360 bytes)
 Total Cylinders 122,087
 Total Sectors 1,961,327,655
 Total Tracks 31,132,185
 Tracks/Cylinder 255
 Partition Disk #2, Partition #0
 Partition Size 935.23 GB (1,004,191,534,080 bytes)
 Partition Starting Offset 8,225,280 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1742-900 SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 19
 SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 2
 SCSI Target ID 1
 Sectors/Track 63
 Size 935.23 GB (1,004,199,759,360 bytes)
 Total Cylinders 122,087
 Total Sectors 1,961,327,655
 Total Tracks 31,132,185
 Tracks/Cylinder 255
 Partition Disk #3, Partition #0
 Partition Size 935.23 GB (1,004,191,534,080 bytes)
 Partition Starting Offset 8,225,280 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM ServeRAID SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 5
 SCSI Target ID 0
 Sectors/Track 32
 Size 33.90 GB (36,400,267,264 bytes)
 Total Cylinders 17,357
 Total Sectors 71,094,272
 Total Tracks 2,221,696
 Tracks/Cylinder 128
 Partition Disk #12, Partition #0
 Partition Size 33.90 GB (36,398,153,728 bytes)
 Partition Starting Offset 16,384 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM ServeRAID SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 5
 SCSI Target ID 1
 Sectors/Track 32
 Size 237.31 GB (254,808,162,304 bytes)
 Total Cylinders 121,502
 Total Sectors 497,672,192
 Total Tracks 15,552,256
 Tracks/Cylinder 128
 Partition Disk #13, Partition #0
 Partition Size 237.31 GB (254,808,145,920 bytes)
 Partition Starting Offset 16,384 bytes

[SCSI]

Item Value
 Name QLogic Fibre Channel Adapter
 Manufacturer QLogic
 Status OK

PNP Device ID
 PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&1070020&08
 I/O Port 0x00002000-0x000021FF
 Memory Address 0xF0B20000-0xF0B20FFF
 IRQ Channel IRQ 18
 Driver c:\windows\system32\drivers\ql2300.sys (9.0.1.64 Beta 3 (wx64 IP), 715.00 KB (732,160 bytes), 2/9/2005 2:21 PM)

Name QLogic Fibre Channel Adapter
 Manufacturer QLogic
 Status OK
 PNP Device ID
 PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&29E81982&0&08
 I/O Port 0x00002200-0x000023FF
 Memory Address 0xF0C20000-0xF0C20FFF
 IRQ Channel IRQ 19
 Driver c:\windows\system32\drivers\ql2300.sys (9.0.1.64 Beta 3 (wx64 IP), 715.00 KB (732,160 bytes), 2/9/2005 2:21 PM)

Name QLogic Fibre Channel Adapter
 Manufacturer QLogic
 Status OK
 PNP Device ID
 PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&474B838&0&08
 I/O Port 0x00002400-0x000025FF
 Memory Address 0xF0D20000-0xF0D20FFF
 IRQ Channel IRQ 52
 Driver c:\windows\system32\drivers\ql2300.sys (9.0.1.64 Beta 3 (wx64 IP), 715.00 KB (732,160 bytes), 2/9/2005 2:21 PM)

Name IBM ServeRAID 6M Controller
 Manufacturer IBM Corporation
 Status OK
 PNP Device ID
 PCI\VEN_9005&DEV_0250&SUBSYS_02791014&REV_02\4&29C8B970&0&4008
 Memory Address 0xF0E80000-0xF0E80FFF
 IRQ Channel IRQ 54
 Driver c:\windows\system32\drivers\infd960.sys (7.10.52 (NT.041001-1408), 52.00 KB (53,248 bytes), 12/3/2004 7:00 AM)

Name QLogic Optimizing and Multipath Driver
 Manufacturer QLogic
 Status Degraded
 PNP Device ID ROOT\SCSIADAPTER\0000

[IDE]

Item Value
 Name Standard Dual Channel PCI IDE Controller
 Manufacturer (Standard IDE ATA/ATAPI controllers)
 Status OK
 PNP Device ID
 PCI\VEN_1166&DEV_0213&SUBSYS_02121166&REV_A0\3&267A616A&0&79
 I/O Port 0x00000700-0x0000070F
 Driver c:\windows\system32\drivers\pciide.sys (5.2.3790.1289 (srv03_sp1_rc1.041202-1618), 6.00 KB (6,144 bytes), 12/3/2004 7:00 AM)

Name Primary IDE Channel
 Manufacturer (Standard IDE ATA/ATAPI controllers)
 Status OK
 PNP Device ID PCI\IDE\IDECHANNEL\4&101988B2&0&0
 I/O Port 0x000001F0-0x000001F7
 I/O Port 0x000003F6-0x000003F6

IRQ Channel IRQ 14
 Driver c:\windows\system32\drivers\atapi.sys (5.2.3790.1289 (srv03_sp1_rc1.041202-1618), 146.00 KB (149,504 bytes), 12/3/2004 7:00 AM)

Name Secondary IDE Channel
 Manufacturer (Standard IDE ATA/ATAPI controllers)
 Status OK
 PNP Device ID PCI\IDE\IDECHANNEL\4&101988B2&0&1
 I/O Port 0x00000170-0x00000177
 I/O Port 0x00000376-0x00000376
 Driver c:\windows\system32\drivers\atapi.sys (5.2.3790.1289 (srv03_sp1_rc1.041202-1618), 146.00 KB (149,504 bytes), 12/3/2004 7:00 AM)

[Printing]

| Name | Driver | Port | Name | Server | Name |
|------|--------|------|------|--------|------|
|------|--------|------|------|--------|------|

[Problem Devices]

| Device | PNP Device ID | Error Code |
|---------------|--------------------------|--|
| Not Available | ACPI\IBM37D4\2&DABA3FF&0 | The drivers for this device are not installed. |

[USB]

| Device | PNP Device ID |
|--|--|
| NEC PCI to USB Open Host Controller | PCI\VEN_1033&DEV_0035&SUBSYS_00351033&REV_43\3&267A616A&0&18 |
| NEC PCI to USB Open Host Controller | PCI\VEN_1033&DEV_0035&SUBSYS_00351033&REV_43\3&267A616A&0&19 |
| Standard Enhanced PCI to USB Host Controller | PCI\VEN_1033&DEV_00E0&SUBSYS_00E01033&REV_04\3&267A616A&0&1A |

[Software Environment]

[System Drivers]

| Name | Description | File | Type | Started | Start Mode |
|----------|-----------------------|--|---------|---------|------------|
| State | Status | Error Control | Accept | Pause | Accept |
| abiosdsk | Abiosdsk | Not Available | Kernel | Driver | No |
| Disabled | Stopped | OK | Ignore | No | No |
| acpi | Microsoft ACPI Driver | | Kernel | Driver | Yes |
| Boot | Running | OK | Normal | No | Yes |
| acpiec | ACPIEC | c:\windows\system32\drivers\acpiec.sys | Kernel | | |
| Driver | No | Disabled | Stopped | OK | Normal |
| No | | | | | |
| adpu160m | adpu160m | Not Available | Kernel | Driver | No |
| Disabled | Stopped | OK | Normal | No | No |
| adpu320 | adpu320 | Not Available | Kernel | Driver | No |
| Disabled | Stopped | OK | Normal | No | No |
| afd | AFD | c:\windows\system32\drivers\afd.sys | Kernel | | |
| Driver | Yes | System | Running | OK | Normal |
| Yes | | | | | |
| aic78u2 | aic78u2 | Not Available | Kernel | Driver | No |
| Disabled | Stopped | OK | Normal | No | No |
| aic78xx | aic78xx | Not Available | Kernel | Driver | No |
| Disabled | Stopped | OK | Normal | No | No |
| aliide | Aliide | Not Available | Kernel | Driver | No |
| Disabled | Stopped | OK | Normal | No | No |

| | | | | | | | | | | | | | | | |
|---|--|---|---------------|-----|--------|---------|----|--------|--|--|-------------------------------------|--|----------------|-----------|-----|
| serenum | Serenum Filter Driver | c:\windows\system32\drivers\serenum.sys | Kernel Driver | Yes | Manual | Running | OK | Normal | | wanarp | Remote Access IP ARP Driver | c:\windows\system32\drivers\wanarp.sys | Kernel Driver | Yes | |
| No | Yes | | | | | | | | | Manual | Running | OK | Normal | No | Yes |
| serial | Serial port driver | c:\windows\system32\drivers\serial.sys | Kernel Driver | Yes | System | Running | OK | Ignore | | wdica | WDICA | Not Available | Kernel Driver | No | No |
| No | Yes | | | | | | | | | Manual | Stopped | OK | Ignore | No | No |
| sfloppy | High-Capacity Floppy Disk Drive | | | | | | | | | wlbs | Network Load Balancing | c:\windows\system32\drivers\wlbs.sys | Kernel Driver | No | No |
| c:\windows\system32\drivers\sfloppy.sys | | | Kernel Driver | | | | | No | | Manual | Stopped | OK | Normal | No | No |
| Manual | Stopped | OK | Normal | No | No | | | | | [Signed Drivers] | | | | | |
| simbad | Simbad | | Kernel Driver | | | | | No | | Device Name | Signed | Device Class | Driver Version | | |
| Disabled | Stopped | OK | Normal | No | No | | | | | Driver Date | Manufacturer | INF Name | Driver Name | | |
| srv | Srv | c:\windows\system32\drivers\srvc.sys | File System | | | | | | | Device ID | | | | | |
| Driver | Yes | Manual | Running | OK | Normal | No | | | | Microsoft System Management BIOS Driver | No | SYSTEM | | | |
| Yes | | | | | | | | | | 5.2.3790.1289 | 10/1/2002 (Standard system devices) | machine.inf | | | |
| swenum | Software Bus Driver | c:\windows\system32\drivers\swenum.sys | Kernel Driver | Yes | Manual | Running | OK | Normal | | Not Available | ROOT\SYSTEM\0002 | | | | |
| Kernel Driver | Yes | | | | | | | | | Microcode Update Device | No | SYSTEM | 5.2.3790.1289 | | |
| No | Yes | | | | | | | | | 10/1/2002 (Standard system devices) | machine.inf | Not | | | |
| symc8xx | symc8xx | Not Available | Kernel Driver | | | | | No | | Available | ROOT\SYSTEM\0001 | | | | |
| Disabled | Stopped | OK | Normal | No | No | | | | | Plug and Play Software Device Enumerator | No | SYSTEM | | | |
| symmpi | symmpi | Not Available | Kernel Driver | | | | | No | | 5.2.3790.1289 | 10/1/2002 (Standard system devices) | machine.inf | | | |
| Disabled | Stopped | OK | Normal | No | No | | | | | Not Available | ROOT\SYSTEM\0000 | | | | |
| sym_hi | sym_hi | Not Available | Kernel Driver | | | | | No | | QLogic Optimizing and Multipath Driver | No | SCSIADAPTER | | | |
| Disabled | Stopped | OK | Normal | No | No | | | | | Not Available | Not Available | Not Available | oem4.inf | | |
| sym_u3 | sym_u3 | Not Available | Kernel Driver | | | | | No | | Not Available | ROOT\SCSIADAPTER\0000 | | | | |
| Disabled | Stopped | OK | Normal | No | No | | | | | Terminal Server Mouse Driver | No | SYSTEM | 5.2.3790.1289 | | |
| tcpip | TCP/IP Protocol Driver | | | | | | | Yes | | 10/1/2002 (Standard system devices) | machine.inf | Not | | | |
| c:\windows\system32\drivers\tcpip.sys | | | Kernel Driver | | | | | | | Available | ROOT\RDP_MOU\0000 | | | | |
| System | Running | OK | Normal | No | Yes | | | | | Terminal Server Keyboard Driver | No | SYSTEM | | | |
| tdpipe | TDPIPE | c:\windows\system32\drivers\tdpipe.sys | Kernel | | | | | No | | 5.2.3790.1289 | 10/1/2002 (Standard system devices) | machine.inf | | | |
| Driver | No | Manual | Stopped | OK | Ignore | | | | | Not Available | ROOT\RDP_KBD\0000 | | | | |
| No | | | | | | | | | | Terminal Server Device Redirector | No | SYSTEM | | | |
| tdtcp | TDTCP | c:\windows\system32\drivers\tdtcp.sys | Kernel | | | | | No | | 5.2.3790.1289 | 10/1/2002 (Standard system devices) | machine.inf | | | |
| Driver | Yes | Manual | Running | OK | Ignore | | | | | Not Available | ROOT\RDPDR\0000 | | | | |
| Yes | | | | | | | | | | Direct Parallel | No | NET | 5.2.3790.1289 | 10/1/2002 | |
| termdd | Terminal Device Driver | | | | | | | Yes | | Microsoft netrasa.inf | Not Available | ROOT\MS_PTMINIPORT\0000 | | | |
| c:\windows\system32\drivers\termdd.sys | | | Kernel Driver | | | | | | | WAN Miniport (PPTP) | No | NET | 5.2.3790.1289 | | |
| System | Running | OK | Normal | No | Yes | | | | | 10/1/2002 Microsoft netrasa.inf | Not Available | | | | |
| toside | TosIde | Not Available | Kernel Driver | | | | | No | | ROOT\MS_PPTPMINIPORT\0000 | | | | | |
| Disabled | Stopped | OK | Normal | No | No | | | | | WAN Miniport (PPPOE) | No | NET | 5.2.3790.1289 | | |
| udfs | Udfs | c:\windows\system32\drivers\udfs.sys | File System | | | | | No | | 10/1/2002 Microsoft netrasa.inf | Not Available | | | | |
| Driver | No | Disabled | Stopped | OK | Normal | | | | | ROOT\MS_PPPOEMINIPORT\0000 | | | | | |
| No | | | | | | | | | | WAN Miniport (IP) | No | NET | 5.2.3790.1289 | 10/1/2002 | |
| ultra | ultra | Not Available | Kernel Driver | | | | | No | | Microsoft netrasa.inf | Not Available | ROOT\MS_NDISWANIP\0000 | | | |
| Disabled | Stopped | OK | Normal | No | No | | | | | WAN Miniport (L2TP) | No | NET | 5.2.3790.1289 | | |
| update | Microcode Update Driver | | | | | | | Yes | | 10/1/2002 Microsoft netrasa.inf | Not Available | | | | |
| c:\windows\system32\drivers\update.sys | | | Kernel Driver | | | | | | | ROOT\MS_L2TPMINIPORT\0000 | | | | | |
| Manual | Running | OK | Normal | No | Yes | | | | | Video Codecs | No | MEDIA | 5.2.3790.1289 | 10/1/2002 | |
| usbehci | Microsoft USB 2.0 Enhanced Host Controller Miniport Driver | | | | | | | Yes | | (Standard system devices) | wave.inf | Not Available | | | |
| c:\windows\system32\drivers\usbehci.sys | | | Kernel Driver | | | | | | | ROOT\MEDIA\MS_MMVID | | | | | |
| Manual | Running | OK | Normal | No | Yes | | | | | Legacy Video Capture Devices | No | MEDIA | 5.2.3790.1289 | | |
| usbhub | USB2 Enabled Hub | c:\windows\system32\drivers\usbhub.sys | Kernel Driver | Yes | Manual | Running | OK | Normal | | 10/1/2002 (Standard system devices) | wave.inf | Not Available | | | |
| Kernel Driver | Yes | | | | | | | | | ROOT\MEDIA\MS_MMVCD | | | | | |
| No | Yes | | | | | | | | | Media Control Devices | No | MEDIA | 5.2.3790.1289 | | |
| usbohci | Microsoft USB Open Host Controller Miniport Driver | | | | | | | Yes | | 10/1/2002 (Standard system devices) | wave.inf | Not Available | | | |
| c:\windows\system32\drivers\usbohci.sys | | | Kernel Driver | | | | | | | ROOT\MEDIA\MS_MMMCI | | | | | |
| Manual | Running | OK | Normal | No | Yes | | | | | Legacy Audio Drivers | No | MEDIA | 5.2.3790.1289 | 10/1/2002 | |
| usbstor | USB Mass Storage Driver | | | | | | | No | | (Standard system devices) | wave.inf | Not Available | | | |
| c:\windows\system32\drivers\usbstor.sys | | | Kernel Driver | | | | | | | ROOT\MEDIA\MS_MMDRV | | | | | |
| Manual | Stopped | OK | Normal | No | No | | | | | Audio Codecs | No | MEDIA | 5.2.3790.1289 | 10/1/2002 | |
| vgasave | VGA Display Controller. | | | | | | | Yes | | (Standard system devices) | wave.inf | Not Available | | | |
| c:\windows\system32\drivers\vga.sys | | | Kernel Driver | | | | | | | ROOT\MEDIA\MS_MMACM | | | | | |
| System | Running | OK | Ignore | No | Yes | | | | | Remote Access IP ARP Driver | Not Available | LEGACYDRIVER | | | |
| viaide | ViaIde | Not Available | Kernel Driver | | | | | No | | Not Available | Not Available | Not Available | Not | | |
| Disabled | Stopped | OK | Normal | No | No | | | | | Available | Not Available | ROOT\LEGACY_WANARP\0000 | | | |
| volsnap | Storage volumes | c:\windows\system32\drivers\volsnap.sys | Kernel Driver | Yes | Boot | Running | OK | Normal | | | | | | | |
| Kernel Driver | Yes | | | | | | | | | | | | | | |
| No | Yes | | | | | | | | | | | | | | |

| | | | |
|--------------------------------------|---------------------------|--------------------------|---------------|
| volsnap | Not Available | LEGACYDRIVER | Not Available |
| Not Available | Not Available | Not Available | Not |
| Available | ROOT\LEGACY_VOLSNAPO\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_TDTCP\0000 | | |
| TCP/IP Protocol Driver | Not Available | LEGACYDRIVER | |
| Not Available | Not Available | Not Available | Not |
| Available | Not Available | ROOT\LEGACY_TCPIP\0000 | |
| Security Driver | Not Available | LEGACYDRIVER | Not |
| Available | Not Available | Not Available | Not Available |
| Not Available | ROOT\LEGACY_SECDRV\0000 | | |
| RDPWD | Not Available | LEGACYDRIVER | Not Available |
| Not Available | Not Available | Not Available | Not |
| Available | ROOT\LEGACY_RDPWD\0000 | | |
| RDPCDD | Not Available | LEGACYDRIVER | Not Available |
| Not Available | Not Available | Not Available | Not |
| Available | ROOT\LEGACY_RDPCDD\0000 | | |
| Remote Access Auto Connection Driver | Not Available | | |
| LEGACYDRIVER | Not Available | Not Available | Not |
| Available | Not Available | Not Available | |
| ROOT\LEGACY_RASACD\0000 | | | |
| qldirect | Not Available | LEGACYDRIVER | Not Available |
| Not Available | Not Available | Not Available | Not |
| Available | ROOT\LEGACY_QLDIRECT\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 | | |
| 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_NDISTAPI\0000 | | | |
| NDIS System Driver | Not Available | LEGACYDRIVER | Not |
| Available | Not Available | Not Available | Not Available |
| Not Available | ROOT\LEGACY_NDIS\0000 | | |
| mountmgr | Not Available | LEGACYDRIVER | Not Available |
| Not Available | Not Available | Not Available | Not |
| Available | ROOT\LEGACY_MOUNTMGR\0000 | | |
| ksecdd | Not Available | LEGACYDRIVER | Not Available |
| Not Available | Not Available | Not Available | Not |
| Available | ROOT\LEGACY_KSECDD\0000 | | |
| IPSEC driver | Not Available | LEGACYDRIVER | Not |
| Available | Not Available | Not Available | Not Available |
| Not Available | ROOT\LEGACY_IPSEC\0000 | | |
| IP Network Address Translator | Not Available | LEGACYDRIVER | |
| Not Available | Not Available | Not Available | Not |
| Available | Not Available | ROOT\LEGACY_IPNAT\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 | Not Available |
| Not Available | Not Available | Not Available | Not |
| Available | ROOT\LEGACY_FIPS\0000 | | |

| | | | |
|--|---------------------------|--------------------------|---------------|
| dmload | Not Available | LEGACYDRIVER | Not Available |
| Not Available | Not Available | Not Available | Not |
| Available | ROOT\LEGACY_DMLOAD\0000 | | |
| dmboot | Not Available | LEGACYDRIVER | Not Available |
| Not Available | Not Available | Not Available | Not |
| Available | 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 | |
| CdaD10BA | Not Available | LEGACYDRIVER | Not |
| Available | Not Available | Not Available | Not Available |
| Not Available | ROOT\LEGACY_CDAD10BA\0000 | | |
| CdaC15BA | Not Available | LEGACYDRIVER | Not |
| Available | Not Available | Not Available | Not Available |
| Not Available | ROOT\LEGACY_CDAC15BA\0000 | | |
| Beep | Not Available | LEGACYDRIVER | Not Available |
| Not Available | Not Available | Not Available | Not |
| Available | ROOT\LEGACY_BEEP\0000 | | |
| AFD | Not Available | LEGACYDRIVER | Not Available |
| Not Available | Not Available | Not Available | Not |
| Available | ROOT\LEGACY_AFD\0000 | | |
| Generic volume | No | VOLUME 5.2.3790.1289 | 10/1/2002 |
| Microsoft | volume.inf | Not Available | |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE2459E01OFFSET4000 | | | |
| LENGTH3B53BFC000 | | | |
| Generic volume | No | VOLUME 5.2.3790.1289 | 10/1/2002 |
| Microsoft | volume.inf | Not Available | |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE244FD00OFFSET4000 | | | |
| LENGTH8797FC000 | | | |
| Generic volume | No | VOLUME 5.2.3790.1289 | 10/1/2002 |
| Microsoft | volume.inf | Not Available | |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0597OFFSET5F3 | | | |
| 09E5000LENGTH3EB9200 | | | |
| Generic volume | No | VOLUME 5.2.3790.1289 | 10/1/2002 |
| Microsoft | volume.inf | Not Available | |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0597OFFSET5F2 | | | |
| CB24000LENGTH3EB9200 | | | |
| Generic volume | No | VOLUME 5.2.3790.1289 | 10/1/2002 |
| Microsoft | volume.inf | Not Available | |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0597OFFSET1A | | | |
| D0B0E200LENGTH445C00E000 | | | |
| Generic volume | No | VOLUME 5.2.3790.1289 | 10/1/2002 |
| Microsoft | volume.inf | Not Available | |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0597OFFSET1A | | | |
| B8AB0000LENGTH18056400 | | | |
| Generic volume | No | VOLUME 5.2.3790.1289 | 10/1/2002 |
| Microsoft | volume.inf | Not Available | |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0597OFFSET1A0 | | | |
| 24A1800LENGTHB6606A00 | | | |
| Generic volume | No | VOLUME 5.2.3790.1289 | 10/1/2002 |
| Microsoft | volume.inf | Not Available | |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0597OFFSET19A | | | |
| 5239C00LENGTH5D25FE00 | | | |
| Generic volume | No | VOLUME 5.2.3790.1289 | 10/1/2002 |
| Microsoft | volume.inf | Not Available | |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0597OFFSET195 | | | |
| 248CA00LENGTH52DA5400 | | | |
| Generic volume | No | VOLUME 5.2.3790.1289 | 10/1/2002 |
| Microsoft | volume.inf | Not Available | |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0597OFFSET186 | | | |
| B5E9C00LENGTH6E9B000 | | | |
| Generic volume | No | VOLUME 5.2.3790.1289 | 10/1/2002 |
| Microsoft | volume.inf | Not Available | |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0597OFFSET130 | | | |
| 2DC8C00LENGTH568819200 | | | |
| Generic volume | No | VOLUME 5.2.3790.1289 | 10/1/2002 |
| Microsoft | volume.inf | Not Available | |

| | |
|---|---|
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET1A D597F600LENGTH445C00E000 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSETB8 200LENGTH7D0400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET1A DIABE600LENGTH3EB9200 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET7E0 000LENGTH7D0400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET1A BC971000LENGTH15145800 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSETA8 6ECC9A00LENGTH3EB9200 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET1A0 6362800LENGTHB6606A00 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSETA8 6AE08A00LENGTH3EB9200 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET19A 90FAC00LENGTH5D25FE00 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET5F2 CB24000LENGTH493E2DCC00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET195 634DA00LENGTH52DA5400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET1A D0B0E200LENGTH445C00E000 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET186 F4AAC00LENGTHHE6E9B000 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET1A B8AB0000LENGTH18056400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET130 6C89C00LENGTH568819200 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET1A 024A1800LENGTHB6606A00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET11C 6268000LENGTH140A19E00 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET19 A5239C00LENGTH5D25FE00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSETA43 54DC00LENGTH782D12600 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET195 248CA00LENGTH52DA5400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET8D E6B2C00LENGTH164E93200 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET186 B5E9C00LENGTHHE6E9B000 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET845 F9A00LENGTH85A0B1400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET130 2DC8C00LENGTH568819200 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET807 38A00LENGTH3EB9200 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET11 C23A7000LENGTH140A19E00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET686 DA800LENGTH18056400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSETA3 F68CC00LENGTH782D12600 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET59B 86C00LENGTHEB4BE00 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET8D A7F1C00LENGTH164E93200 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET1F6 8600LENGTH57C16800 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET807 38A00LENGTH85A0B1400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE1409B099OFFSET179 0400LENGTH7D0400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET686 DA800LENGTH18056400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |

| | |
|---|---|
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET59B86C00LENGTHB4BE00 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET80738A00LENGTH85A0B1400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET1F68600LENGTH57C16800 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET686DA800LENGTH18056400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET1790400LENGTH7D0400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET59B86C00LENGTHB4BE00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSETFB8200LENGTH7D0400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET1F68600LENGTH57C16800 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F059AOFFSET7E0000LENGTH7D0400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET1790400LENGTH7D0400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET5F309E5000LENGTH3EB9200 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSETFB8200LENGTH7D0400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET5F2CB24000LENGTH3EB9200 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET7E0000LENGTH7D0400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET1AD0B0E200LENGTH445C00E000 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0594OFFSETA86ECC9A00LENGTH3EB9200 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET1A8B8A0000LENGTH18056400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0594OFFSETA86AE08A00LENGTH3EB9200 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET1A024A1800LENGTHB6606A00 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0594OFFSET5F2CB24000LENGTH493E2DCC00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET19A5239C00LENGTH5D25FE00 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0594OFFSET1AD0B0E200LENGTH445C00E000 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET195248CA00LENGTH52DA5400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0594OFFSET1A8B8A0000LENGTH18056400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET186B5E9C00LENGTH6E9B000 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0594OFFSET1A024A1800LENGTHB6606A00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET1302DC8C00LENGTH568819200 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0594OFFSET19A5239C00LENGTH5D25FE00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET11C23A7000LENGTH140A19E00 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0594OFFSET195248CA00LENGTH52DA5400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSETA3F68CC00LENGTH782D12600 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0594OFFSET186B5E9C00LENGTH6E9B000 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0595OFFSET8DA7F1C00LENGTH164E93200 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0594OFFSET1302DC8C00LENGTH568819200 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |

| | |
|---|---|
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSET19A5239C00LENGTH5D25FE00 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET1A8A0000LENGTH18056400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSET195248CA00LENGTH52DA5400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET1A024A1800LENGTHB6606A00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSET186B5E9C00LENGTHE6E9B000 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET19A5239C00LENGTH5D25FE00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSET1302DC8C00LENGTH568819200 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET195248CA00LENGTH52DA5400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSET11C23A7000LENGTHH140A19E00 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET186B5E9C00LENGTHE6E9B000 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSETA3F68CC00LENGTH782D12600 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET1302DC8C00LENGTH568819200 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSET8DA7F1C00LENGTHH164E93200 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET11C23A7000LENGTHH140A19E00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSET80738A00LENGTH85A0B1400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSETA3F68CC00LENGTH782D12600 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSET686DA800LENGTH18056400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET8DA7F1C00LENGTHH164E93200 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSET59B86C00LENGTHEB4BE00 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET80738A00LENGTH85A0B1400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSET1F68600LENGTH57C16800 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET686DA800LENGTH18056400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSET1790400LENGTH7D0400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET59B86C00LENGTHEB4BE00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSETFB8200LENGTH7D0400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET1F68600LENGTH57C16800 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F0592OFFSET7E0000LENGTH7D0400 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET1790400LENGTH7D0400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET5F309E5000LENGTH3EB9200 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSETFB8200LENGTH7D0400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET5F2CB24000LENGTH3EB9200 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET7E0000LENGTH7D0400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET1AD0B0E200LENGTH445C00E000 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ADOFFSET5F309E5000LENGTH3EB9200 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Generic volume No VOLUME 5.2.3790.1289 10/1/2002 |
| Microsoft volume.infNot Available | Microsoft volume.infNot Available |

| | |
|--|---|
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET5F2CB24000LENGTH3EB9200 | STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET7E0000LENGTH7D0400 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Volume Manager No SYSTEM 5.2.3790.1289 10/1/2002 |
| Microsoft volume.inf Not Available | (Standard system devices) machine.inf Not Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET1A D0B0E200LENGTH445C00E000 | ROOT\FTDISK\0000 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Logical Disk Manager No SYSTEM 5.2.3790.1289 10/1/2002 |
| Microsoft volume.inf Not Available | (Standard system devices) machine.inf Not Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET1A B8AB0000LENGTH18056400 | ROOT\DMIO\0000 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | IBM Dummy Device No SYSTEM 5.2.3790.1289 10/1/2002 |
| Microsoft volume.inf Not Available | IBM scsidev.inf Not Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET1A 024A1800LENGTHB6606A00 | SCSI\BRIDGE&VEN_IBM&PROD_DUMMY_DEVICE&REV_7.10\5&804C5&0&300 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | SCSI Processor Device No SYSTEM 5.2.3790.1289 |
| Microsoft volume.inf Not Available | 10/1/2002 IBM scsidev.inf Not Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET19 A5239C00LENGTH5D25FE00 | SCSI\PROCESSOR&VEN_IBM&PROD_EXP400__S320&REV_D110\5&804C5&0&2F0 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | SCSI Processor Device No SYSTEM 5.2.3790.1289 |
| Microsoft volume.inf Not Available | 10/1/2002 IBM scsidev.inf Not Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET19 5248CA00LENGTH5DA5400 | SCSI\PROCESSOR&VEN_IBM&PROD_EXP400__S320&REV_D110\5&804C5&0&1F0 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 |
| Microsoft volume.inf Not Available | (Standard disk drives) disk.inf Not Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET18 6B5E9C00LENGTH6E9B000 | SCSI\DISK&VEN_IBM&PROD_SERVERAID&REV_6.10\5&804C5&0&010 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 |
| Microsoft volume.inf Not Available | (Standard disk drives) disk.inf Not Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET13 02DC8C00LENGTH568819200 | SCSI\DISK&VEN_IBM&PROD_SERVERAID&REV_6.10\5&804C5&0&000 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | IBM ServeRAID 6M Controller No SCSIADAPTER |
| Microsoft volume.inf Not Available | 5.2.3790.1289 10/1/2002 IBM Corporation pnpscsi.inf Not Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET11 C23A7000LENGTH140A19E00 | PCI\VEN_9005&DEV_0250&SUBSYS_02791014&REV_02\4&29C8B970&0&4008 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | PCI standard PCI-to-PCI bridge No SYSTEM 5.2.3790.1289 |
| Microsoft volume.inf Not Available | 10/1/2002 (Standard system devices) machine.inf Not Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSETA3 F68CC00LENGTH782D12600 | PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\3&20FEA912&0&08 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | PCI standard host CPU bridge No SYSTEM 5.2.3790.1289 |
| Microsoft volume.inf Not Available | 10/1/2002 (Standard system devices) machine.inf Not Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET8D A7F1C00LENGTH164E93200 | PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&20FEA912&0&00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | PCI bus No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available |
| Microsoft volume.inf Not Available | ACPI\PNP0A03\7 |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET80 738A00LENGTH85A0B1400 | PCI standard host CPU bridge No SYSTEM 5.2.3790.1289 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | 10/1/2002 (Standard system devices) machine.inf Not Available |
| Microsoft volume.inf Not Available | Available |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET68 6DA800LENGTH18056400 | PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&E44F86D&0&00 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | PCI bus No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available |
| Microsoft volume.inf Not Available | ACPI\PNP0A03\6 |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET59 B86C00LENGTHEB4BE00 | Qlogic processor device No SYSTEM 5.2.3790.1289 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | 10/1/2002 QLOGIC scsidev.inf Not Available |
| Microsoft volume.inf Not Available | SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_4&1D1C1BB3&0&07F0 |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET1F 68600LENGTH57C16800 | Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | (Standard disk drives) disk.inf Not Available |
| Microsoft volume.inf Not Available | SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&1D1C1BB3&0&013 |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSET17 90400LENGTH7D0400 | Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | (Standard disk drives) disk.inf Not Available |
| Microsoft volume.inf Not Available | SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&1D1C1BB3&0&012 |
| STORAGE\VOLUME\1&30A96598&0&SIGNATURE181F05ACOFFSETFB 8200LENGTH7D0400 | Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 |
| Generic volume No VOLUME 5.2.3790.1289 10/1/2002 | (Standard disk drives) disk.inf Not Available |
| Microsoft volume.inf Not Available | |

| | |
|---|---|
| SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&1D1C1BB3&0&01 | SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3B4E3515&1&001 |
| Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 (Standard disk drives) disk.inf Not Available | Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 (Standard disk drives) disk.inf Not Available |
| SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&1D1C1BB3&0&00 | SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3B4E3515&1&000 |
| QLogic Fibre Channel Adapter No SCSIADAPTER 9.0.1.64 2/18/2005 QLogic oem6.inf Not Available | QLogic Fibre Channel Adapter No SCSIADAPTER 9.0.1.64 2/18/2005 QLogic oem6.inf Not Available |
| PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&474B838&0&08 | PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&1070020&0&08 |
| PCI standard host CPU bridge No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available | PCI standard host CPU bridge No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available |
| PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&474B838&0&00 | PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&1070020&0&00 |
| PCI bus No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available | PCI bus No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available |
| ACPI\PNP0A03\5 | ACPI\PNP0A03\2 |
| PCI standard host CPU bridge No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available | Broadcom NetXtreme Gigabit Ethernet No NET 7.80.0.0 10/1/2002 Broadcom net57amd.inf Not Available |
| Available | PCI\VEN_14E4&DEV_1648&SUBSYS_02E71014&REV_10\3&13C0B0C5&0&09 |
| PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&172E68DD&0&00 | Broadcom NetXtreme Gigabit Ethernet No NET 7.80.0.0 10/1/2002 Broadcom net57amd.inf Not Available |
| PCI bus No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available | PCI\VEN_14E4&DEV_1648&SUBSYS_02E71014&REV_10\3&13C0B0C5&0&08 |
| ACPI\PNP0A03\4 | PCI standard host CPU bridge No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available |
| Qlogic processor device No SYSTEM 5.2.3790.1289 10/1/2002 QLOGIC scsasidev.inf Not Available | Available |
| SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_4&232FAFE&1&07F0 | PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&13C0B0C5&0&00 |
| Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 (Standard disk drives) disk.inf Not Available | PCI bus No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available |
| SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&232FAFE&1&013 | ACPI\PNP0A03\1 |
| Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 (Standard disk drives) disk.inf Not Available | Memory Module No MEMORY 5.2.3790.1289 10/1/2002 Microsoft memory.inf Not Available ACPI\PNP0C80\0 |
| SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&232FAFE&1&012 | Intel Processor No PROCESSOR 5.2.3790.1289 10/1/2002 Intel cpu.inf Not Available |
| Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 (Standard disk drives) disk.inf Not Available | ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_7 |
| SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&232FAFE&1&001 | Intel Processor No PROCESSOR 5.2.3790.1289 10/1/2002 Intel cpu.inf Not Available |
| Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 (Standard disk drives) disk.inf Not Available | ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_6 |
| SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&232FAFE&1&000 | Intel Processor No PROCESSOR 5.2.3790.1289 10/1/2002 Intel cpu.inf Not Available |
| QLogic Fibre Channel Adapter No SCSIADAPTER 9.0.1.64 2/18/2005 QLogic oem6.inf Not Available | ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_5 |
| PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&29E81982&0&08 | Intel Processor No PROCESSOR 5.2.3790.1289 10/1/2002 Intel cpu.inf Not Available |
| PCI standard host CPU bridge No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available | ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_4 |
| Available | Intel Processor No PROCESSOR 5.2.3790.1289 10/1/2002 Intel cpu.inf Not Available |
| PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&29E81982&0&00 | ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_3 |
| PCI bus No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available | Intel Processor No PROCESSOR 5.2.3790.1289 10/1/2002 Intel cpu.inf Not Available |
| ACPI\PNP0A03\3 | ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_2 |
| Qlogic processor device No SYSTEM 5.2.3790.1289 10/1/2002 QLOGIC scsasidev.inf Not Available | Intel Processor No PROCESSOR 5.2.3790.1289 10/1/2002 Intel cpu.inf Not Available |
| SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_4&3B4E3515&1&07F0 | ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_1 |
| Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 (Standard disk drives) disk.inf Not Available | Intel Processor No PROCESSOR 5.2.3790.1289 10/1/2002 Intel cpu.inf Not Available |
| SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3B4E3515&1&013 | ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_0 |
| Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 (Standard disk drives) disk.inf Not Available | ACPI Fixed Feature Button No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available |
| SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3B4E3515&1&012 | Available ACPI\FIXEDBUTTON\2&DABA3FF&0 |
| Disk drive No DISKDRIVE 5.2.3790.1289 10/1/2002 (Standard disk drives) disk.inf Not Available | System board No SYSTEM 5.2.3790.1289 10/1/2002 (Standard system devices) machine.inf Not Available |
| SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3B4E3515&1&011 | ACPI\PNP0C01\1 |

| | | | |
|---|--|---------------|---------------|
| Not Available | Not Available | Not Available | Not Available |
| Available | Not Available | Not Available | Not Available |
| Not Available | ACPI\IBM37D42&DABA3FF&0 | | |
| Motherboard resources | No | SYSTEM | 5.2.3790.1289 |
| 10/1/2002 (Standard system devices) | | machine.inf | Not Available |
| Available | ACPI\PNP0C023 | | |
| Numeric data processor | No | SYSTEM | 5.2.3790.1289 |
| 10/1/2002 (Standard system devices) | | machine.inf | Not Available |
| Available | ACPI\PNP0C04\4&13245C1&0 | | |
| System speaker | No | SYSTEM | 5.2.3790.1289 |
| (Standard system devices) | | machine.inf | Not Available |
| Available | ACPI\PNP0800\4&13245C1&0 | | |
| System CMOS/real time clock | No | SYSTEM | 5.2.3790.1289 |
| 10/1/2002 (Standard system devices) | | machine.inf | Not Available |
| Available | ACPI\PNP0B00\4&13245C1&0 | | |
| System timer | No | SYSTEM | 5.2.3790.1289 |
| (Standard system devices) | | machine.inf | Not Available |
| Available | ACPI\PNP0100\4&13245C1&0 | | |
| Direct memory access controller | No | SYSTEM | 5.2.3790.1289 |
| 10/1/2002 (Standard system devices) | | machine.inf | Not Available |
| Available | ACPI\PNP0200\4&13245C1&0 | | |
| Advanced programmable interrupt controller | No | SYSTEM | 5.2.3790.1289 |
| 10/1/2002 (Standard system devices) | | machine.inf | Not Available |
| Available | ACPI\PNP0003\4&13245C1&0 | | |
| Communications Port | No | PORTS | 5.2.3790.1289 |
| (Standard port types) | | msports.inf | Not Available |
| Available | ACPI\PNP0501\2 | | |
| Communications Port | No | PORTS | 5.2.3790.1289 |
| (Standard port types) | | msports.inf | Not Available |
| Available | ACPI\PNP0501\1 | | |
| PS/2 Compatible Mouse | No | MOUSE | 5.2.3790.1289 |
| 10/1/2002 (Standard system devices) | | msmouse.inf | Not Available |
| Available | ACPI\PNP0F13\4&13245C1&0 | | |
| Standard 101/102-Key or Microsoft Natural PS/2 Keyboard | No | KEYBOARD | 5.2.3790.1289 |
| 10/1/2002 (Standard keyboards) | | keyboard.inf | Not Available |
| Available | ACPI\PNP0303\4&13245C1&0 | | |
| Serverworks Champion CSB6 - SouthBridge 6 LPC | No | SYSTEM | 5.2.3790.1289 |
| 10/1/2002 (Standard system devices) | | machine.inf | Not Available |
| Available | PCI\VEN_1166&DEV_0227&SUBSYS_00000000&REV_00\3&267A616A&0&7B | | |
| Secondary IDE Channel | No | HDC | 5.2.3790.1289 |
| 10/1/2002 (Standard IDE ATA/ATAPI controllers) | | mshdc.inf | Not Available |
| Available | PCI\IDE\IDECHANNEL\4&101988B2&0&1 | | |
| CD-ROM Drive | No | CDROM | 5.2.3790.1289 |
| (Standard CD-ROM drives) | | cdrom.inf | Not Available |
| Available | IDE\CDROM\ATSHITA_DVD-ROM_SR-8178 | | PJ22 |
| 5&A8D2D22&0&0.0.0 | | | |
| Primary IDE Channel | No | HDC | 5.2.3790.1289 |
| 10/1/2002 (Standard IDE ATA/ATAPI controllers) | | mshdc.inf | Not Available |
| Available | PCI\IDE\IDECHANNEL\4&101988B2&0&0 | | |
| Standard Dual Channel PCI IDE Controller | No | HDC | 5.2.3790.1289 |
| 10/1/2002 (Standard IDE ATA/ATAPI controllers) | | mshdc.inf | Not Available |
| Available | PCI\VEN_1166&DEV_0213&SUBSYS_02121166&REV_A0\3&267A616A&0&79 | | |
| ServerWorks Champion CSB6 - SouthBridge 6 | No | SYSTEM | 5.2.3790.1289 |
| 10/1/2002 (Standard system devices) | | machine.inf | Not Available |
| Available | PCI\VEN_1166&DEV_0203&SUBSYS_00000000&REV_A0\3&267A616A&0&78 | | |
| USB Root Hub | No | USB | 5.2.3790.1289 |
| (Standard USB Host Controller) | | usbport.inf | Not Available |
| Available | USB\ROOT_HUB20\4&2B778F81&0 | | |
| Standard Enhanced PCI to USB Host Controller | No | USB | 5.2.3790.1289 |
| 10/1/2002 (Standard USB Host Controller) | | usbport.inf | Not Available |

| | | | |
|--|--|---------------|---------------|
| PCI\VEN_1033&DEV_00E0&SUBSYS_00E01033&REV_04\3&267A616A&0&1A | | | |
| USB Root Hub | No | USB | 5.2.3790.1289 |
| (Standard USB Host Controller) | | usbport.inf | Not Available |
| Available | USB\ROOT_HUB\4&2DDBD7B&0 | | |
| NEC PCI to USB Open Host Controller | No | USB | 5.2.3790.1289 |
| 10/1/2002 (Standard system devices) | | usbport.inf | Not Available |
| Available | PCI\VEN_1033&DEV_0035&SUBSYS_00351033&REV_43\3&267A616A&0&19 | | |
| USB Root Hub | No | USB | 5.2.3790.1289 |
| (Standard USB Host Controller) | | usbport.inf | Not Available |
| Available | USB\ROOT_HUB\4&15976E20&0 | | |
| NEC PCI to USB Open Host Controller | No | USB | 5.2.3790.1289 |
| 10/1/2002 (Standard system devices) | | usbport.inf | Not Available |
| Available | PCI\VEN_1033&DEV_0035&SUBSYS_00351033&REV_43\3&267A616A&0&18 | | |
| Plug and Play Monitor | No | MONITOR | 5.2.3790.1289 |
| (Standard monitor types) | | monitor.inf | Not Available |
| Available | DISPLAY\AVO0000\4&36FA8DD8&0&10000080&00&01 | | |
| Plug and Play Monitor | No | MONITOR | 5.2.3790.1289 |
| (Standard monitor types) | | monitor.inf | Not Available |
| Available | DISPLAY\IBM029A\4&36FA8DD8&0&10000082&00&01 | | |
| Radeon 7000 / RADEON VE Family (Microsoft Corporation) | No | DISPLAY | 6.14.10.6490 |
| 10/19/2004 (Standard system devices) | | atiixpag.inf | Not Available |
| Available | PCI\VEN_1002&DEV_5159&SUBSYS_02C81014&REV_00\3&267A616A&0&08 | | |
| PCI standard host CPU bridge | No | SYSTEM | 5.2.3790.1289 |
| 10/1/2002 (Standard system devices) | | machine.inf | Not Available |
| Available | PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&267A616A&0&00 | | |
| PCI bus | No | SYSTEM | 5.2.3790.1289 |
| (Standard system devices) | | machine.inf | Not Available |
| Available | ACPI\PNP0A03\0 | | |
| Microsoft ACPI-Compliant System | No | SYSTEM | 5.2.3790.1289 |
| 10/1/2002 (Standard system devices) | | acpi.inf | Not Available |
| Available | ACPI_HAL\PNP0C08\0 | | |
| ACPI Multiprocessor x64-based PC | No | COMPUTER | 5.2.3790.1289 |
| (Standard computers) | | hal.inf | Not Available |
| Available | ROOT\ACPI_HAL\0000 | | |
| Not Available | Not Available | Not Available | Not Available |
| Available | Not Available | Not Available | Not Available |
| Not Available | HTREE\ROOT\0 | | |

[Environment Variables]

| Variable | Value | User Name |
|----------------------|---|-----------|
| CLASSPATH | .;C:\SQLLIB\java\db2java.zip;C:\SQLLIB\java\db2jcc.jar;C:\SQLLIB\java\sqlj.zip;C:\SQLLIB\java\db2jcc_license_cisuz.jar;C:\SQLLIB\java\db2jcc_license_cu.jar;C:\SQLLIB\bin;C:\SQLLIB\java\common.jar | <SYSTEM> |
| ClusterLog | C:\WINDOWS\Cluster\cluster.log | <SYSTEM> |
| ComSpec | %SystemRoot%\system32\cmd.exe | <SYSTEM> |
| DB2INSTANCE | DB2 | <SYSTEM> |
| DB2TEMPDIR | C:\SQLLIB\ | <SYSTEM> |
| FP_NO_HOST_CHECK | NO | <SYSTEM> |
| INCLUDE | C:\MsSDKx64\Include;C:\MsSDKx64\Include\cert;C:\MsSDKx64\Include\cert\sys;C:\MsSDKx64\Include\mf;C:\MsSDKx64\Include\atl;C:\VC\include;C:\VC\include\sys;C:\SQLLIB\INCLUDE;C:\SQLLIB\LIB | <SYSTEM> |
| LIB | C:\MsSDKx64\Lib\AMD64;C:\MsSDKx64\Lib\AMD64\mf;C:\SQLLIB\LIB | <SYSTEM> |
| NUMBER_OF_PROCESSORS | 8 | <SYSTEM> |
| OS | Windows_NT | <SYSTEM> |

```

Path
C:\MsSDKx64\Bin\Win64\x86\AMD64;C:\MsSDKx64\Bin;C:\MsSDKx64\Bin\Winnt;C:\Perl\bin;%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\SQLLIB\BIN;C:\SQLLIB\FUNCTION;C:\SQLLIB\SAMPLES\REPL;C:\tpckit\tpc-c.ibm\utils\Windows;c:\tools
<SYSTEM>
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
<SYSTEM>
PROCESSOR_ARCHITECTURE AMD64 <SYSTEM>
PROCESSOR_IDENTIFIER EM64T Family 15 Model 4 Stepping 1, GenuineIntel
<SYSTEM>
PROCESSOR_LEVEL 15 <SYSTEM>
PROCESSOR_REVISION 0401 <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 DB2SERV1\Administrator
TMP %USERPROFILE%\Local Settings\Temp DB2SERV1\Administrator
VCToolkitInstallDir C:\MSVC\DB2SERV1\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 |
|------------|----------------------|------|------------|------------|
| Y: | \\192.168.50.224\c\$ | Disk | Persistent | Connection |
| DB2SERV1\ | Administrator | | | |
| Z: | \\fsserv\ddrive | Disk | Current | Connection |
| DB2SERV1\ | Administrator | | | |

[Running Tasks]

| Name | Path | Process ID | Priority | Min Working Set | Max Working Set |
|-----------------------------|----------------------------------|---------------|-------------------|-----------------|-------------------|
| | | | | Size | File Date |
| system idle process | | Not Available | | 0 | 0 |
| Available | Not Available | Not Available | | Not Available | Not Available |
| Not Available | Not Available | Not Available | | Not Available | Not Available |
| system | Not Available | 4 | | 8 | 0 |
| Not Available | Not Available | Not Available | | Not Available | Not Available |
| Available | | | | | |
| smss.exe | Not Available | 588 | | 11 | 204800 |
| 3/22/2005 8:02 AM | Not Available | Not Available | | Not Available | Not Available |
| Available | | | | | |
| csrss.exe | Not Available | 816 | | 13 | Not Available |
| Not Available | 3/22/2005 8:03 AM | Not Available | | Not Available | Not Available |
| Available | Not Available | | | | |
| winlogon.exe | c:\windows\system32\winlogon.exe | | | 204 | |
| 13 | 204800 | 1413120 | 3/22/2005 8:04 AM | 5.2.3790.1289 | |
| (srv03_sp1_rc1.041202-1618) | 848.00 KB (868,352 bytes) | | | | 12/3/2004 7:00 AM |

| | | | | | |
|-----------------------------|--|---------------------------|-----------------------------|-------------------------|-------------------|
| services.exe | c:\windows\system32\services.exe | 364 | 9 | | |
| 204800 | 1413120 | 3/22/2005 8:04 AM | 5.2.3790.1289 | | |
| (srv03_sp1_rc1.041202-1618) | 217.50 KB (222,720 bytes) | | | | 12/3/2004 7:00 AM |
| lsass.exe | c:\windows\system32\lsass.exe | 384 | 9 | 204800 | |
| 1413120 | 3/22/2005 8:04 AM | 5.2.3790.1289 | (srv03_sp1_rc1.041202-1618) | | |
| 13.50 KB (13,824 bytes) | | | | | 12/3/2004 7:00 AM |
| svchost.exe | c:\windows\system32\svchost.exe | 636 | 8 | | |
| 204800 | 1413120 | 3/22/2005 8:04 AM | 5.2.3790.1289 | | |
| (srv03_sp1_rc1.041202-1618) | 24.50 KB (25,088 bytes) | | | | 12/3/2004 7:00 AM |
| svchost.exe | Not Available | 748 | 8 | Not Available | |
| Available | Not Available | 3/22/2005 8:04 AM | Not Available | | |
| Not Available | Not Available | | | | |
| svchost.exe | Not Available | 808 | 8 | Not Available | |
| Available | Not Available | 3/22/2005 8:04 AM | Not Available | | |
| Not Available | Not Available | | | | |
| svchost.exe | c:\windows\system32\svchost.exe | 848 | 8 | | |
| 204800 | 1413120 | 3/22/2005 8:04 AM | 5.2.3790.1289 | | |
| (srv03_sp1_rc1.041202-1618) | 24.50 KB (25,088 bytes) | | | | 12/3/2004 7:00 AM |
| msdtc.exe | Not Available | 1228 | 8 | Not Available | |
| Not Available | 3/22/2005 8:04 AM | Not Available | Not Available | | |
| Available | Not Available | | | | |
| svchost.exe | c:\windows\system32\svchost.exe | 1392 | 8 | | |
| 204800 | 1413120 | 3/22/2005 8:04 AM | 5.2.3790.1289 | | |
| (srv03_sp1_rc1.041202-1618) | 24.50 KB (25,088 bytes) | | | | 12/3/2004 7:00 AM |
| wrshdnt.exe | c:\wrshdnt\wrshdnt.exe | 1424 | 8 | | |
| 204800 | 1413120 | 3/22/2005 8:04 AM | 2.23.00 | 92.00 KB (94,208 bytes) | |
| 2/22/2005 12:16 PM | | | | | |
| db2sec.exe | c:\sql\bin\db2sec.exe | 1808 | 8 | 204800 | |
| 1413120 | 3/22/2005 8:04 AM | 8.1.9.787 | 8.50 KB (8,704 bytes) | | 2/4/2005 10:11 AM |
| svchost.exe | c:\windows\system32\svchost.exe | 1868 | 8 | | |
| 204800 | 1413120 | 3/22/2005 8:04 AM | 5.2.3790.1289 | | |
| (srv03_sp1_rc1.041202-1618) | 24.50 KB (25,088 bytes) | | | | 12/3/2004 7:00 AM |
| wmiprvse.exe | Not Available | 1852 | 8 | Not Available | |
| Available | Not Available | 3/22/2005 8:05 AM | Not Available | | |
| Not Available | Not Available | | | | |
| explorer.exe | c:\windows\explorer.exe | 1272 | 8 | | |
| 204800 | 1413120 | 3/22/2005 8:11 AM | 6.00.3790.1289 | | |
| (srv03_sp1_rc1.041202-1618) | 1.31 MB (1,373,184 bytes) | | | | 12/3/2004 7:00 AM |
| helpctr.exe | c:\windows\pchealth\helpctr\binaries\helpctr.exe | 572 | 8 | | |
| 204800 | 1413120 | 3/22/2005 8:12 AM | 5.2.3790.1289 | | |
| (srv03_sp1_rc1.041202-1618) | 1.30 MB (1,365,504 bytes) | | | | 2/3/2005 11:52 AM |
| wmiprvse.exe | Not Available | 1724 | 8 | Not Available | |
| Available | Not Available | 3/22/2005 8:12 AM | Not Available | | |
| Not Available | Not Available | | | | |
| helpsvc.exe | c:\windows\pchealth\helpctr\binaries\helpsvc.exe | 968 | 8 | 204800 | |
| 5.2.3790.1289 | (srv03_sp1_rc1.041202-1618) | 1.52 MB (1,590,272 bytes) | | | 2/3/2005 11:52 AM |

[Loaded Modules]

| Name | Version | Size | File Date | Manufacturer | Path |
|----------|---|---------------------------|-------------------|-----------------------|----------------------------------|
| winlogon | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 848.00 KB (868,352 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\winlogon.exe |
| ntdll | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.20 MB (1,259,008 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\ntdll.dll |

| | | | | | |
|----------------------------------|---|-------------|---|---|---------------------------|
| kernel32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.43 MB | shlwapi | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 608.50 KB |
| (1,498,112 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | (623,104 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| c:\windows\system32\kernel32.dll | | | c:\windows\system32\shlwapi.dll | | |
| advapi32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1,008.50 KB | sfc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 6.00 KB |
| (1,032,704 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | (6,144 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| c:\windows\system32\advapi32.dll | | | c:\windows\system32\sfc.dll | | |
| rpert4 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.63 MB | sfc_os | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 183.00 KB |
| (1,707,520 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | (187,392 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| c:\windows\system32\rpert4.dll | | | c:\windows\system32\sfc_os.dll | | |
| crypt32 | 5.131.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.36 MB | wintrust | 5.131.3790.1289 (srv03_sp1_rc1.041202-1618) | 297.00 KB |
| (1,423,872 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | (304,128 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| c:\windows\system32\crypt32.dll | | | c:\windows\system32\wintrust.dll | | |
| msasn1 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 151.50 KB | imagehlp | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 57.50 KB |
| (155,136 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | (58,880 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| c:\windows\system32\msasn1.dll | | | c:\windows\system32\imagehlp.dll | | |
| msvcr7 | 7.0.3790.1289 (srv03_sp1_rc1.041202-1618) | 512.00 KB | ole32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 2.42 MB |
| (524,288 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | (2,539,520 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| c:\windows\system32\msvcr7.dll | | | c:\windows\system32\ole32.dll | | |
| user32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.05 MB | comctl32 | 6.0 (srv03_sp1_rc1.041202-1618) | 1.51 MB (1,583,616 bytes) |
| (1,100,288 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | | 2/3/2005 6:00 AM Microsoft Corporation | |
| c:\windows\system32\user32.dll | | | c:\windows\winsxs\amd64_microsoft.windows.common-controls_6595b64144c | | |
| gdi32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 588.50 KB | cfldf | 6.0.3790.1289_x-ww_a9fe16ed\comctl32.dll | |
| (602,624 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | winscard | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 229.00 KB |
| c:\windows\system32\gdi32.dll | | | (234,496 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| nddeapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 25.00 KB | c:\windows\system32\wincard.dll | | |
| (25,600 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | wtsapi32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 27.50 KB |
| c:\windows\system32\nddeapi.dll | | | (28,160 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| profmap | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 35.00 KB | c:\windows\system32\wtsapi32.dll | | |
| (35,840 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | winmm | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 296.00 KB |
| c:\windows\system32\profmap.dll | | | (303,104 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| netapi32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 590.00 KB | c:\windows\system32\winmm.dll | | |
| (604,160 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | shell32 | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 10.04 MB |
| c:\windows\system32\netapi32.dll | | | (10,531,328 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| userenv | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.02 MB | c:\windows\system32\shell32.dll | | |
| (1,067,520 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | sxs | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.91 MB |
| c:\windows\system32\userenv.dll | | | (2,003,968 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| psapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 28.50 KB | c:\windows\system32\sxs.dll | | |
| (29,184 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | rsaenh | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 241.46 KB |
| c:\windows\system32\psapi.dll | | | (247,256 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| regapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 107.00 KB | c:\windows\system32\rsaenh.dll | | |
| (109,568 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | wldap32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 367.00 KB |
| c:\windows\system32\regapi.dll | | | (375,808 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| secur32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 121.00 KB | c:\windows\system32\wldap32.dll | | |
| (123,904 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | csd.dll | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 149.50 KB |
| c:\windows\system32\secur32.dll | | | (153,088 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| setupapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.46 MB | c:\windows\system32\csd.dll | | |
| (1,533,952 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | dimsntfy | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 27.50 KB |
| c:\windows\system32\setupapi.dll | | | (28,160 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| version | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 28.00 KB | c:\windows\system32\dimsntfy.dll | | |
| (28,672 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | wlnotify | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 147.50 KB |
| c:\windows\system32\version.dll | | | (151,040 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| winsta | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 87.50 KB | c:\windows\system32\wlnotify.dll | | |
| (89,600 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | mpr | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 115.00 KB |
| c:\windows\system32\winsta.dll | | | (117,760 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| ws2_32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 174.50 KB | c:\windows\system32\mpr.dll | | |
| (178,688 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | oleaut32 | 5.2.3790.1289 | 1.07 MB (1,118,720 bytes) |
| c:\windows\system32\ws2_32.dll | | | | 12/3/2004 7:00 AM Microsoft Corporation | |
| ws2help | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 29.00 KB | c:\windows\system32\oleaut32.dll | | |
| (29,696 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | winspool | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 246.50 KB |
| c:\windows\system32\ws2help.dll | | | (252,416 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| msgina | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.22 MB | c:\windows\system32\winspool.drv | | |
| (1,277,440 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | comctl32 | 5.82 (srv03_sp1_rc1.041202-1618) | 935.50 KB (957,952 bytes) |
| c:\windows\system32\msgina.dll | | | | 2/3/2005 6:00 AM Microsoft Corporation | |
| shsvcs | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 193.50 KB | c:\windows\winsxs\amd64_microsoft.windows.common-controls_6595b64144c | | |
| (198,144 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | cfldf | 5.82.3790.1289_x-ww_4a89d168\comctl32.dll | |
| c:\windows\system32\shsvcs.dll | | | uxtheme | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 457.00 KB |
| | | | (467,968 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | |
| | | | c:\windows\system32\uxtheme.dll | | |

| | | | | | | |
|----------|---|-----------|-------------------|-------------------|-----------------------|----------------------------------|
| mprapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 153.50 KB | (157,184 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\mprapi.dll |
| activeds | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 347.50 KB | (355,840 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\activeds.dll |
| adslidpc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 241.00 KB | (246,784 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\adslidpc.dll |
| credui | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 201.50 KB | (206,336 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\credui.dll |
| atl | 3.05.2284 96.00 KB (98,304 bytes) | | | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\atl.dll |
| rtutils | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 65.00 KB | (66,560 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\rtutils.dll |
| samlib | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 67.00 KB | (68,608 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\samlib.dll |
| clbcatq | 2001.12.4720.1289 (srv03_sp1_rc1.041202-1618) | 867.00 KB | (887,808 bytes) | 2/3/2005 11:50 AM | Microsoft Corporation | c:\windows\system32\clbcatq.dll |
| comres | 2001.12.4720.1289 (srv03_sp1_rc1.041202-1618) | 778.50 KB | (797,184 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\comres.dll |
| xpsp2res | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 2.77 MB | (2,899,456 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\xpsp2res.dll |
| cscui | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 441.50 KB | (452,096 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\cscui.dll |
| drprov | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 22.00 KB | (22,528 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\drprov.dll |
| ntlman | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 71.50 KB | (73,216 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\ntlman.dll |
| netui0 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 129.50 KB | (132,608 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\netui0.dll |
| netui1 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 338.00 KB | (346,112 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\netui1.dll |
| davclnt | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 36.50 KB | (37,376 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\davclnt.dll |
| mprui | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 67.50 KB | (69,120 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\mprui.dll |
| netui2 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 545.50 KB | (558,592 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\netui2.dll |
| comdlg32 | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 445.00 KB | (455,680 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\comdlg32.dll |
| netmsg | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 179.00 KB | (183,296 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\netmsg.dll |
| ntmarta | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 222.00 KB | (227,328 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\ntmarta.dll |
| services | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 217.50 KB | (222,720 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\services.exe |
| ncobjapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 79.50 KB | (81,408 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\ncobjapi.dll |
| msvcpx60 | 6.10.30806.0 | 822.00 KB | (841,728 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\msvcpx60.dll |
| scesrv | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 594.50 KB | (608,768 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\scesrv.dll |
| authz | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 165.50 KB | (169,472 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\authz.dll |
| umpnpgmr | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 201.00 KB | (205,824 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\umpnpgmr.dll |
| eventlog | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 124.00 KB | (126,976 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\eventlog.dll |
| lsass | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 13.50 KB | (13,824 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\lsass.exe |
| lsasrv | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.49 MB | (1,565,696 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\lsasrv.dll |
| ntdsapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 127.00 KB | (130,048 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\ntdsapi.dll |
| dnsapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 297.00 KB | (304,128 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\dnsapi.dll |
| samsrv | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.01 MB | (1,062,400 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\samsrv.dll |
| cryptdll | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 47.00 KB | (48,128 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\cryptdll.dll |
| msprivs | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 47.50 KB | (48,640 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\msprivs.dll |
| kerberos | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 702.00 KB | (718,848 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\kerberos.dll |
| msv1_0 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 251.00 KB | (257,024 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\msv1_0.dll |
| iphlpapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 176.50 KB | (180,736 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\iphlpapi.dll |
| netlogon | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 664.50 KB | (680,448 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\netlogon.dll |
| w32time | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 397.50 KB | (407,040 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\w32time.dll |
| schannel | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 249.00 KB | (254,976 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\schannel.dll |
| wdigest | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 130.50 KB | (133,632 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\wdigest.dll |
| rassfm | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 35.00 KB | (35,840 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\rassfm.dll |
| kdcsvc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 409.50 KB | (419,328 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\kdcsvc.dll |
| ntdsa | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 2.95 MB | (3,098,112 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\ntdsa.dll |
| esent | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 2.25 MB | (2,361,856 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\esent.dll |

| | | | | | | |
|----------|---|-----------|-------------------|-------------------|-----------------------|---------------------------------------|
| ntdsatq | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 50.50 KB | (51,712 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\ntdsatq.dll |
| mswsock | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 476.50 KB | (487,936 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\mswsock.dll |
| scecli | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 308.00 KB | (315,392 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\scecli.dll |
| ws03res | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 775.00 KB | (793,600 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\ws03res.dll |
| hnctfg | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 562.50 KB | (576,000 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\hnctfg.dll |
| wshtcpip | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 28.00 KB | (28,672 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\wshtcpip.dll |
| pstorsvc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 36.00 KB | (36,864 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\pstorsvc.dll |
| psbase | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 123.50 KB | (126,464 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\psbase.dll |
| dssenh | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 226.46 KB | (231,896 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\dssenh.dll |
| svchost | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 24.50 KB | (25,088 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\svchost.exe |
| rpss | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 671.50 KB | (687,616 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\rpss.dll |
| wkssvc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 219.50 KB | (224,768 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\wkssvc.dll |
| wiarpc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 57.00 KB | (58,368 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\wiarpc.dll |
| aelupsvc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 31.00 KB | (31,744 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\aelupsvc.dll |
| apphelp | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 241.00 KB | (246,784 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\apphelp.dll |
| dmserver | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 35.00 KB | (35,840 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\dmserver.dll |
| srvsvc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 155.00 KB | (158,720 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\srvsvc.dll |
| es | 2001.12.4720.1289 (srv03_sp1_rc1.041202-1618) | 357.00 KB | (365,568 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\es.dll |
| seclogon | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 26.50 KB | (27,136 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\seclogon.dll |
| wmisvc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 227.50 KB | (232,960 bytes) | 2/3/2005 11:50 AM | Microsoft Corporation | c:\windows\system32\wbem\wmisvc.dll |
| vssapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.26 MB | (1,319,424 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\vssapi.dll |
| comsvcs | 2001.12.4720.1289 (srv03_sp1_rc1.041202-1618) | 2.06 MB | (2,158,080 bytes) | 2/3/2005 11:50 AM | Microsoft Corporation | c:\windows\system32\comsvcs.dll |
| sens | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 63.00 KB | (64,512 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\sens.dll |
| wbemcore | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.24 MB | (1,300,992 bytes) | 2/3/2005 11:50 AM | Microsoft Corporation | c:\windows\system32\wbem\wbemcore.dll |
| esscli | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 627.00 KB | (642,048 bytes) | 2/3/2005 11:50 AM | Microsoft Corporation | c:\windows\system32\wbem\esscli.dll |
| wbemcomn | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 525.00 KB | (537,600 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\wbem\wbemcomn.dll |
| fastprox | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 868.00 KB | (888,832 bytes) | 2/3/2005 11:50 AM | Microsoft Corporation | c:\windows\system32\wbem\fastprox.dll |
| wmiutils | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 170.50 KB | (174,592 bytes) | 2/3/2005 11:50 AM | Microsoft Corporation | c:\windows\system32\wbem\wmiutils.dll |
| repdrvfs | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 350.50 KB | (358,912 bytes) | 2/3/2005 11:50 AM | Microsoft Corporation | c:\windows\system32\wbem\repdrvfs.dll |
| wmiprvsd | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 737.00 KB | (754,688 bytes) | 2/3/2005 11:50 AM | Microsoft Corporation | c:\windows\system32\wbem\wmiprvsd.dll |
| wbemess | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 533.00 KB | (545,792 bytes) | 2/3/2005 11:50 AM | Microsoft Corporation | c:\windows\system32\wbem\wbemess.dll |
| ncprov | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 72.50 KB | (74,240 bytes) | 2/3/2005 11:50 AM | Microsoft Corporation | c:\windows\system32\wbem\ncprov.dll |
| wbemsvc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 58.00 KB | (59,392 bytes) | 2/3/2005 11:50 AM | Microsoft Corporation | c:\windows\system32\wbem\wbemsvc.dll |
| wbemcons | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 65.50 KB | (67,072 bytes) | 2/3/2005 11:50 AM | Microsoft Corporation | c:\windows\system32\wbem\wbemcons.dll |
| actxprxy | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 220.50 KB | (225,792 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\actxprxy.dll |
| netman | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 457.50 KB | (468,480 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\netman.dll |
| netshell | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 2.33 MB | (2,438,144 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\netshell.dll |
| clusapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 125.50 KB | (128,512 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\clusapi.dll |
| rasapi32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 410.50 KB | (420,352 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\rasapi32.dll |
| rasman | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 93.00 KB | (95,232 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\rasman.dll |
| tapi32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 332.50 KB | (340,480 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\tapi32.dll |
| wininet | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.17 MB | (1,223,680 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation | c:\windows\system32\wininet.dll |
| wzcsapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 57.00 KB | (58,368 bytes) | 12/2/2004 2:10 PM | Microsoft Corporation | c:\windows\system32\wzcsapi.dll |
| wzcsvc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 491.00 KB | (502,784 bytes) | 12/2/2004 2:10 PM | Microsoft Corporation | c:\windows\system32\wzcsvc.dll |

| | | | | | | | |
|--|---|---------------------------|--------------------|--|---|--------------------------|-------------------|
| wmi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 7.00 KB | | db2install64 | 8.1.9.787 | 21.00 KB (21,504 bytes) | 2/4/2005 |
| (7,168 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | | 10:11 AM International Business Machines Corporation | | | |
| c:\windows\system32\wmi.dll | | | | c:\sql\bin\db2install64.dll | | | |
| dhcpcsvc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 218.00 KB | | db2dascmn64 | 8.1.9.787 | 98.00 KB (100,352 bytes) | 2/4/2005 |
| (223,232 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | | 10:11 AM International Business Machines Corporation | | | |
| c:\windows\system32\dhcpcsvc.dll | | | | c:\sql\bin\db2dascmn64.dll | | | |
| netcfgx | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.29 MB | | db2osse_db264 | 8.1.9.787 | 66.50 KB (68,096 bytes) | 2/4/2005 |
| (1,354,752 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | | 10:11 AM International Business Machines Corporation | | | |
| c:\windows\system32\netcfgx.dll | | | | c:\sql\bin\db2osse_db264.dll | | | |
| winipsec | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 52.00 KB | | db2sec64 | 8.1.9.787 | 23.50 KB (24,064 bytes) | 2/4/2005 10:11 AM |
| (53,248 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | | International Business Machines Corporation | | | |
| c:\windows\system32\winipsec.dll | | | | c:\sql\bin\db2sec64.dll | | | |
| wbemprox | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 37.50 KB | | db2trcapi64 | 8.1.9.787 | 33.00 KB (33,792 bytes) | 2/4/2005 |
| (38,400 bytes) | 2/3/2005 11:50 AM Microsoft Corporation | | | 10:11 AM International Business Machines Corporation | | | |
| c:\windows\system32\wbem\wbemprox.dll | | | | c:\sql\bin\db2trcapi64.dll | | | |
| rasdlg | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 859.50 KB | | termsrv | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 352.00 KB | |
| (880,128 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | | (360,448 bytes) | 2/3/2005 11:50 AM Microsoft Corporation | | |
| c:\windows\system32\rasdlg.dll | | | | c:\windows\system32\termsrv.dll | | | |
| rasadhlp | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 10.00 KB | | icaapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 26.50 KB | |
| (10,240 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | | (27,136 bytes) | 2/3/2005 11:50 AM Microsoft Corporation | | |
| c:\windows\system32\rasadhlp.dll | | | | c:\windows\system32\icaapi.dll | | | |
| pchsvc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 75.00 KB | | mstlsapi | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 187.00 KB | |
| (76,800 bytes) | 2/3/2005 11:52 AM Microsoft Corporation | | | (191,488 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| c:\windows\pchealth\helpctr\binaries\pchsvc.dll | | | | c:\windows\system32\mstlsapi.dll | | | |
| ersvc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 30.50 KB | | rdpwsx | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 170.63 KB | |
| (31,232 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | | (174,728 bytes) | 2/3/2005 11:50 AM Microsoft Corporation | | |
| c:\windows\system32\ersvc.dll | | | | c:\windows\system32\rdpwsx.dll | | | |
| wrshdnt | 2.23.00 | 92.00 KB (94,208 bytes) | 2/22/2005 12:16 PM | explorer | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.31 MB | |
| Denicomp Systems | c:\wrshdnt\wrshdnt.exe | | | (1,373,184 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| wow64 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 246.00 KB | | c:\windows\explorer.exe | | | |
| (251,904 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | | browseui | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.54 MB | |
| c:\windows\system32\wow64.dll | | | | (1,613,312 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| wow64win | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 278.50 KB | | c:\windows\system32\browseui.dll | | | |
| (285,184 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | | shdocvw | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 2.33 MB | |
| c:\windows\system32\wow64win.dll | | | | (2,444,800 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| wow64cpu | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 17.00 KB | | c:\windows\system32\shdocvw.dll | | | |
| (17,408 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | | cryptui | 5.131.3790.1289 (srv03_sp1_rc1.041202-1618) | 706.00 KB | |
| c:\windows\system32\wow64cpu.dll | | | | (722,944 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| db2sec | 8.1.9.787 | 8.50 KB (8,704 bytes) | 2/4/2005 10:11 AM | c:\windows\system32\cryptui.dll | | | |
| International Business Machines Corporation | c:\sql\bin\db2sec.exe | | | themeui | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 530.00 KB | |
| db2sys64 | 8.1.9.787 | 1.43 MB (1,503,744 bytes) | 2/4/2005 10:11 AM | (542,720 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| International Business Machines Corporation | c:\sql\bin\db2sys64.dll | | | c:\windows\system32\themeui.dll | | | |
| db2wint64 | 8.1.9.787 | 32.00 KB (32,768 bytes) | 2/4/2005 10:11 AM | msimg32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 6.50 KB | |
| International Business Machines Corporation | c:\sql\bin\db2wint64.dll | | | (6,656 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| db2sysp64 | 8.1.9.787 | 166.00 KB (169,984 bytes) | 2/4/2005 10:11 AM | c:\windows\system32\msimg32.dll | | | |
| International Business Machines Corporation | c:\sql\bin\db2sysp64.dll | | | linkinfo | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 30.50 KB | |
| db2app64 | 8.1.9.787 | 6.40 MB (6,713,344 bytes) | 2/4/2005 10:11 AM | (31,232 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| International Business Machines Corporation | c:\sql\bin\db2app64.dll | | | c:\windows\system32\linkinfo.dll | | | |
| db2osse64 | 8.1.9.787 | 1.90 MB (1,989,120 bytes) | 2/4/2005 10:11 AM | ntshrui | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 183.50 KB | |
| International Business Machines Corporation | c:\sql\bin\db2osse64.dll | | | (187,904 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| libmmd | 8.1.8.2 | 802.00 KB (821,248 bytes) | 2/4/2005 10:11 AM | c:\windows\system32\ntshrui.dll | | | |
| Intel Corporation | c:\sql\bin\libmmd.dll | | | urlmon | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 1,016.50 KB | |
| db2locale64 | 8.1.9.787 | 43.50 KB (44,544 bytes) | 2/4/2005 | (1,040,896 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| 10:11 AM International Business Machines Corporation | c:\sql\bin\db2locale64.dll | | | c:\windows\system32\urlmon.dll | | | |
| db2g11n64 | 8.1.9.787 | 437.50 KB (448,000 bytes) | 2/4/2005 | webcheck | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 440.50 KB | |
| 10:11 AM International Business Machines Corporation | c:\sql\bin\db2g11n64.dll | | | (451,072 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| db2genreg64 | 8.1.9.787 | 183.50 KB (187,904 bytes) | 2/4/2005 | c:\windows\system32\webcheck.dll | | | |
| 10:11 AM International Business Machines Corporation | c:\sql\bin\db2genreg64.dll | | | wsock32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 23.50 KB | |
| | | | | (24,064 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| | | | | c:\windows\system32\wsock32.dll | | | |
| | | | | stobject | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 141.50 KB | |
| | | | | (144,896 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| | | | | c:\windows\system32\stobject.dll | | | |
| | | | | batmeter | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 40.50 KB | |
| | | | | (41,472 bytes) | 12/3/2004 7:00 AM Microsoft Corporation | | |
| | | | | c:\windows\system32\batmeter.dll | | | |

| | | | | | |
|---|--|-----------|-------------------|-------------------|-----------------------|
| powrprof | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 31.00 KB | (31,744 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\powrprof.dll | | | | | |
| browselc | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 63.00 KB | (64,512 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\browselc.dll | | | | | |
| shdoclc | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 589.50 KB | (603,648 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\shdoclc.dll | | | | | |
| netplwiz | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 938.00 KB | (960,512 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\netplwiz.dll | | | | | |
| helpctr | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.30 MB | (1,365,504 bytes) | 2/3/2005 11:52 AM | Microsoft Corporation |
| c:\windows\pchealth\helpctr\binaries\helpctr.exe | | | | | |
| hcappres | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 7.50 KB | (7,680 bytes) | 2/3/2005 11:52 AM | Microsoft Corporation |
| c:\windows\pchealth\helpctr\binaries\hcappres.dll | | | | | |
| itss | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 208.00 KB | (212,992 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\itss.dll | | | | | |
| msxml3 | 8.70.1006.0 | 1.97 MB | (2,068,992 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\msxml3.dll | | | | | |
| pchshell | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 153.50 KB | (157,184 bytes) | 2/3/2005 11:52 AM | Microsoft Corporation |
| c:\windows\pchealth\helpctr\binaries\pchshell.dll | | | | | |
| mlang | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 687.00 KB | (703,488 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\mlang.dll | | | | | |
| mshtml | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 5.73 MB | (6,010,368 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\mshtml.dll | | | | | |
| mssl31 | 3.10.349.0 | 356.50 KB | (365,056 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\mssl31.dll | | | | | |
| msimtf | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 380.50 KB | (389,632 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\msimtf.dll | | | | | |
| msctf | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 617.00 KB | (631,808 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\msctf.dll | | | | | |
| jscript | 5.6.0.8827 | 983.00 KB | (1,006,592 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\jscript.dll | | | | | |
| imm32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 209.00 KB | (214,016 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\imm32.dll | | | | | |
| mshtml | 6.00.3790.1289 (srv03_sp1_rc1.041202-1618) | 912.50 KB | (934,400 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\mshtml.dll | | | | | |
| vbscript | 5.6.0.8827 | 647.00 KB | (662,528 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\vbscript.dll | | | | | |
| msinfo | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 635.50 KB | (650,752 bytes) | 2/3/2005 11:52 AM | Microsoft Corporation |
| c:\windows\pchealth\helpctr\binaries\msinfo.dll | | | | | |
| mfc42u | 6.50.9146.0 | 1.39 MB | (1,462,272 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\mfc42u.dll | | | | | |
| riched32 | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 5.50 KB | (5,632 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\riched32.dll | | | | | |
| riched20 | 5.31.23.1223 | 1.10 MB | (1,156,608 bytes) | 12/3/2004 7:00 AM | Microsoft Corporation |
| c:\windows\system32\riched20.dll | | | | | |
| helpsvc | 5.2.3790.1289 (srv03_sp1_rc1.041202-1618) | 1.52 MB | (1,590,272 bytes) | 2/3/2005 11:52 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 | |

| | | | | |
|---|---|--|----------------|---------------|
| Application Experience Lookup Service | AeLookupSvc | Running | | |
| 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 | Manual | 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\v1.1.4322\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 | Stopped | Manual | Share Process |
| c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | | 0 |
| Indexing Service | CiSvc | Stopped | Disabled | Share Process |
| c:\windows\system32\cisvc.exe | Normal | LocalSystem | | 0 |
| ClipBook | ClipSrv | Stopped | Disabled | Own Process |
| c:\windows\system32\clipsrv.exe | Normal | LocalSystem | | 0 |
| COM+ System Application | COMSysApp | Stopped | Manual | |
| Own Process | c:\windows\system32\dlhhost.exe | | | |
| /processid:{02d4b3f1-fd88-11d1-960d-00805fc79235} | Normal | | | |
| LocalSystem | 0 | | | |
| Cryptographic Services | CryptSvc | Stopped | Manual | Share |
| Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | | |
| LocalSystem | 0 | | | |
| DB2 - DB2 | DB2 | Stopped | Manual | Own Process |
| c:\sql\bin\db2syscs.exe | Normal | \Administrator | | 0 |
| DB2DAS - DB2DAS00 | DB2DAS00 | Stopped | Manual | |
| Own Process | "c:\sql\bin\db2dasrrm.exe" | Normal | | |
| LocalSystem | 0 | | | |
| DB2 Governor | DB2GOVERNOR | Stopped | Manual | Own |
| Process | "c:\sql\bin\db2govds.exe" | Normal | \administrator | 0 |
| DB2 JDBC Applet Server | DB2JDS | Stopped | Manual | Own |
| Process | "c:\sql\bin\db2jds.exe" | Normal | LocalSystem | 0 |
| DB2 License Server | DB2LICD | Stopped | Manual | Own Process |
| c:\sql\bin\db2licd.exe | Ignore | LocalSystem | | 0 |
| DB2 Registry Reflector | DB2NTREGREFLECTOR | Stopped | | |
| Manual | Own Process | "c:\sql\bin\db2reg64.exe" | Normal | |
| \administrator | 0 | | | |
| DB2 Security Server | DB2NTSECSEVER | Running | Auto | Own |
| Process | "c:\sql\bin\db2sec.exe" | Normal | LocalSystem | 0 |
| DB2 Remote Command Server | DB2REMOTECMD | Stopped | Manual | |
| Own Process | "c:\sql\bin\db2rcmd.exe" | Ignore | | |
| \administrator | 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 | Stopped | Manual | 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 Managerdmserver | Running | Auto | Share Process | |
| c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | | 0 |

| | | | | | | | | | |
|---|---|---------|-------------|-------------------------------|--|--|---------------------------------|-------------|---------------|
| DNS Client | Dnscache | Stopped | Manual | Share Process | Plug and Play | PlugPlay | Running | Auto | Share Process |
| c:\windows\system32\svchost.exe -k networkservice | | | | Normal | c:\windows\system32\services.exe | | | | Normal |
| AUTHORITY\NetworkService | | | | NT | IPSEC Services | PolicyAgent | Stopped | Manual | Share |
| Error Reporting Service | ERSvc | Running | Auto | Share | Process | c:\windows\system32\lsass.exe | Normal | LocalSystem | 0 |
| Process | c:\windows\system32\svchost.exe -k winerr | Ignore | | | Protected Storage | ProtectedStorage | Running | Auto | Share |
| LocalSystem | | | | 0 | Process | c:\windows\system32\lsass.exe | Normal | LocalSystem | 0 |
| Event Log | Eventlog | Running | Auto | Share Process | Remote Access Auto Connection Manager | RasAuto | Stopped | Manual | |
| c:\windows\system32\services.exe | | | | Normal | Share Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| COM+ Event System | EventSystem | Running | Auto | Share | LocalSystem | | | | 0 |
| Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | | | Remote Access Connection Manager | RasMan | Stopped | Manual | |
| LocalSystem | | | | 0 | Share Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| Help and Support | helpsvc | Running | Manual | Share Process | Remote Desktop Help Session Manager | RDSessMgr | Stopped | | |
| c:\windows\system32\svchost.exe -k netsvcs | | | | Normal | Manual | Own Process | c:\windows\system32\sessmgr.exe | | |
| Human Interface Device Access | HidServ | Stopped | Disabled | Share | Normal | LocalSystem | | | 0 |
| Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | | | Remote Shell Daemon | Remote Shell Daemon | Running | | |
| LocalSystem | | | | 0 | Auto | Own Process | c:\wrsdnt\wrsdnt.exe | Normal | |
| HTTP SSLHTTPFilter | | Stopped | Manual | Share Process | LocalSystem | | | | 0 |
| c:\windows\system32\lsass.exe | | Normal | LocalSystem | 0 | Routing and Remote Access | RemoteAccess | Stopped | Disabled | |
| IAS Jet Database Access | IASJet | Stopped | Manual | Share | Share Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| Process | c:\windows\syswow64\svchost.exe -k iasjet | Normal | | | LocalSystem | | | | 0 |
| LocalSystem | | | | 0 | Remote Registry | RemoteRegistry | Stopped | Manual | Share |
| IMAPI CD-Burning COM Service | | | | ImapiService | Process | c:\windows\system32\svchost.exe -k regsvc | Normal | NT | |
| Disabled | Own Process | | | c:\windows\system32\imapi.exe | AUTHORITY\LocalService | | | | 0 |
| LocalSystem | | | | 0 | Remote Procedure Call (RPC) Locator | RpcLocator | Stopped | | |
| Intersite Messaging | IsmServ | Stopped | Disabled | Own Process | Manual | Own Process | c:\windows\system32\locator.exe | Normal | |
| c:\windows\system32\ismserv.exe | | | | Normal | NT AUTHORITY\NetworkService | | | | 0 |
| Kerberos Key Distribution Center | kdc | Stopped | Disabled | Share | Remote Procedure Call (RPC) | RpcSs | Running | Auto | Share |
| Share Process | c:\windows\system32\lsass.exe | Normal | | | Process | c:\windows\system32\svchost.exe -k rpsvc | Normal | NT | |
| LocalSystem | | | | 0 | AUTHORITY\NetworkService | | | | 0 |
| Server | lanmanserver | Running | Auto | Share Process | Resultant Set of Policy Provider | RSOPProv | Stopped | Manual | Share |
| c:\windows\system32\svchost.exe -k netsvcs | | | | Normal | Process | c:\windows\system32\rsopprov.exe | Normal | LocalSystem | 0 |
| Workstation | lanmanworkstation | Running | Auto | Share | Special Administration Console Helper | sacsvr | Stopped | Manual | |
| Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | | | Share Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| LocalSystem | | | | 0 | Security Accounts Manager | SamSs | Running | Auto | Share |
| License Logging | LicenseService | Stopped | Disabled | Own | Process | c:\windows\system32\lsass.exe | Normal | LocalSystem | 0 |
| Process | c:\windows\system32\llssrv.exe | Normal | NT | | Smart Card | SCardSvr | Stopped | Manual | Share Process |
| AUTHORITY\NetworkService | | | | 0 | c:\windows\system32\scardsvr.exe | | Ignore | NT | |
| TCP/IP NetBIOS Helper | LmHosts | Running | Auto | Share | AUTHORITY\LocalService | | | | 0 |
| Process | c:\windows\system32\svchost.exe -k localservice | Normal | | | Task Scheduler | Schedule | Stopped | Manual | Share Process |
| NT AUTHORITY\LocalService | | | | 0 | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | | 0 |
| Messenger | Messenger | Stopped | Disabled | Share Process | Secondary Logon | seclogon | Running | Auto | Share Process |
| c:\windows\system32\svchost.exe -k netsvcs | | | | Normal | c:\windows\system32\svchost.exe -k netsvcs | Ignore | LocalSystem | | 0 |
| NetMeeting Remote Desktop Sharing | mnmsrvc | Stopped | Disabled | Own Process | System Event Notification | SENS | Running | Manual | Share |
| Own Process | c:\windows\system32\mnmsrvc.exe | Normal | | | Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| LocalSystem | | | | 0 | Distributed Transaction Coordinator | MSDTC | Running | Auto | |
| Distributed Transaction Coordinator | | | | MSDTC | Own Process | c:\windows\system32\msdtc.exe | Normal | NT | |
| Own Process | c:\windows\system32\msdtc.exe | Normal | NT | | AUTHORITY\NetworkService | | | | 0 |
| AUTHORITY\NetworkService | | | | 0 | Windows Installer | MSIServer | Stopped | Manual | Share Process |
| Windows Installer | MSIServer | Stopped | Manual | Share Process | c:\windows\system32\msiexec.exe /v | | | | Normal |
| c:\windows\system32\msiexec.exe /v | | | | Normal | Network DDE | NetDDE | Stopped | Disabled | Share Process |
| Network DDE | NetDDE | Stopped | Disabled | Share Process | c:\windows\system32\netdde.exe | | | | Normal |
| c:\windows\system32\netdde.exe | | | | LocalSystem | 0 | | | | 0 |
| Network DDE DSDM | NetDDEdsdm | Stopped | Disabled | Share | Process | c:\windows\system32\netdde.exe | Normal | LocalSystem | 0 |
| Process | c:\windows\system32\netdde.exe | Normal | LocalSystem | 0 | Net Logon | Netlogon | Stopped | Manual | Share Process |
| Net Logon | Netlogon | Stopped | Manual | Share Process | c:\windows\system32\lsass.exe | | | | Normal |
| c:\windows\system32\lsass.exe | | | | LocalSystem | 0 | | | | 0 |
| Network Connections | Netman | Running | Manual | Share Process | Network Connections | Netman | Running | Manual | Share Process |
| c:\windows\system32\svchost.exe -k netsvcs | | | | Normal | Share Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| Network Location Awareness (NLA) | Nla | Running | Manual | Share Process | LocalSystem | | | | 0 |
| Share Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | | | File Replication | NtFrs | Stopped | Manual | Own Process |
| LocalSystem | | | | 0 | c:\windows\system32\ntfrs.exe | | Ignore | LocalSystem | 0 |
| File Replication | NtFrs | Stopped | Manual | Own Process | NT LM Security Support Provider | NtLmSsp | Stopped | Manual | Share Process |
| c:\windows\system32\ntfrs.exe | | | | LocalSystem | Share Process | c:\windows\system32\lsass.exe | Normal | LocalSystem | 0 |
| NT LM Security Support Provider | | | | NTLmSsp | LocalSystem | | | | 0 |
| Share Process | c:\windows\system32\lsass.exe | Normal | | | Removable Storage | NtmsSvc | Stopped | Manual | Share Process |
| LocalSystem | | | | 0 | c:\windows\system32\svchost.exe -k netsvcs | | | | Normal |
| Removable Storage | NtmsSvc | Stopped | Manual | Share Process | LocalSystem | | | | 0 |
| c:\windows\system32\svchost.exe -k netsvcs | | | | Normal | LocalSystem | | | | 0 |

| | | | | | | |
|--|---|---------------------------|---------------------------|---------------|--|--|
| Terminal Services | TermService | Running | Manual | Share | Accessories\Communications | All Users:Accessories\Communications |
| Process | c:\windows\system32\svchost.exe -k termsvcs | | | Normal | All Users | |
| LocalSystem | 0 | | | | Accessories\Entertainment | All Users:Accessories\Entertainment |
| Themes | Themes | Stopped | Disabled | Share Process | All Users | |
| Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 | Accessories\System Tools | All Users:Accessories\System Tools |
| Telnet | TlntSvr | Stopped | Disabled | Own Process | All Users | |
| Process | c:\windows\system32\tlntsvr.exe | Normal | NT AUTHORITY\LocalService | 0 | ActiveState ActivePerl 5.8 | All Users:ActiveState ActivePerl 5.8 |
| Distributed Link Tracking Server | TrkSvr | Stopped | Disabled | Share | All Users | |
| Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 | Administrative Tools | All Users:Administrative Tools All Users |
| LocalSystem | 0 | | | | IBM DB2 | All Users:IBM DB2 All Users |
| Distributed Link Tracking Client | TrkWks | Stopped | Manual | Share | IBM DB2\Command Line Tools | All Users:IBM DB2\Command Line Tools |
| Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 | All Users | |
| LocalSystem | 0 | | | | IBM DB2\Development Tools | All Users:IBM DB2\Development Tools |
| Terminal Services Session Directory | Tssdis | Stopped | Disabled | | All Users | |
| Own Process | c:\windows\system32\tssdis.exe | Normal | LocalSystem | 0 | IBM DB2\General Administration Tools | All Users:IBM DB2\General Administration Tools All Users |
| LocalSystem | 0 | | | | IBM DB2\Information | All Users:IBM DB2\Information All Users |
| Windows User Mode Driver Framework | UMWdf | Stopped | Manual | | IBM DB2\Monitoring Tools | All Users:IBM DB2\Monitoring Tools |
| Own Process | c:\windows\system32\wdfmgr.exe | Normal | LocalSystem | 0 | All Users | |
| LocalSystem | 0 | | | | IBM DB2\Set-up Tools | All Users:IBM DB2\Set-up Tools |
| Uninterruptible Power Supply | UPS | Stopped | Manual | Own | All Users | |
| Process | c:\windows\system32\ups.exe | Normal | LocalSystem | 0 | ServeRAID Manager | All Users:ServeRAID Manager All Users |
| Virtual Disk Service | vds | Stopped | Manual | Own Process | Startup | All Users:Startup All Users |
| Process | c:\windows\system32\vds.exe | Normal | LocalSystem | 0 | WinZip | All Users:WinZip All Users |
| Volume Shadow Copy | VSS | Stopped | Manual | Own | Accessories | NT AUTHORITY\SYSTEM:Accessories NT |
| Process | c:\windows\system32\vssvc.exe | Normal | LocalSystem | 0 | AUTHORITY\SYSTEM | |
| Windows Time | W32Time | Stopped | Manual | Share Process | Accessories\Accessibility | NT |
| Process | c:\windows\system32\svchost.exe -k localservice | Normal | NT AUTHORITY\LocalService | 0 | AUTHORITY\SYSTEM:Accessories\Accessibility | NT |
| AUTHORITY\LocalService | 0 | | | | AUTHORITY\SYSTEM | |
| WebClient | WebClient | Stopped | Disabled | Share Process | Accessories\Entertainment | NT |
| Process | c:\windows\system32\svchost.exe -k localservice | Normal | NT AUTHORITY\LocalService | 0 | AUTHORITY\SYSTEM:Accessories\Entertainment | NT |
| AUTHORITY\LocalService | 0 | | | | AUTHORITY\SYSTEM | |
| WinHTTP Web Proxy Auto-Discovery Service | WinHttpAutoProxySvc | Stopped | Manual | Share Process | Startup | NT AUTHORITY\SYSTEM:Startup NT |
| Process | c:\windows\system32\svchost.exe -k localservice | Normal | NT AUTHORITY\LocalService | 0 | AUTHORITY\SYSTEM | |
| LocalSystem | 0 | | | | Accessories | DB2SERV1\Administrator:Accessories |
| Windows Management Instrumentation | winmgmt | Running | Auto | Share Process | DB2SERV1\Administrator | |
| Process | c:\windows\system32\svchost.exe -k netsvcs | Ignore | LocalSystem | 0 | Accessories\Accessibility | |
| LocalSystem | 0 | | | | DB2SERV1\Administrator:Accessories\Accessibility | |
| Portable Media Serial Number Service | WmdmPmSN | Stopped | Manual | Share Process | DB2SERV1\Administrator | |
| Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 | Accessories\Entertainment | |
| LocalSystem | 0 | | | | DB2SERV1\Administrator:Accessories\Entertainment | |
| Windows Management Instrumentation Driver Extensions | Wmi | Stopped | Manual | Share Process | DB2SERV1\Administrator | |
| Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 | Microsoft Visual C++ Toolkit 2003 | |
| LocalSystem | 0 | | | | DB2SERV1\Administrator:Microsoft Visual C++ Toolkit 2003 | |
| WMI Performance Adapter | WmiApSrv | Stopped | Manual | Own Process | DB2SERV1\Administrator | |
| Process | c:\windows\system32\wbem\wmiaprv.exe | Normal | LocalSystem | 0 | Startup | DB2SERV1\Administrator:Startup |
| LocalSystem | 0 | | | | DB2SERV1\Administrator | |
| Automatic Updates | wuauerv | Stopped | Manual | Share Process | | |
| Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 | | |
| LocalSystem | 0 | | | | | |
| Wireless Configuration | WZCSVC | Stopped | Manual | Share Process | | |
| Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 | | |
| LocalSystem | 0 | | | | | |
| Network Provisioning Service | xmlprov | Stopped | Manual | Share Process | | |
| Process | c:\windows\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 | | |
| LocalSystem | 0 | | | | | |
| [Program Groups] | | | | | | |
| Group Name | Name | User Name | | | | |
| Accessories | Default User: | Accessories | Default User | | | |
| Accessories\Accessibility | Default User: | Accessories\Accessibility | Default User | | | |
| Accessories\Entertainment | Default User: | Accessories\Entertainment | Default User | | | |
| Startup | Default User: | Startup | Default User | | | |
| Accessories | All Users: | Accessories | All Users | | | |
| Accessories\Accessibility | All Users: | Accessories\Accessibility | All Users | | | |
| All Users | | | | | | |
| [OLE Registration] | | | | | | |
| Object | Local Server | | | | | |
| Sound (OLE2) | sndrec32.exe | | | | | |
| Media Clip | mplay32.exe | | | | | |
| Video Clip | mplay32.exe /avi | | | | | |
| MIDI Sequence | mplay32.exe /mid | | | | | |
| Sound | Not Available | | | | | |
| Media Clip | Not Available | | | | | |

WordPad Document "%programfiles%\windows nt\accessories\wordpad.exe"
 Bitmap Image mspaint.exe

[Windows Error Reporting]

| Time | Type | Details |
|-------------------|------------------|--|
| 3/21/2005 1:08 PM | Application Hang | Hanging application mmc.exe, version 5.2.3790.1289, hang module hungapp, version 0.0.0.0, hang address 0x0000000000000000.
 |

[Internet Settings]

[Internet Explorer]

[Following are sub-categories of this main category]

[Summary]

| Item | Value |
|------------------|------------------------------------|
| Version | 6.0.3790.1289 |
| Build | 63790.1289 |
| Application Path | C:\Program Files\Internet Explorer |
| Language | English (United States) |
| Active Printer | Not Available |

| | |
|-----------------|----------|
| Cipher Strength | 128-bit |
| Content Advisor | Disabled |
| IEAK Install | No |

[File Versions]

| File | Version | Size | Date | Path | Company |
|--------------|----------------|---------------|----------------------|------------------------------------|-----------------------|
| actxprxy.dll | 6.0.3790.1289 | 221 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| advpack.dll | 6.0.3790.1289 | 146 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| asctrls.ocx | 6.0.3790.1289 | 147 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| browserc.dll | 6.0.3790.1289 | 63 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| browseui.dll | 6.0.3790.1289 | 1,576 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| cdfview.dll | 6.0.3790.1289 | 215 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| comctl32.dll | 5.82.3790.1289 | 936 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| dxtrans.dll | 6.3.3790.1289 | 326 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| dxtrans.dll | 6.3.3790.1289 | 555 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| iecont.dll | <File Missing> | Not Available | Not Available | Not Available | Not Available |
| iecontlc.dll | <File Missing> | Not Available | Not Available | Not Available | Not Available |
| iedkcs32.dll | 16.0.3790.1289 | 417 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| iepeers.dll | 6.0.3790.1289 | 360 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| iesetup.dll | 6.0.3790.1289 | 71 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| ieunit.inf | Not Available | 24 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Not Available |
| ieexplor.exe | 6.0.3790.1289 | 94 KB | 12/3/2004 7:00:00 AM | C:\Program Files\Internet Explorer | Microsoft Corporation |
| imgutil.dll | 6.0.3790.1289 | 62 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |

| | | | | | |
|--------------|----------------|---------------|----------------------|---------------------|-----------------------|
| inetctl.cpl | 6.0.3790.1289 | 428 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| inetctl.dll | 6.0.3790.1289 | 110 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| inseng.dll | 6.0.3790.1289 | 146 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| mlang.dll | 6.0.3790.1289 | 687 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| msencode.dll | <File Missing> | Not Available | Not Available | Not Available | Not Available |
| mshta.exe | 6.0.3790.1289 | 38 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| mshtml.dll | 6.0.3790.1289 | 5,870 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| mshtml.tlb | 6.0.3790.1289 | 1,320 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| mshtml.dll | 6.0.3790.1289 | 913 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| mshtmlr.dll | 6.0.3790.1289 | 56 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| msident.dll | 6.0.3790.1289 | 70 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| msidntld.dll | 6.0.3790.1289 | 16 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| msieftp.dll | 6.0.3790.1289 | 369 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| msrating.dll | 6.0.3790.1289 | 240 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| mstime.dll | 6.0.3790.1289 | 880 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| occache.dll | 6.0.3790.1289 | 126 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| proctexe.ocx | <File Missing> | Not Available | Not Available | Not Available | Not Available |
| sendmail.dll | 6.0.3790.1289 | 64 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| shdoclc.dll | 6.0.3790.1289 | 590 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| shdocvw.dll | 6.0.3790.1289 | 2,388 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| shfolder.dll | 6.0.3790.1289 | 34 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| shlwapi.dll | 6.0.3790.1289 | 609 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| tdc.ocx | 1.3.0.3130 | 92 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| url.dll | 6.0.3790.1289 | 40 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| urlmon.dll | 6.0.3790.1289 | 1,017 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| webcheck.dll | 6.0.3790.1289 | 441 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |
| wininet.dll | 6.0.3790.1289 | 1,195 KB | 12/3/2004 7:00:00 AM | C:\WINDOWS\system32 | Microsoft Corporation |

[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 | Custom |
| Internet | High |
| Restricted sites | Custom |

ServeRAID-6M Disk Controller Configuration Parameters

March 21, 2005 3:56:03 PM EST

Configuration summary

```

Server name.....db2serv1
ServeRAID Manager agent.....6.10.26 (1253)
ServeRAID Manager console.....6.10.26 (1253)
Number of controllers.....1
Operating system.....Windows 2003

```

Configuration information for controller 1

```

Controller type.....ServeRAID-6M
SCSI backend type.....AIC-7902
SCSI backend revision.....3
Controller FRU.....02R0998
Battery FRU.....02R0986
Serial number.....60B73939
Part number.....
Physical slot.....6
BIOS version.....7.10.18
Firmware version.....7.10.18
Device driver version.....7.10.52
Battery-backup cache.....Installed
Battery temperature.....Normal
Battery charge level.....100 %
Battery-backup cache size.....256 MB
Read-ahead cache mode.....Adaptive
Stripe-unit size.....64 KB
Rebuild rate.....High
Hot-swap rebuild.....Enabled
Copy back.....Enabled
Data scrubbing.....Enabled
Auto-synchronization.....Enabled
Clustering.....Disabled
Unattended mode.....Disabled
BIOS-compatibility mapping.....Limited
Number of arrays.....2
Number of logical drives.....2
Number of hot-spare drives.....0
Number of ready drives.....0

```

Spanned array 1

```

Array identifier.....1
Array size.....486010 MB
Free space.....0 MB
Number of logical drives.....1
Number of physical drives.....14

```

Arrays in spanned array 1

```

Array identifier.....B
Array size.....69430 MB
Stripe order (channel/device)...1/0 2/0
Number of physical drives.....2

```

```

Array identifier.....C
Array size.....69430 MB
Stripe order (channel/device)...1/1 2/1
Number of physical drives.....2

```

```

Array identifier.....D
Array size.....69430 MB
Stripe order (channel/device)...1/2 2/2
Number of physical drives.....2

```

```

Array identifier.....E
Array size.....69430 MB
Stripe order (channel/device)...1/3 2/3
Number of physical drives.....2

```

```

Array identifier.....F
Array size.....69430 MB
Stripe order (channel/device)...1/4 2/4
Number of physical drives.....2

```

```

Array identifier.....G
Array size.....69430 MB

```


Stripe order (channel/device)...1/5 2/5
Number of physical drives.....2

Array identifier.....H
Array size.....69430 MB
Stripe order (channel/device)...1/6 2/6
Number of physical drives.....2

Physical drives in array B

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0R106
Firmware level.....B85B
Channel.....1
SCSI ID.....0
Size.....34715 MB
State.....Online
Array letter.....B
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0QMSB
Firmware level.....B85B
Channel.....2
SCSI ID.....0
Size.....34715 MB
State.....Online
Array letter.....B
PFA error.....No

Physical drives in array C

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0PEAA
Firmware level.....B85B
Channel.....1
SCSI ID.....1
Size.....34715 MB
State.....Online
Array letter.....C
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0KW2R
Firmware level.....B85B
Channel.....2
SCSI ID.....1
Size.....34715 MB
State.....Online
Array letter.....C
PFA error.....No

Physical drives in array D

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0QPDR
Firmware level.....B85B
Channel.....1
SCSI ID.....2
Size.....34715 MB
State.....Online
Array letter.....D
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0LK0L
Firmware level.....B85B
Channel.....2
SCSI ID.....2
Size.....34715 MB
State.....Online
Array letter.....D
PFA error.....No

Physical drives in array E

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0QMT4
Firmware level.....B85B
Channel.....1
SCSI ID.....3
Size.....34715 MB
State.....Online
Array letter.....E
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0KR5T
Firmware level.....B85B
Channel.....2
SCSI ID.....3
Size.....34715 MB
State.....Online
Array letter.....E
PFA error.....No

Physical drives in array F

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0QPBFB
Firmware level.....B85B
Channel.....1
SCSI ID.....4
Size.....34715 MB

State.....Online
Array letter.....F
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0KW78
Firmware level.....B85B
Channel.....2
SCSI ID.....4
Size.....34715 MB
State.....Online
Array letter.....F
PFA error.....No

Physical drives in array G

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0R239
Firmware level.....B85B
Channel.....1
SCSI ID.....5
Size.....34715 MB
State.....Online
Array letter.....G
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0MMA5
Firmware level.....B85B
Channel.....2
SCSI ID.....5
Size.....34715 MB
State.....Online
Array letter.....G
PFA error.....No

Physical drives in array H

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0R1AF
Firmware level.....B85B
Channel.....1
SCSI ID.....6
Size.....34715 MB
State.....Online
Array letter.....H
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0MP10
Firmware level.....B85B

Channel.....2
SCSI ID.....6
Size.....34715 MB
State.....Online
Array letter.....H
PFA error.....No

Logical drives in spanned array 1

Logical drive.....2
Spanned array number1
State.....Okay
RAID level.....10
Data space.....243005 MB
Parity space.....243005 MB
Date created.....02/03/2005
Write-cache mode.....Write through
Merge-group number.....207
Merge-group state.....Non-shared

Array A

Array identifier.....A
Array size.....34715 MB
Free space.....0 MB
Number of logical drives.....1
Stripe order (channel/device)...2/14
Number of physical drives.....1

Logical drives in array A

Logical drive.....1
Array letter.....A
State.....Okay
RAID level.....0
Data space.....34715 MB
Parity space.....0 MB
Date created.....02/03/2005
Write-cache mode.....Write through
Merge-group number.....207
Merge-group state.....Non-shared

Physical drives in array A

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0MK56
Firmware level.....B85B
Channel.....2
SCSI ID.....14
Size.....34715 MB
State.....Online
Array letter.....A
PFA error.....No

SCSI channel 1

Number of drives.....7
SCSI transfer speed.....Optimal
SCSI initiator ID.....7

Type.....Hard disk drive
Vendor.....IBM-ESXS

Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0R106
Firmware level.....B85B
Channel.....1
SCSI ID.....0
Size.....34715 MB
State.....Online
Array letter.....B
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0PEAA
Firmware level.....B85B
Channel.....1
SCSI ID.....1
Size.....34715 MB
State.....Online
Array letter.....C
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0QPDR
Firmware level.....B85B
Channel.....1
SCSI ID.....2
Size.....34715 MB
State.....Online
Array letter.....D
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0QMT4
Firmware level.....B85B
Channel.....1
SCSI ID.....3
Size.....34715 MB
State.....Online
Array letter.....E
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0QPBF
Firmware level.....B85B
Channel.....1
SCSI ID.....4
Size.....34715 MB
State.....Online
Array letter.....F
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0R239

Firmware level.....B85B
Channel.....1
SCSI ID.....5
Size.....34715 MB
State.....Online
Array letter.....G
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0R1AF
Firmware level.....B85B
Channel.....1
SCSI ID.....6
Size.....34715 MB
State.....Online
Array letter.....H
PFA error.....No

Type.....Enclosure
Vendor.....IBM
Product or model number.....EXP400
Serial number.....23M0051
Firmware level.....D110
FRU type.....MIDPLANE
FRU vendor.....IBM
FRU date of manufacture.....07/2003
FRU part number.....59P4865
FRU serial number.....1373197
FRU type.....CARD
FRU vendor.....IBM
FRU date of manufacture.....07/2003
FRU part number.....59P4866
FRU serial number.....1R043373199
FRU type.....Power
FRU vendor.....IBM
FRU date of manufacture.....07/2003
FRU part number.....71P8146
FRU serial number.....1R045373201
FRU type.....Power
FRU vendor.....IBM
FRU date of manufacture.....07/2003
FRU part number.....71P8146
FRU serial number.....1R045373202
FRU type.....CARD
FRU vendor.....IBM
FRU date of manufacture.....07/2003
FRU part number.....59P4869
FRU serial number.....1R044373200
Channel.....1
SCSI ID.....15
Enclosure ID.....0
Enclosure status.....Okay
Fan 1 status.....Okay
Fan 2 status.....Okay
Power supply 1 status.....Okay
Power supply 2 status.....Okay
Temperature status.....Normal

SCSI channel 2

Number of drives.....8
SCSI transfer speed.....Optimal
SCSI initiator ID.....7
Type.....Hard disk drive

Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0QM5B
Firmware level.....B85B
Channel.....2
SCSI ID.....0
Size.....34715 MB
State.....Online
Array letter.....B
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0KW2R
Firmware level.....B85B
Channel.....2
SCSI ID.....1
Size.....34715 MB
State.....Online
Array letter.....C
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0LK0L
Firmware level.....B85B
Channel.....2
SCSI ID.....2
Size.....34715 MB
State.....Online
Array letter.....D
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0KR5T
Firmware level.....B85B
Channel.....2
SCSI ID.....3
Size.....34715 MB
State.....Online
Array letter.....E
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0KW78
Firmware level.....B85B
Channel.....2
SCSI ID.....4
Size.....34715 MB
State.....Online
Array letter.....F
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736

Serial number.....3HX0MMA5
Firmware level.....B85B
Channel.....2
SCSI ID.....5
Size.....34715 MB
State.....Online
Array letter.....G
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0MP10
Firmware level.....B85B
Channel.....2
SCSI ID.....6
Size.....34715 MB
State.....Online
Array letter.....H
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0MK56
Firmware level.....B85B
Channel.....2
SCSI ID.....14
Size.....34715 MB
State.....Online
Array letter.....A
PFA error.....No

Type.....Enclosure
Vendor.....IBM
Product or model number.....EXP400
Serial number.....23M0139
Firmware level.....D110
FRU type.....MIDPLANE
FRU vendor.....IBM
FRU date of manufacture.....07/2003
FRU part number.....59P4865
FRU serial number.....1379163
FRU type.....CARD
FRU vendor.....IBM
FRU date of manufacture.....07/2003
FRU part number.....59P4866
FRU serial number.....1R043379164
FRU type.....Power
FRU vendor.....IBM
FRU date of manufacture.....07/2003
FRU part number.....71P8146
FRU serial number.....1R045379167
FRU type.....Power
FRU vendor.....IBM
FRU date of manufacture.....07/2003
FRU part number.....71P8146
FRU serial number.....1R045379168
FRU type.....CARD
FRU vendor.....IBM
FRU date of manufacture.....07/2003
FRU part number.....59P4869
FRU serial number.....1R044379166
Channel.....2
SCSI ID.....15
Enclosure ID.....0

Enclosure status.....Okay
Fan 1 status.....Okay
Fan 2 status.....Okay
Power supply 1 status.....Okay
Power supply 2 status.....Okay
Temperature status.....Normal

End of the configuration information for controller 1

DS4500 Disk Subsystem Configuration

Rack 1

PROFILE FOR STORAGE SUBSYSTEM: RACK1 (3/21/05 3:40:27 PM)

SUMMARY-----

Number of controllers: 2
Number of arrays: 4
Total number of logical drives (includes an access logical drive): 5 of 2048 used
Number of standard logical drives: 4
Number of access logical drives: 1
Number of drives: 112
Supported drive types: Fibre (112)
Total hot spare drives: 0
Standby: 0
In use: 0
Access logical drive: None mapped
Default host type: Windows 2000/Server 2003 Non-Clustered (Host type index 2)
Current configuration
Firmware version: 06.10.06.00
NVRAM version: N1742F900R910V02
Pending configuration
Staged firmware download supported?: Yes
Firmware version: None
NVRAM version: None
Transferred on: None
NVRAM configured for batteries?: Yes
Start cache flushing at (in percentage): 80
Stop cache flushing at (in percentage): 80
Cache block size (in KB): 4
Media scan frequency (in days): Disabled
Failover alert delay (in minutes): 5
Feature enable identifier: 30353134300030353131310041B429DB
Storage Subsystem worldwide name (ID):
600A0B800013C6EA000000041B47D54

CONTROLLERS-----

Number of controllers: 2

Controller in Slot A

Status: Online

Current configuration

Firmware version: 06.10.06.00
Appware version: 06.10.06.00
Bootware version: 06.10.01.00
NVRAM version: N1742F900R910V02

Pending configuration

Firmware version: None
Appware version: None
Bootware version: None
NVRAM version: None
Transferred on: None

Board ID: 5884

Product ID: 1742-900
Product revision: 0520
Serial number: 1T41105140
Date of manufacture: March 25, 2004
Cache/processor size (MB): 1024/128
Date/Time: Mon Mar 21 15:38:55 EST 2005
Associated Logical Drives (* = Preferred Owner):
RACK1_0*, RACK1_1*
Ethernet port: 1
MAC address: 00:a0:b8:13:c6:ea
Host name: RACK1_A
Network configuration: Static
IP address: 192.168.128.100
Subnet mask: 255.255.255.0
Gateway: 0.0.0.0
Remote login: Disabled

Drive interface: Fibre

Channel: 1
Current ID: 125/0x1
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Drive interface: Fibre

Channel: 2
Current ID: 125/0x1
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Drive interface: Fibre

Channel: 3
Current ID: 125/0x1
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Drive interface: Fibre

Channel: 4
Current ID: 125/0x1
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Host interface: Fibre

Port: 1
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 126/0x0
NL-Port ID: 0x010000
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up
Topology: Fabric Attach
World-wide port name: 20:04:00:a0:b8:13:c6:eb
World-wide node name: 20:04:00:a0:b8:13:c6:ea
Part type: HPFC-5400 revision 6

Host interface: Fibre

Port: 2
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 1/0xE8
NL-Port ID: 0x000000
Maximum data rate: 2 Gbps
Current data rate: 1 Gbps
Data rate control: Switch
Link status: Down
Topology: Not available

World-wide port name: 20:04:00:a0:b8:13:c6:ec
World-wide node name: 20:04:00:a0:b8:13:c6:ea
Part type: HPFC-5400 revision 6

Controller in Slot B

Status: Online

Current configuration

Firmware version: 06.10.06.00
Appware version: 06.10.06.00
Bootware version: 06.10.01.00
NVSRAM version: N1742F900R910V02

Pending configuration

Firmware version: None
Appware version: None
Bootware version: None
NVSRAM version: None
Transferred on: None

Board ID: 5884

Product ID: 1742-900

Product revision: 0520

Serial number: 1T41105111

Date of manufacture: March 26, 2004

Cache/processor size (MB): 1024/128

Date/Time: Mon Mar 21 15:38:28 EST 2005

Associated Logical Drives (* = Preferred Owner):

RACK1_2*, RACK1_3*

Ethernet port: 1

MAC address: 00:a0:b8:13:c5:fa

Host name: RACK1_B

Network configuration: Static

IP address: 192.168.128.101

Subnet mask: 255.255.255.0

Gateway: 0.0.0.0

Remote login: Disabled

Drive interface: Fibre

Channel: 1

Current ID: 124/0x2

Maximum data rate: 2 Gbps

Current data rate: 2 Gbps

Data rate control: Switch

Link status: Up

Drive interface: Fibre

Channel: 2

Current ID: 124/0x2

Maximum data rate: 2 Gbps

Current data rate: 2 Gbps

Data rate control: Switch

Link status: Up

Drive interface: Fibre

Channel: 3

Current ID: 124/0x2

Maximum data rate: 2 Gbps

Current data rate: 2 Gbps

Data rate control: Switch

Link status: Up

Drive interface: Fibre

Channel: 4

Current ID: 124/0x2

Maximum data rate: 2 Gbps

Current data rate: 2 Gbps

Data rate control: Switch

Link status: Up

Host interface: Fibre

Port: 1

Current ID: Not applicable/0xFFFFFFFF

Preferred ID: 126/0x0

NL-Port ID: 0x010100

Maximum data rate: 2 Gbps

Current data rate: 2 Gbps

Data rate control: Switch

Link status: Up

Topology: Fabric Attach

World-wide port name: 20:05:00:a0:b8:13:c6:eb

World-wide node name: 20:05:00:a0:b8:13:c6:ea

Part type: HPFC-5400 revision 6

Host interface: Fibre

Port: 2

Current ID: Not applicable/0xFFFFFFFF

Preferred ID: 3/0xE2

NL-Port ID: 0x000000

Maximum data rate: 2 Gbps

Current data rate: 1 Gbps

Data rate control: Switch

Link status: Down

Topology: Not available

World-wide port name: 20:05:00:a0:b8:13:c6:ec

World-wide node name: 20:05:00:a0:b8:13:c6:ea

Part type: HPFC-5400 revision 6

ARRAYS-----

Number of arrays: 4

Array 1 (RAID 0)

Status: Online

Drive type: Fibre Channel

Enclosure loss protection: No

Current owner: Controller in slot A

Associated logical drives and free capacities:

RACK1_0 (935.238 GB)

Associated drives (in piece order):

Drive at Enclosure 0, Slot 1
Drive at Enclosure 0, Slot 2
Drive at Enclosure 0, Slot 3
Drive at Enclosure 0, Slot 4
Drive at Enclosure 0, Slot 5
Drive at Enclosure 0, Slot 6
Drive at Enclosure 0, Slot 7
Drive at Enclosure 0, Slot 8
Drive at Enclosure 0, Slot 9
Drive at Enclosure 0, Slot 10
Drive at Enclosure 0, Slot 11
Drive at Enclosure 0, Slot 12
Drive at Enclosure 0, Slot 13
Drive at Enclosure 0, Slot 14
Drive at Enclosure 1, Slot 1
Drive at Enclosure 1, Slot 2
Drive at Enclosure 1, Slot 3
Drive at Enclosure 1, Slot 4
Drive at Enclosure 1, Slot 5
Drive at Enclosure 1, Slot 6
Drive at Enclosure 1, Slot 7
Drive at Enclosure 1, Slot 8
Drive at Enclosure 1, Slot 9
Drive at Enclosure 1, Slot 10
Drive at Enclosure 1, Slot 11
Drive at Enclosure 1, Slot 12
Drive at Enclosure 1, Slot 13
Drive at Enclosure 1, Slot 14

Array 2 (RAID 0)

Status: Online

Drive type: Fibre Channel

Enclosure loss protection: No

Current owner: Controller in slot A

Associated logical drives and free capacities:

RACK1_1 (935.238 GB)

Associated drives (in piece order):

- Drive at Enclosure 2, Slot 1
- Drive at Enclosure 2, Slot 2
- Drive at Enclosure 2, Slot 3
- Drive at Enclosure 2, Slot 4
- Drive at Enclosure 2, Slot 5
- Drive at Enclosure 2, Slot 6
- Drive at Enclosure 2, Slot 7
- Drive at Enclosure 2, Slot 8
- Drive at Enclosure 2, Slot 9
- Drive at Enclosure 2, Slot 10
- Drive at Enclosure 2, Slot 11
- Drive at Enclosure 2, Slot 12
- Drive at Enclosure 2, Slot 13
- Drive at Enclosure 2, Slot 14
- Drive at Enclosure 3, Slot 1
- Drive at Enclosure 3, Slot 2
- Drive at Enclosure 3, Slot 3
- Drive at Enclosure 3, Slot 4
- Drive at Enclosure 3, Slot 5
- Drive at Enclosure 3, Slot 6
- Drive at Enclosure 3, Slot 7
- Drive at Enclosure 3, Slot 8
- Drive at Enclosure 3, Slot 9
- Drive at Enclosure 3, Slot 10
- Drive at Enclosure 3, Slot 11
- Drive at Enclosure 3, Slot 12
- Drive at Enclosure 3, Slot 13
- Drive at Enclosure 3, Slot 14

Array 3 (RAID 0)

Status: Online

Drive type: Fibre Channel

Enclosure loss protection: No

Current owner: Controller in slot B

Associated logical drives and free capacities:

RACK1_2 (935.238 GB)

Associated drives (in piece order):

- Drive at Enclosure 10, Slot 1
- Drive at Enclosure 10, Slot 2
- Drive at Enclosure 10, Slot 3
- Drive at Enclosure 10, Slot 4
- Drive at Enclosure 10, Slot 5
- Drive at Enclosure 10, Slot 6
- Drive at Enclosure 10, Slot 7
- Drive at Enclosure 10, Slot 8
- Drive at Enclosure 10, Slot 9
- Drive at Enclosure 10, Slot 10
- Drive at Enclosure 10, Slot 11
- Drive at Enclosure 10, Slot 12
- Drive at Enclosure 10, Slot 13
- Drive at Enclosure 10, Slot 14
- Drive at Enclosure 11, Slot 1
- Drive at Enclosure 11, Slot 2
- Drive at Enclosure 11, Slot 3
- Drive at Enclosure 11, Slot 4
- Drive at Enclosure 11, Slot 5
- Drive at Enclosure 11, Slot 6
- Drive at Enclosure 11, Slot 7
- Drive at Enclosure 11, Slot 8
- Drive at Enclosure 11, Slot 9
- Drive at Enclosure 11, Slot 10
- Drive at Enclosure 11, Slot 11
- Drive at Enclosure 11, Slot 12
- Drive at Enclosure 11, Slot 13
- Drive at Enclosure 11, Slot 14

Array 4 (RAID 0)

Status: Online

Drive type: Fibre Channel

Enclosure loss protection: No

Current owner: Controller in slot B

Associated logical drives and free capacities:

RACK1_3 (935.238 GB)

Associated drives (in piece order):

- Drive at Enclosure 12, Slot 1
- Drive at Enclosure 12, Slot 2
- Drive at Enclosure 12, Slot 3
- Drive at Enclosure 12, Slot 4
- Drive at Enclosure 12, Slot 5
- Drive at Enclosure 12, Slot 6
- Drive at Enclosure 12, Slot 7
- Drive at Enclosure 12, Slot 8
- Drive at Enclosure 12, Slot 9
- Drive at Enclosure 12, Slot 10
- Drive at Enclosure 12, Slot 11
- Drive at Enclosure 12, Slot 12
- Drive at Enclosure 12, Slot 13
- Drive at Enclosure 12, Slot 14
- Drive at Enclosure 13, Slot 1
- Drive at Enclosure 13, Slot 2
- Drive at Enclosure 13, Slot 3
- Drive at Enclosure 13, Slot 4
- Drive at Enclosure 13, Slot 5
- Drive at Enclosure 13, Slot 6
- Drive at Enclosure 13, Slot 7
- Drive at Enclosure 13, Slot 8
- Drive at Enclosure 13, Slot 9
- Drive at Enclosure 13, Slot 10
- Drive at Enclosure 13, Slot 11
- Drive at Enclosure 13, Slot 12
- Drive at Enclosure 13, Slot 13
- Drive at Enclosure 13, Slot 14

STANDARD LOGICAL DRIVES-----

SUMMARY

Number of standard logical drives: 4

See other Logical Drives sub-tabs for premium feature information.

| NAME | STATUS | CAPACITY | RAID | LEVEL | ARRAY |
|---------|---------|------------|------|-------|-------|
| RACK1_0 | Optimal | 935.238 GB | 0 | 1 | |
| RACK1_1 | Optimal | 935.238 GB | 0 | 2 | |
| RACK1_2 | Optimal | 935.238 GB | 0 | 3 | |
| RACK1_3 | Optimal | 935.238 GB | 0 | 4 | |

DETAILS

Logical Drive name: RACK1_0

Logical Drive ID: 60:0a:0b:80:00:13:c6:ea:00:00:00:03:41:bd:50:9a

Subsystem ID (SSID): 0

Status: Optimal

Drive type: Fibre Channel

Enclosure loss protection: No

Preferred owner: Controller in slot A

Current owner: Controller in slot A

Capacity: 935.238 GB

RAID level: 0

Segment size: 64 KB

Modification priority: Highest

Associated array: 1

Read cache: Disabled

Write cache: Disabled

Write cache without batteries: Disabled

Write cache with mirroring: Disabled

Flush write cache after (in seconds): 10.00

Cache read ahead multiplier: 0

Enable background media scan: Disabled

Media scan with redundancy check: Disabled

Logical Drive name: RACK1_1
 Logical Drive ID: 60:0a:0b:80:00:13:c6:ea:00:00:00:05:41:bd:50:d8
 Subsystem ID (SSID): 1
 Status: Optimal
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Preferred owner: Controller in slot A
 Current owner: Controller in slot A
 Capacity: 935.238 GB
 RAID level: 0
 Segment size: 64 KB
 Modification priority: Highest
 Associated array: 2
 Read cache: Disabled
 Write cache: Disabled
 Write cache without batteries: Disabled
 Write cache with mirroring: Disabled
 Flush write cache after (in seconds): 10.00
 Cache read ahead multiplier: 0
 Enable background media scan: Disabled
 Media scan with redundancy check: Disabled

Logical Drive name: RACK1_2
 Logical Drive ID: 60:0a:0b:80:00:13:c5:fa:00:00:00:01:41:bd:50:c5
 Subsystem ID (SSID): 2
 Status: Optimal
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Preferred owner: Controller in slot B
 Current owner: Controller in slot B
 Capacity: 935.238 GB
 RAID level: 0
 Segment size: 64 KB
 Modification priority: Highest
 Associated array: 3
 Read cache: Disabled
 Write cache: Disabled
 Write cache without batteries: Disabled
 Write cache with mirroring: Disabled
 Flush write cache after (in seconds): 10.00
 Cache read ahead multiplier: 0
 Enable background media scan: Disabled
 Media scan with redundancy check: Disabled

Logical Drive name: RACK1_3
 Logical Drive ID: 60:0a:0b:80:00:13:c5:fa:00:00:00:03:41:bd:50:f7
 Subsystem ID (SSID): 3
 Status: Optimal
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Preferred owner: Controller in slot B
 Current owner: Controller in slot B
 Capacity: 935.238 GB
 RAID level: 0
 Segment size: 64 KB
 Modification priority: Highest
 Associated array: 4
 Read cache: Disabled
 Write cache: Disabled
 Write cache without batteries: Disabled
 Write cache with mirroring: Disabled
 Flush write cache after (in seconds): 10.00
 Cache read ahead multiplier: 0
 Enable background media scan: Disabled
 Media scan with redundancy check: Disabled

MISSING LOGICAL DRIVES-----

Number of missing logical drives: 0

DRIVES-----

SUMMARY

Number of drives: 112
 Supported drive types: Fibre (112)

BASIC:

| TRAY | SLOT | STATUS | CAPACITY | CURRENT DATA RATE |
|------------|----------|-----------|----------|-------------------|
| PRODUCT ID | FIRMWARE | VERSION | | |
| 0, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |

| | | | | | | |
|--------|---------|-----------|--------|------------|---|------|
| 12, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |

DRIVE CHANNELS:

| TRAY, SLOT | PREFERRED CHANNEL | REDUNDANT CHANNEL |
|------------|-------------------|-------------------|
|------------|-------------------|-------------------|

| | | |
|-------|---|---|
| 0, 1 | 4 | 3 |
| 0, 2 | 3 | 4 |
| 0, 3 | 4 | 3 |
| 0, 4 | 3 | 4 |
| 0, 5 | 4 | 3 |
| 0, 6 | 3 | 4 |
| 0, 7 | 4 | 3 |
| 0, 8 | 3 | 4 |
| 0, 9 | 4 | 3 |
| 0, 10 | 3 | 4 |
| 0, 11 | 4 | 3 |
| 0, 12 | 3 | 4 |
| 0, 13 | 4 | 3 |
| 0, 14 | 3 | 4 |
| 1, 1 | 4 | 3 |
| 1, 2 | 3 | 4 |
| 1, 3 | 4 | 3 |
| 1, 4 | 3 | 4 |
| 1, 5 | 4 | 3 |
| 1, 6 | 3 | 4 |
| 1, 7 | 4 | 3 |
| 1, 8 | 3 | 4 |
| 1, 9 | 4 | 3 |
| 1, 10 | 3 | 4 |
| 1, 11 | 4 | 3 |
| 1, 12 | 3 | 4 |
| 1, 13 | 4 | 3 |
| 1, 14 | 3 | 4 |
| 2, 1 | 4 | 3 |
| 2, 2 | 3 | 4 |

| | | |
|--------|---|---|
| 2, 3 | 4 | 3 |
| 2, 4 | 3 | 4 |
| 2, 5 | 4 | 3 |
| 2, 6 | 3 | 4 |
| 2, 7 | 4 | 3 |
| 2, 8 | 3 | 4 |
| 2, 9 | 4 | 3 |
| 2, 10 | 3 | 4 |
| 2, 11 | 4 | 3 |
| 2, 12 | 3 | 4 |
| 2, 13 | 4 | 3 |
| 2, 14 | 3 | 4 |
| 3, 1 | 4 | 3 |
| 3, 2 | 3 | 4 |
| 3, 3 | 4 | 3 |
| 3, 4 | 3 | 4 |
| 3, 5 | 4 | 3 |
| 3, 6 | 3 | 4 |
| 3, 7 | 4 | 3 |
| 3, 8 | 3 | 4 |
| 3, 9 | 4 | 3 |
| 3, 10 | 3 | 4 |
| 3, 11 | 4 | 3 |
| 3, 12 | 3 | 4 |
| 3, 13 | 4 | 3 |
| 3, 14 | 3 | 4 |
| 10, 1 | 2 | 1 |
| 10, 2 | 1 | 2 |
| 10, 3 | 2 | 1 |
| 10, 4 | 1 | 2 |
| 10, 5 | 2 | 1 |
| 10, 6 | 1 | 2 |
| 10, 7 | 2 | 1 |
| 10, 8 | 1 | 2 |
| 10, 9 | 2 | 1 |
| 10, 10 | 1 | 2 |
| 10, 11 | 2 | 1 |
| 10, 12 | 1 | 2 |
| 10, 13 | 2 | 1 |
| 10, 14 | 1 | 2 |
| 11, 1 | 2 | 1 |
| 11, 2 | 1 | 2 |
| 11, 3 | 2 | 1 |
| 11, 4 | 1 | 2 |
| 11, 5 | 2 | 1 |
| 11, 6 | 1 | 2 |
| 11, 7 | 2 | 1 |
| 11, 8 | 1 | 2 |
| 11, 9 | 2 | 1 |
| 11, 10 | 1 | 2 |
| 11, 11 | 2 | 1 |
| 11, 12 | 1 | 2 |
| 11, 13 | 2 | 1 |
| 11, 14 | 1 | 2 |
| 12, 1 | 2 | 1 |
| 12, 2 | 1 | 2 |
| 12, 3 | 2 | 1 |
| 12, 4 | 1 | 2 |
| 12, 5 | 2 | 1 |
| 12, 6 | 1 | 2 |
| 12, 7 | 2 | 1 |
| 12, 8 | 1 | 2 |
| 12, 9 | 2 | 1 |
| 12, 10 | 1 | 2 |
| 12, 11 | 2 | 1 |
| 12, 12 | 1 | 2 |
| 12, 13 | 2 | 1 |
| 12, 14 | 1 | 2 |

| | | |
|--------|---|---|
| 13, 1 | 2 | 1 |
| 13, 2 | 1 | 2 |
| 13, 3 | 2 | 1 |
| 13, 4 | 1 | 2 |
| 13, 5 | 2 | 1 |
| 13, 6 | 1 | 2 |
| 13, 7 | 2 | 1 |
| 13, 8 | 1 | 2 |
| 13, 9 | 2 | 1 |
| 13, 10 | 1 | 2 |
| 13, 11 | 2 | 1 |
| 13, 12 | 1 | 2 |
| 13, 13 | 2 | 1 |
| 13, 14 | 1 | 2 |

HOT SPARE COVERAGE:

The following arrays are not protected: 2, 1, 4, 3

Total hot spare drives: 0
 Standby: 0
 In use: 0

DETAILS

Drive at Enclosure 0, Slot 1

Drive port: 1, Channel: 4, ID: 0/0xEF
 Drive port: 2, Channel: 3, ID: 0/0xEF
 Drive path redundancy: OK
 Status: Optimal
 Raw capacity: 33.902 GB
 Usable capacity: 33.402 GB
 Current data rate: 2 Gbps
 Product ID: ST336753FC F
 Firmware version: B954
 Serial number: 3HX2NEQ200007502AZRN
 Vendor: IBM-ESXS
 Date of manufacture: July 21, 2004
 World-wide name: 20:00:00:0c:50:45:ab:88
 Drive type: Fibre Channel
 Speed: 15015 RPM
 Mode: Assigned
 Associated array: 1

Drive at Enclosure 0, Slot 2

Drive port: 1, Channel: 3, ID: 1/0xE8
 Drive port: 2, Channel: 4, ID: 1/0xE8
 Drive path redundancy: OK
 Status: Optimal
 Raw capacity: 33.902 GB
 Usable capacity: 33.402 GB
 Current data rate: 2 Gbps
 Product ID: ST336753FC F
 Firmware version: B954
 Serial number: 3HX2DMZH000074478XSY
 Vendor: IBM-ESXS
 Date of manufacture: May 26, 2004
 World-wide name: 20:00:00:0c:50:d6:cf:29
 Drive type: Fibre Channel
 Speed: 15015 RPM
 Mode: Assigned
 Associated array: 1

Drive at Enclosure 0, Slot 3

Drive port: 1, Channel: 4, ID: 2/0xE4
 Drive port: 2, Channel: 3, ID: 2/0xE4
 Drive path redundancy: OK
 Status: Optimal
 Raw capacity: 33.902 GB
 Usable capacity: 33.402 GB

Current data rate: 2 Gbps
 Product ID: ST336753FC F
 Firmware version: B954
 Serial number: 3HX2EAQ400007446H3ZS
 Vendor: IBM-ESXS
 Date of manufacture: May 26, 2004
 World-wide name: 20:00:00:0c:50:d6:cf:96
 Drive type: Fibre Channel
 Speed: 15015 RPM
 Mode: Assigned
 Associated array: 1

Drive at Enclosure 0, Slot 4

Drive port: 1, Channel: 3, ID: 3/0xE2
 Drive port: 2, Channel: 4, ID: 3/0xE2
 Drive path redundancy: OK
 Status: Optimal

Raw capacity: 33.902 GB
 Usable capacity: 33.402 GB
 Current data rate: 2 Gbps
 Product ID: ST336753FC F
 Firmware version: B954
 Serial number: 3HX26SJZ00007440BFBY
 Vendor: IBM-ESXS
 Date of manufacture: May 31, 2004
 World-wide name: 20:00:00:0c:50:d6:fc:e9
 Drive type: Fibre Channel
 Speed: 15015 RPM
 Mode: Assigned
 Associated array: 1

Drive at Enclosure 0, Slot 5

Drive port: 1, Channel: 4, ID: 4/0xE1
 Drive port: 2, Channel: 3, ID: 4/0xE1
 Drive path redundancy: OK
 Status: Optimal

Raw capacity: 33.902 GB
 Usable capacity: 33.402 GB
 Current data rate: 2 Gbps
 Product ID: ST336753FC F
 Firmware version: B954
 Serial number: 3HX2DWG200007447AYA1
 Vendor: IBM-ESXS
 Date of manufacture: May 26, 2004
 World-wide name: 20:00:00:0c:50:d6:cf:fe
 Drive type: Fibre Channel
 Speed: 15015 RPM
 Mode: Assigned
 Associated array: 1

Drive at Enclosure 0, Slot 6

Drive port: 1, Channel: 3, ID: 5/0xE0
 Drive port: 2, Channel: 4, ID: 5/0xE0
 Drive path redundancy: OK
 Status: Optimal

Raw capacity: 33.902 GB
 Usable capacity: 33.402 GB
 Current data rate: 2 Gbps
 Product ID: ST336753FC F
 Firmware version: B954
 Serial number: 3HX2DQTT00007447NFYS
 Vendor: IBM-ESXS
 Date of manufacture: May 26, 2004
 World-wide name: 20:00:00:0c:50:d6:d8:5b
 Drive type: Fibre Channel
 Speed: 15015 RPM
 Mode: Assigned
 Associated array: 1

Drive at Enclosure 0, Slot 7
Drive port: 1, Channel: 4, ID: 6/0xDC
Drive port: 2, Channel: 3, ID: 6/0xDC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DBPB00007447QTBR
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:ce:3f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 8
Drive port: 1, Channel: 3, ID: 7/0xDA
Drive port: 2, Channel: 4, ID: 7/0xDA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DEDC0000744780SZ
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:b9
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 9
Drive port: 1, Channel: 4, ID: 80/0x55
Drive port: 2, Channel: 3, ID: 80/0x55
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MKSK000075029SW2
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:c0
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 10
Drive port: 1, Channel: 3, ID: 96/0x3A
Drive port: 2, Channel: 4, ID: 96/0x3A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954

Serial number: 3HX2MYRF00007502CJ80
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ac:8e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 11
Drive port: 1, Channel: 4, ID: 64/0x72
Drive port: 2, Channel: 3, ID: 64/0x72
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DJA800007447NFLS
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:6e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 12
Drive port: 1, Channel: 3, ID: 72/0x67
Drive port: 2, Channel: 4, ID: 72/0x67
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DY1L0000744270NY
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d8:e5
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 13
Drive port: 1, Channel: 4, ID: 88/0x4B
Drive port: 2, Channel: 3, ID: 88/0x4B
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E23P00007447LWJT
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:ce:8e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 14
Drive port: 1, Channel: 3, ID: 104/0x2E

Drive port: 2, Channel: 4, ID: 104/0x2E
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EAPC00007447KV7V
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:d9
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 1

Drive port: 1, Channel: 4, ID: 8/0xD9
Drive port: 2, Channel: 3, ID: 8/0xD9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2CALH00007447RB09
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d7:c9
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 2

Drive port: 1, Channel: 3, ID: 9/0xD6
Drive port: 2, Channel: 4, ID: 9/0xD6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E2320000744780ME
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d0:8a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 3

Drive port: 1, Channel: 4, ID: 10/0xD5
Drive port: 2, Channel: 3, ID: 10/0xD5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DND9000074478XSF
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004

World-wide name: 20:00:00:0c:50:d6:ce:80
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 4

Drive port: 1, Channel: 3, ID: 11/0xD4
Drive port: 2, Channel: 4, ID: 11/0xD4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N1E700007453V842
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:16
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 5

Drive port: 1, Channel: 4, ID: 12/0xD3
Drive port: 2, Channel: 3, ID: 12/0xD3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EE7B00007447LW2A
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:d3
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 6

Drive port: 1, Channel: 3, ID: 13/0xD2
Drive port: 2, Channel: 4, ID: 13/0xD2
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MZXQ00007502CWDB
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:9f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 7

Drive port: 1, Channel: 4, ID: 14/0xD1
Drive port: 2, Channel: 3, ID: 14/0xD1
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MNA4000075029S1Y
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a9:70
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 8

Drive port: 1, Channel: 3, ID: 15/0xCE
Drive port: 2, Channel: 4, ID: 15/0xCE
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DD8700007447LWKE
Vendor: IBM-ESXS
Date of manufacture: May 25, 2004
World-wide name: 20:00:00:0c:50:d6:cf:c1
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 9

Drive port: 1, Channel: 4, ID: 81/0x54
Drive port: 2, Channel: 3, ID: 81/0x54
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DTA700007447QSUW
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d0:0b
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 10

Drive port: 1, Channel: 3, ID: 97/0x39
Drive port: 2, Channel: 4, ID: 97/0x39
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2HAKT00007502BES1
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:9b
Drive type: Fibre Channel
Speed: 15015 RPM

Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 11

Drive port: 1, Channel: 4, ID: 65/0x71
Drive port: 2, Channel: 3, ID: 65/0x71
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EBND00007447BVRL
Vendor: IBM-ESXS
Date of manufacture: May 25, 2004
World-wide name: 20:00:00:0c:50:d6:d0:38
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 12

Drive port: 1, Channel: 3, ID: 73/0x66
Drive port: 2, Channel: 4, ID: 73/0x66
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DN0000007447829M
Vendor: IBM-ESXS
Date of manufacture: May 25, 2004
World-wide name: 20:00:00:0c:50:d6:cf:33
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 13

Drive port: 1, Channel: 4, ID: 89/0x4A
Drive port: 2, Channel: 3, ID: 89/0x4A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DQVQ00007447NFNF
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:64
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 14

Drive port: 1, Channel: 3, ID: 105/0x2D
Drive port: 2, Channel: 4, ID: 105/0x2D
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps

Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DK1V00007447QTH8
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:07
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 2, Slot 1

Drive port: 1, Channel: 4, ID: 16/0xCD
Drive port: 2, Channel: 3, ID: 16/0xCD
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NENC00007453W8P0
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:9a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 2

Drive port: 1, Channel: 3, ID: 17/0xCC
Drive port: 2, Channel: 4, ID: 17/0xCC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DW7C000074478XPC
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:db:74
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 3

Drive port: 1, Channel: 4, ID: 18/0xCB
Drive port: 2, Channel: 3, ID: 18/0xCB
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DSFV00007447825S
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:f9
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 4

Drive port: 1, Channel: 3, ID: 19/0xCA
Drive port: 2, Channel: 4, ID: 19/0xCA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DR2W000074478XTE
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:f3
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 5

Drive port: 1, Channel: 4, ID: 20/0xC9
Drive port: 2, Channel: 3, ID: 20/0xC9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DDRQ00007447AYBB
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:35
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 6

Drive port: 1, Channel: 3, ID: 21/0xC7
Drive port: 2, Channel: 4, ID: 21/0xC7
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E1800007447RBGZ
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:db:1b
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 7

Drive port: 1, Channel: 4, ID: 22/0xC6
Drive port: 2, Channel: 3, ID: 22/0xC6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2CXVS0000744781LS

Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:30
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 8

Drive port: 1, Channel: 3, ID: 23/0xC5
Drive port: 2, Channel: 4, ID: 23/0xC5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DNKW00007447KU1N
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:89
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 9

Drive port: 1, Channel: 4, ID: 82/0x53
Drive port: 2, Channel: 3, ID: 82/0x53
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DNBQ00007447KUQU
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:ce:97
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 10

Drive port: 1, Channel: 3, ID: 98/0x36
Drive port: 2, Channel: 4, ID: 98/0x36
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NF3900007453DWQG
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:df
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 11

Drive port: 1, Channel: 4, ID: 66/0x6E
Drive port: 2, Channel: 3, ID: 66/0x6E

Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E9JK00007447BWBZ
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:e0
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 12

Drive port: 1, Channel: 3, ID: 74/0x65
Drive port: 2, Channel: 4, ID: 74/0x65
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EBR800007447AYAM
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:1a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 13

Drive port: 1, Channel: 4, ID: 90/0x49
Drive port: 2, Channel: 3, ID: 90/0x49
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E2RR00007447RBFU
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:db:1d
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 14

Drive port: 1, Channel: 3, ID: 106/0x2C
Drive port: 2, Channel: 4, ID: 106/0x2C
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E1WB00007441YPTM
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:cf:b1

Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 1

Drive port: 1, Channel: 4, ID: 24/0xC3
Drive port: 2, Channel: 3, ID: 24/0xC3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2CZ1Q00007447QT8K
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:d9:98
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 2

Drive port: 1, Channel: 3, ID: 25/0xBC
Drive port: 2, Channel: 4, ID: 25/0xBC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EDZA00007447RB32
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:48
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 3

Drive port: 1, Channel: 4, ID: 26/0xBA
Drive port: 2, Channel: 3, ID: 26/0xBA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EC0100007447QT8X
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:a4
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 4

Drive port: 1, Channel: 3, ID: 27/0xB9
Drive port: 2, Channel: 4, ID: 27/0xB9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB

Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MVG900007502CJB5
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a7:d3
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 5

Drive port: 1, Channel: 4, ID: 28/0xB6
Drive port: 2, Channel: 3, ID: 28/0xB6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E9A6000074478XRP
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:19
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 6

Drive port: 1, Channel: 3, ID: 29/0xB5
Drive port: 2, Channel: 4, ID: 29/0xB5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E9K00007447QT58
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:d9:a7
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 7

Drive port: 1, Channel: 4, ID: 30/0xB4
Drive port: 2, Channel: 3, ID: 30/0xB4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DHMD00007447KVCS
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:2b
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned

Associated array: 2

Drive at Enclosure 3, Slot 8

Drive port: 1, Channel: 3, ID: 31/0xB3
Drive port: 2, Channel: 4, ID: 31/0xB3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DHSG000074445YU0
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:2e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 9

Drive port: 1, Channel: 4, ID: 83/0x52
Drive port: 2, Channel: 3, ID: 83/0x52
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E1EP00007447BW3T
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:26
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 10

Drive port: 1, Channel: 3, ID: 99/0x35
Drive port: 2, Channel: 4, ID: 99/0x35
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DCS600007447BW2Q
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:23
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 11

Drive port: 1, Channel: 4, ID: 67/0x6D
Drive port: 2, Channel: 3, ID: 67/0x6D
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F

Firmware version: B954
Serial number: 3HX2NF23000075029RX3
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ab:ec
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 12

Drive port: 1, Channel: 3, ID: 75/0x63
Drive port: 2, Channel: 4, ID: 75/0x63
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEL600007502CJ9M
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:96
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 13

Drive port: 1, Channel: 4, ID: 91/0x47
Drive port: 2, Channel: 3, ID: 91/0x47
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EA5M00007447RB8L
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:b6
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 14

Drive port: 1, Channel: 3, ID: 107/0x2B
Drive port: 2, Channel: 4, ID: 107/0x2B
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MSP5000075029T24
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ac:92
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 10, Slot 1

Drive port: 1, Channel: 2, ID: 0/0xEF
Drive port: 2, Channel: 1, ID: 0/0xEF
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DSV200007447AXF9
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:b1:7a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 2
Drive port: 1, Channel: 1, ID: 1/0xE8
Drive port: 2, Channel: 2, ID: 1/0xE8
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EAKA00007447QTAN
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:25
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 3
Drive port: 1, Channel: 2, ID: 2/0xE4
Drive port: 2, Channel: 1, ID: 2/0xE4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E33B00007447LX6C
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d8:d3
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 4
Drive port: 1, Channel: 1, ID: 3/0xE2
Drive port: 2, Channel: 2, ID: 3/0xE2
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E06Q0000744478X3
Vendor: IBM-ESXS

Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:6f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 5
Drive port: 1, Channel: 2, ID: 4/0xE1
Drive port: 2, Channel: 1, ID: 4/0xE1
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EC0G0000744781XQ
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:d9:e6
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 6
Drive port: 1, Channel: 1, ID: 5/0xE0
Drive port: 2, Channel: 2, ID: 5/0xE0
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E2QQ00007447AXRA
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:28
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 7
Drive port: 1, Channel: 2, ID: 6/0xDC
Drive port: 2, Channel: 1, ID: 6/0xDC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2L2ZQ00007502A7DU
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a9:72
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 8
Drive port: 1, Channel: 1, ID: 7/0xDA
Drive port: 2, Channel: 2, ID: 7/0xDA
Drive path redundancy: OK

Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E1A600007447BW8L
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:56
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 9
Drive port: 1, Channel: 2, ID: 80/0x55
Drive port: 2, Channel: 1, ID: 80/0x55
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NENQ00007502CJ2A
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:24
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 10
Drive port: 1, Channel: 1, ID: 96/0x3A
Drive port: 2, Channel: 2, ID: 96/0x3A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MDLM00007502CWGK
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a7:e1
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 11
Drive port: 1, Channel: 2, ID: 64/0x72
Drive port: 2, Channel: 1, ID: 64/0x72
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NET600007502A7NZ
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:aa:b9
Drive type: Fibre Channel

Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 12
Drive port: 1, Channel: 1, ID: 72/0x67
Drive port: 2, Channel: 2, ID: 72/0x67
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MSE400007502CWHG
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:aa
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 13
Drive port: 1, Channel: 2, ID: 88/0x4B
Drive port: 2, Channel: 1, ID: 88/0x4B
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N5VY00007502CJ9R
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a8:31
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 14
Drive port: 1, Channel: 1, ID: 104/0x2E
Drive port: 2, Channel: 2, ID: 104/0x2E
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N3MJ00007502CWCS
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:a7
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 1
Drive port: 1, Channel: 2, ID: 8/0xD9
Drive port: 2, Channel: 1, ID: 8/0xD9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB

Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DVH700007447LVRP
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d7:24
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 2
Drive port: 1, Channel: 1, ID: 9/0xD6
Drive port: 2, Channel: 2, ID: 9/0xD6
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEJT000075029S2D
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:6f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 3
Drive port: 1, Channel: 2, ID: 10/0xD5
Drive port: 2, Channel: 1, ID: 10/0xD5
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E8CY00007447QTDX
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:5e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 4
Drive port: 1, Channel: 1, ID: 11/0xD4
Drive port: 2, Channel: 2, ID: 11/0xD4
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DZXC00007447QTA9
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:31
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 5
Drive port: 1, Channel: 2, ID: 12/0xD3
Drive port: 2, Channel: 1, ID: 12/0xD3
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DKSL00007447LX3E
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:45
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 6
Drive port: 1, Channel: 1, ID: 13/0xD2
Drive port: 2, Channel: 2, ID: 13/0xD2
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2CSGG00007447AYLX
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:ce:ae
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 7
Drive port: 1, Channel: 2, ID: 14/0xD1
Drive port: 2, Channel: 1, ID: 14/0xD1
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DNC60000744779DE
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:ce:e0
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 8
Drive port: 1, Channel: 1, ID: 15/0xCE
Drive port: 2, Channel: 2, ID: 15/0xCE
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954

Serial number: 3HX2EBVK00007447AYE9
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:a0
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 9

Drive port: 1, Channel: 2, ID: 81/0x54
Drive port: 2, Channel: 1, ID: 81/0x54
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DQYL00007447RB9K
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d8:96
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 10

Drive port: 1, Channel: 1, ID: 97/0x39
Drive port: 2, Channel: 2, ID: 97/0x39
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DPGS00007447BW35
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d8:02
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 11

Drive port: 1, Channel: 2, ID: 65/0x71
Drive port: 2, Channel: 1, ID: 65/0x71
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DQBQ00007447NG5H
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:fb
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 12

Drive port: 1, Channel: 1, ID: 73/0x66

Drive port: 2, Channel: 2, ID: 73/0x66
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E31A00007447BW50
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:d8:a6
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 13

Drive port: 1, Channel: 2, ID: 89/0x4A
Drive port: 2, Channel: 1, ID: 89/0x4A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MY5100007502J0MF
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:7f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 14

Drive port: 1, Channel: 1, ID: 105/0x2D
Drive port: 2, Channel: 2, ID: 105/0x2D
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DND800007447JQE0
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:ce:ab
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 12, Slot 1

Drive port: 1, Channel: 2, ID: 16/0xCD
Drive port: 2, Channel: 1, ID: 16/0xCD
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NENG00007502BENY
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004

World-wide name: 20:00:00:0c:50:45:aa:f7
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 2

Drive port: 1, Channel: 1, ID: 17/0xCC
Drive port: 2, Channel: 2, ID: 17/0xCC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DWKC00007447RB4T
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:3d
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 3

Drive port: 1, Channel: 2, ID: 18/0xCB
Drive port: 2, Channel: 1, ID: 18/0xCB
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DYG300007447AXVT
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:73
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 4

Drive port: 1, Channel: 1, ID: 19/0xCA
Drive port: 2, Channel: 2, ID: 19/0xCA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DKL800007447PXHK
Vendor: IBM-ESXS
Date of manufacture: May 25, 2004
World-wide name: 20:00:00:0c:50:d6:d0:5a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 5

Drive port: 1, Channel: 2, ID: 20/0xC9
Drive port: 2, Channel: 1, ID: 20/0xC9
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E7YA00007447RB0F
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:61
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 6

Drive port: 1, Channel: 1, ID: 21/0xC7
Drive port: 2, Channel: 2, ID: 21/0xC7
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E2BS00007447PXHE
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:d9:de
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 7

Drive port: 1, Channel: 2, ID: 22/0xC6
Drive port: 2, Channel: 1, ID: 22/0xC6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E2CS00007447KUZR
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d8:db
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 8

Drive port: 1, Channel: 1, ID: 23/0xC5
Drive port: 2, Channel: 2, ID: 23/0xC5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MJTS00007502B0UC
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:c9
Drive type: Fibre Channel
Speed: 15015 RPM

Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 9

Drive port: 1, Channel: 2, ID: 82/0x53
Drive port: 2, Channel: 1, ID: 82/0x53
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2JRFH00007502J0VC
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:89
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 10

Drive port: 1, Channel: 1, ID: 98/0x36
Drive port: 2, Channel: 2, ID: 98/0x36
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MMY2000075029S6Y
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:1c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 11

Drive port: 1, Channel: 2, ID: 66/0x6E
Drive port: 2, Channel: 1, ID: 66/0x6E
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2832900007440P56C
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:fd:45
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 12

Drive port: 1, Channel: 1, ID: 74/0x65
Drive port: 2, Channel: 2, ID: 74/0x65
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps

Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E8WZ00007447BVW4
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:fe
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 13

Drive port: 1, Channel: 2, ID: 90/0x49
Drive port: 2, Channel: 1, ID: 90/0x49
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EBZD00007445Q6G6
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:ce:e7
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 14

Drive port: 1, Channel: 1, ID: 106/0x2C
Drive port: 2, Channel: 2, ID: 106/0x2C
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DWG600007447QT5G
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:9c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 1

Drive port: 1, Channel: 2, ID: 24/0xC3
Drive port: 2, Channel: 1, ID: 24/0xC3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E11L00007447NG1Z
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:da:33
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 2
Drive port: 1, Channel: 1, ID: 25/0xBC
Drive port: 2, Channel: 2, ID: 25/0xBC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DAGQ00007447NFNJ
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d8:62
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 3
Drive port: 1, Channel: 2, ID: 26/0xBA
Drive port: 2, Channel: 1, ID: 26/0xBA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NENE00007502AZQQ
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ac:1e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 4
Drive port: 1, Channel: 1, ID: 27/0xB9
Drive port: 2, Channel: 2, ID: 27/0xB9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DPB60000744780M7
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d8:d8
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 5
Drive port: 1, Channel: 2, ID: 28/0xB6
Drive port: 2, Channel: 1, ID: 28/0xB6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2LXLN00007502BE4G

Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:2a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 6
Drive port: 1, Channel: 1, ID: 29/0xB5
Drive port: 2, Channel: 2, ID: 29/0xB5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EBM000007447RBS6
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:db:36
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 7
Drive port: 1, Channel: 2, ID: 30/0xB4
Drive port: 2, Channel: 1, ID: 30/0xB4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EDQS00007447RB9B
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:d9:52
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 8
Drive port: 1, Channel: 1, ID: 31/0xB3
Drive port: 2, Channel: 2, ID: 31/0xB3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E96500007447AY95
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d8:46
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 9
Drive port: 1, Channel: 2, ID: 83/0x52
Drive port: 2, Channel: 1, ID: 83/0x52

Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DZAQ00007447RB3N
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d8:74
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 10
Drive port: 1, Channel: 1, ID: 99/0x35
Drive port: 2, Channel: 2, ID: 99/0x35
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N4ND00007501G32B
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ac:29
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 11
Drive port: 1, Channel: 2, ID: 67/0x6D
Drive port: 2, Channel: 1, ID: 67/0x6D
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MY7500007502AZE7
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:85
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 12
Drive port: 1, Channel: 1, ID: 75/0x63
Drive port: 2, Channel: 2, ID: 75/0x63
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2D9KM00007447QTDL
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:58

Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 13
Drive port: 1, Channel: 2, ID: 91/0x47
Drive port: 2, Channel: 1, ID: 91/0x47
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEZC000075029S7J
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:34
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 14
Drive port: 1, Channel: 1, ID: 107/0x2B
Drive port: 2, Channel: 2, ID: 107/0x2B
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEKK000075029T44
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:a7
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Rack 2

PROFILE FOR STORAGE SUBSYSTEM: RACK2 (3/21/05 3:43:11 PM)

SUMMARY-----

Number of controllers: 2
Number of arrays: 4
Total number of logical drives (includes an access logical drive): 5 of 2048 used
Number of standard logical drives: 4
Number of access logical drives: 1
Number of drives: 112
Supported drive types: Fibre (112)
Total hot spare drives: 0
Standby: 0
In use: 0
Access logical drive: None mapped
Default host type: Windows 2000/Server 2003 Non-Clustered (Host type index 2)
Current configuration
Firmware version: 06.10.06.00
NVS RAM version: N1742F900R910V02
Pending configuration
Staged firmware download supported?: Yes
Firmware version: None

NVSRAM version: None
Transferred on: None
NVSRAM configured for batteries?: Yes
Start cache flushing at (in percentage): 80
Stop cache flushing at (in percentage): 80
Cache block size (in KB): 4
Media scan frequency (in days): Disabled
Failover alert delay (in minutes): 5
Feature enable identifier: 31393137370031393137370041FE3B17
Storage Subsystem worldwide name (ID):
600A0B80001340170000000041B48262

CONTROLLERS-----

Number of controllers: 2

Controller in Slot A

Status: Online
Current configuration
Firmware version: 06.10.06.00
Appware version: 06.10.06.00
Bootware version: 06.10.01.00
NVSRAM version: N1742F900R910V02
Pending configuration
Firmware version: None
Appware version: None
Bootware version: None
NVSRAM version: None
Transferred on: None
Board ID: 5884
Product ID: 1742-900
Product revision: 0520
Serial number: 1T35119155
Date of manufacture: December 29, 2003
Cache/processor size (MB): 1024/128
Date/Time: Mon Mar 21 15:41:56 EST 2005
Associated Logical Drives (* = Preferred Owner):
RACK2_0*, RACK2_1*
Ethernet port: 1
MAC address: 00:a0:b8:13:3e:61
Host name: RACK2_A
Network configuration: Static
IP address: 192.168.128.102
Subnet mask: 255.255.255.0
Gateway: 0.0.0.0
Remote login: Disabled

Drive interface: Fibre
Channel: 1
Current ID: 125/0x1
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Drive interface: Fibre
Channel: 2
Current ID: 125/0x1
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Drive interface: Fibre
Channel: 3
Current ID: 125/0x1
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Drive interface: Fibre

Channel: 4
Current ID: 125/0x1
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up
Host interface: Fibre
Port: 1
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 126/0x0
NL-Port ID: 0x010400
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up
Topology: Fabric Attach
World-wide port name: 20:02:00:a0:b8:13:40:18
World-wide node name: 20:02:00:a0:b8:13:40:17
Part type: HPFC-5400 revision 6
Host interface: Fibre
Port: 2
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 1/0xE8
NL-Port ID: 0x000000
Maximum data rate: 2 Gbps
Current data rate: 1 Gbps
Data rate control: Switch
Link status: Down
Topology: Not available
World-wide port name: 20:02:00:a0:b8:13:40:19
World-wide node name: 20:02:00:a0:b8:13:40:17
Part type: HPFC-5400 revision 6

Controller in Slot B

Status: Online
Current configuration
Firmware version: 06.10.06.00
Appware version: 06.10.06.00
Bootware version: 06.10.01.00
NVSRAM version: N1742F900R910V02
Pending configuration
Firmware version: None
Appware version: None
Bootware version: None
NVSRAM version: None
Transferred on: None
Board ID: 5884
Product ID: 1742-900
Product revision: 0520
Serial number: 1T35119177
Date of manufacture: December 28, 2003
Cache/processor size (MB): 1024/128
Date/Time: Mon Mar 21 15:41:30 EST 2005
Associated Logical Drives (* = Preferred Owner):
RACK2_2*, RACK2_3*
Ethernet port: 1
MAC address: 00:a0:b8:13:40:17
Host name: RACK2_B
Network configuration: Static
IP address: 192.168.128.103
Subnet mask: 255.255.255.0
Gateway: 0.0.0.0
Remote login: Disabled

Drive interface: Fibre
Channel: 1
Current ID: 124/0x2
Maximum data rate: 2 Gbps

Current data rate: 2 Gbps
 Data rate control: Switch
 Link status: Up
 Drive interface: Fibre
 Channel: 2
 Current ID: 124/0x2
 Maximum data rate: 2 Gbps
 Current data rate: 2 Gbps
 Data rate control: Switch
 Link status: Up
 Drive interface: Fibre
 Channel: 3
 Current ID: 124/0x2
 Maximum data rate: 2 Gbps
 Current data rate: 2 Gbps
 Data rate control: Switch
 Link status: Up
 Drive interface: Fibre
 Channel: 4
 Current ID: 124/0x2
 Maximum data rate: 2 Gbps
 Current data rate: 2 Gbps
 Data rate control: Switch
 Link status: Up
 Host interface: Fibre
 Port: 1
 Current ID: Not applicable/0xFFFFFFFF
 Preferred ID: 126/0x0
 NL-Port ID: 0x010500
 Maximum data rate: 2 Gbps
 Current data rate: 2 Gbps
 Data rate control: Switch
 Link status: Up
 Topology: Fabric Attach
 World-wide port name: 20:03:00:a0:b8:13:40:18
 World-wide node name: 20:03:00:a0:b8:13:40:17
 Part type: HPFC-5400 revision 6
 Host interface: Fibre
 Port: 2
 Current ID: Not applicable/0xFFFFFFFF
 Preferred ID: 3/0xE2
 NL-Port ID: 0x000000
 Maximum data rate: 2 Gbps
 Current data rate: 1 Gbps
 Data rate control: Switch
 Link status: Down
 Topology: Not available
 World-wide port name: 20:03:00:a0:b8:13:40:19
 World-wide node name: 20:03:00:a0:b8:13:40:17
 Part type: HPFC-5400 revision 6

ARRAYS-----

Number of arrays: 4

Array 1 (RAID 0)
 Status: Online
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Current owner: Controller in slot A
 Associated logical drives and free capacities:
 RACK2_0 (935.238 GB)
 Associated drives (in piece order):
 Drive at Enclosure 0, Slot 1
 Drive at Enclosure 0, Slot 2
 Drive at Enclosure 0, Slot 3
 Drive at Enclosure 0, Slot 4
 Drive at Enclosure 0, Slot 5
 Drive at Enclosure 0, Slot 6

Drive at Enclosure 0, Slot 7
 Drive at Enclosure 0, Slot 8
 Drive at Enclosure 0, Slot 9
 Drive at Enclosure 0, Slot 10
 Drive at Enclosure 0, Slot 11
 Drive at Enclosure 0, Slot 12
 Drive at Enclosure 0, Slot 13
 Drive at Enclosure 0, Slot 14
 Drive at Enclosure 1, Slot 1
 Drive at Enclosure 1, Slot 2
 Drive at Enclosure 1, Slot 3
 Drive at Enclosure 1, Slot 4
 Drive at Enclosure 1, Slot 5
 Drive at Enclosure 1, Slot 6
 Drive at Enclosure 1, Slot 7
 Drive at Enclosure 1, Slot 8
 Drive at Enclosure 1, Slot 9
 Drive at Enclosure 1, Slot 10
 Drive at Enclosure 1, Slot 11
 Drive at Enclosure 1, Slot 12
 Drive at Enclosure 1, Slot 13
 Drive at Enclosure 1, Slot 14
 Array 2 (RAID 0)
 Status: Online
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Current owner: Controller in slot A
 Associated logical drives and free capacities:
 RACK2_1 (935.238 GB)
 Associated drives (in piece order):
 Drive at Enclosure 2, Slot 1
 Drive at Enclosure 2, Slot 2
 Drive at Enclosure 2, Slot 3
 Drive at Enclosure 2, Slot 4
 Drive at Enclosure 2, Slot 5
 Drive at Enclosure 2, Slot 6
 Drive at Enclosure 2, Slot 7
 Drive at Enclosure 2, Slot 8
 Drive at Enclosure 2, Slot 9
 Drive at Enclosure 2, Slot 10
 Drive at Enclosure 2, Slot 11
 Drive at Enclosure 2, Slot 12
 Drive at Enclosure 2, Slot 13
 Drive at Enclosure 2, Slot 14
 Drive at Enclosure 3, Slot 1
 Drive at Enclosure 3, Slot 2
 Drive at Enclosure 3, Slot 3
 Drive at Enclosure 3, Slot 4
 Drive at Enclosure 3, Slot 5
 Drive at Enclosure 3, Slot 6
 Drive at Enclosure 3, Slot 7
 Drive at Enclosure 3, Slot 8
 Drive at Enclosure 3, Slot 9
 Drive at Enclosure 3, Slot 10
 Drive at Enclosure 3, Slot 11
 Drive at Enclosure 3, Slot 12
 Drive at Enclosure 3, Slot 13
 Drive at Enclosure 3, Slot 14
 Array 3 (RAID 0)
 Status: Online
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Current owner: Controller in slot B
 Associated logical drives and free capacities:
 RACK2_2 (935.238 GB)
 Associated drives (in piece order):
 Drive at Enclosure 10, Slot 1
 Drive at Enclosure 10, Slot 2

Drive at Enclosure 10, Slot 3
 Drive at Enclosure 10, Slot 4
 Drive at Enclosure 10, Slot 5
 Drive at Enclosure 10, Slot 6
 Drive at Enclosure 10, Slot 7
 Drive at Enclosure 10, Slot 8
 Drive at Enclosure 10, Slot 9
 Drive at Enclosure 10, Slot 10
 Drive at Enclosure 10, Slot 11
 Drive at Enclosure 10, Slot 12
 Drive at Enclosure 10, Slot 13
 Drive at Enclosure 10, Slot 14
 Drive at Enclosure 11, Slot 1
 Drive at Enclosure 11, Slot 2
 Drive at Enclosure 11, Slot 3
 Drive at Enclosure 11, Slot 4
 Drive at Enclosure 11, Slot 5
 Drive at Enclosure 11, Slot 6
 Drive at Enclosure 11, Slot 7
 Drive at Enclosure 11, Slot 8
 Drive at Enclosure 11, Slot 9
 Drive at Enclosure 11, Slot 10
 Drive at Enclosure 11, Slot 11
 Drive at Enclosure 11, Slot 12
 Drive at Enclosure 11, Slot 13
 Drive at Enclosure 11, Slot 14

Array 4 (RAID 0)

Status: Online
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Current owner: Controller in slot B
 Associated logical drives and free capacities:
 RACK2_3 (935.238 GB)

Associated drives (in piece order):

Drive at Enclosure 12, Slot 1
 Drive at Enclosure 12, Slot 2
 Drive at Enclosure 12, Slot 3
 Drive at Enclosure 12, Slot 4
 Drive at Enclosure 12, Slot 5
 Drive at Enclosure 12, Slot 6
 Drive at Enclosure 12, Slot 7
 Drive at Enclosure 12, Slot 8
 Drive at Enclosure 12, Slot 9
 Drive at Enclosure 12, Slot 10
 Drive at Enclosure 12, Slot 11
 Drive at Enclosure 12, Slot 12
 Drive at Enclosure 12, Slot 13
 Drive at Enclosure 12, Slot 14
 Drive at Enclosure 13, Slot 1
 Drive at Enclosure 13, Slot 2
 Drive at Enclosure 13, Slot 3
 Drive at Enclosure 13, Slot 4
 Drive at Enclosure 13, Slot 5
 Drive at Enclosure 13, Slot 6
 Drive at Enclosure 13, Slot 7
 Drive at Enclosure 13, Slot 8
 Drive at Enclosure 13, Slot 9
 Drive at Enclosure 13, Slot 10
 Drive at Enclosure 13, Slot 11
 Drive at Enclosure 13, Slot 12
 Drive at Enclosure 13, Slot 13
 Drive at Enclosure 13, Slot 14

STANDARD LOGICAL DRIVES-----

SUMMARY

Number of standard logical drives: 4
 See other Logical Drives sub-tabs for premium feature information.

| NAME | STATUS | CAPACITY | RAID LEVEL | ARRAY |
|---------|---------|------------|------------|-------|
| RACK2_0 | Optimal | 935.238 GB | 0 | 1 |
| RACK2_1 | Optimal | 935.238 GB | 0 | 2 |
| RACK2_2 | Optimal | 935.238 GB | 0 | 3 |
| RACK2_3 | Optimal | 935.238 GB | 0 | 4 |

DETAILS

Logical Drive name: RACK2_0
 Logical Drive ID: 60:0a:0b:80:00:13:40:17:00:00:00:41:fe:02:84
 Subsystem ID (SSID): 0
 Status: Optimal
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Preferred owner: Controller in slot A
 Current owner: Controller in slot A
 Capacity: 935.238 GB
 RAID level: 0
 Segment size: 64 KB
 Modification priority: High
 Associated array: 1
 Read cache: Disabled
 Write cache: Disabled
 Write cache without batteries: Disabled
 Write cache with mirroring: Disabled
 Flush write cache after (in seconds): 10.00
 Cache read ahead multiplier: 0
 Enable background media scan: Disabled
 Media scan with redundancy check: Disabled

Logical Drive name: RACK2_1
 Logical Drive ID: 60:0a:0b:80:00:13:40:17:00:00:00:05:41:bd:55:44
 Subsystem ID (SSID): 1
 Status: Optimal
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Preferred owner: Controller in slot A
 Current owner: Controller in slot A
 Capacity: 935.238 GB
 RAID level: 0
 Segment size: 64 KB
 Modification priority: Highest
 Associated array: 2
 Read cache: Disabled
 Write cache: Disabled
 Write cache without batteries: Disabled
 Write cache with mirroring: Disabled
 Flush write cache after (in seconds): 10.00
 Cache read ahead multiplier: 0
 Enable background media scan: Disabled
 Media scan with redundancy check: Disabled

Logical Drive name: RACK2_2
 Logical Drive ID: 60:0a:0b:80:00:13:3e:61:00:00:00:01:41:bd:55:d1
 Subsystem ID (SSID): 2
 Status: Optimal
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Preferred owner: Controller in slot B
 Current owner: Controller in slot B
 Capacity: 935.238 GB
 RAID level: 0
 Segment size: 64 KB
 Modification priority: Highest
 Associated array: 3
 Read cache: Disabled
 Write cache: Disabled
 Write cache without batteries: Disabled

Write cache with mirroring: Disabled
 Flush write cache after (in seconds): 10.00
 Cache read ahead multiplier: 0
 Enable background media scan: Disabled
 Media scan with redundancy check: Disabled

Logical Drive name: RACK2_3
 Logical Drive ID: 60:0a:0b:80:00:13:3e:61:00:00:00:03:41:bd:56:03
 Subsystem ID (SSID): 3
 Status: Optimal
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Preferred owner: Controller in slot B
 Current owner: Controller in slot B
 Capacity: 935.238 GB
 RAID level: 0
 Segment size: 64 KB
 Modification priority: Highest
 Associated array: 4
 Read cache: Disabled
 Write cache: Disabled
 Write cache without batteries: Disabled
 Write cache with mirroring: Disabled
 Flush write cache after (in seconds): 10.00
 Cache read ahead multiplier: 0
 Enable background media scan: Disabled
 Media scan with redundancy check: Disabled

MISSING LOGICAL DRIVES-----

Number of missing logical drives: 0

DRIVES-----

SUMMARY

Number of drives: 112
 Supported drive types: Fibre (112)

BASIC:

| TRAY | SLOT | STATUS | CAPACITY | CURRENT DATA RATE |
|------------|----------|-----------|----------|-------------------|
| PRODUCT ID | FIRMWARE | VERSION | | |
| 0, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 0, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |

| | | | | |
|-------|---------|-----------|--------|-------------------|
| 0, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 1, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 2, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 3, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 3, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 3, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 3, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |
| 3, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC F B954 |

| | | | | | | |
|--------|---------|-----------|--------|------------|---|------|
| 3, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 3, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 3, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 3, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 3, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 3, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 3, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 3, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 3, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |

| | | | | | | |
|--------|---------|-----------|--------|------------|---|------|
| 11, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |

DRIVE CHANNELS:

| | | |
|------------|-------------------|-------------------|
| TRAY, SLOT | PREFERRED CHANNEL | REDUNDANT CHANNEL |
| 0, 1 | 4 | 3 |
| 0, 2 | 3 | 4 |

| | | |
|--------|---|---|
| 0, 3 | 4 | 3 |
| 0, 4 | 3 | 4 |
| 0, 5 | 4 | 3 |
| 0, 6 | 3 | 4 |
| 0, 7 | 4 | 3 |
| 0, 8 | 3 | 4 |
| 0, 9 | 4 | 3 |
| 0, 10 | 3 | 4 |
| 0, 11 | 4 | 3 |
| 0, 12 | 3 | 4 |
| 0, 13 | 4 | 3 |
| 0, 14 | 3 | 4 |
| 1, 1 | 4 | 3 |
| 1, 2 | 3 | 4 |
| 1, 3 | 4 | 3 |
| 1, 4 | 3 | 4 |
| 1, 5 | 4 | 3 |
| 1, 6 | 3 | 4 |
| 1, 7 | 4 | 3 |
| 1, 8 | 3 | 4 |
| 1, 9 | 4 | 3 |
| 1, 10 | 3 | 4 |
| 1, 11 | 4 | 3 |
| 1, 12 | 3 | 4 |
| 1, 13 | 4 | 3 |
| 1, 14 | 3 | 4 |
| 2, 1 | 4 | 3 |
| 2, 2 | 3 | 4 |
| 2, 3 | 4 | 3 |
| 2, 4 | 3 | 4 |
| 2, 5 | 4 | 3 |
| 2, 6 | 3 | 4 |
| 2, 7 | 4 | 3 |
| 2, 8 | 3 | 4 |
| 2, 9 | 4 | 3 |
| 2, 10 | 3 | 4 |
| 2, 11 | 4 | 3 |
| 2, 12 | 3 | 4 |
| 2, 13 | 4 | 3 |
| 2, 14 | 3 | 4 |
| 3, 1 | 4 | 3 |
| 3, 2 | 3 | 4 |
| 3, 3 | 4 | 3 |
| 3, 4 | 3 | 4 |
| 3, 5 | 4 | 3 |
| 3, 6 | 3 | 4 |
| 3, 7 | 4 | 3 |
| 3, 8 | 3 | 4 |
| 3, 9 | 4 | 3 |
| 3, 10 | 3 | 4 |
| 3, 11 | 4 | 3 |
| 3, 12 | 3 | 4 |
| 3, 13 | 4 | 3 |
| 3, 14 | 3 | 4 |
| 10, 1 | 2 | 1 |
| 10, 2 | 1 | 2 |
| 10, 3 | 2 | 1 |
| 10, 4 | 1 | 2 |
| 10, 5 | 2 | 1 |
| 10, 6 | 1 | 2 |
| 10, 7 | 2 | 1 |
| 10, 8 | 1 | 2 |
| 10, 9 | 2 | 1 |
| 10, 10 | 1 | 2 |
| 10, 11 | 2 | 1 |
| 10, 12 | 1 | 2 |
| 10, 13 | 2 | 1 |
| 10, 14 | 1 | 2 |

| | | |
|--------|---|---|
| 11, 1 | 2 | 1 |
| 11, 2 | 1 | 2 |
| 11, 3 | 2 | 1 |
| 11, 4 | 1 | 2 |
| 11, 5 | 2 | 1 |
| 11, 6 | 1 | 2 |
| 11, 7 | 2 | 1 |
| 11, 8 | 1 | 2 |
| 11, 9 | 2 | 1 |
| 11, 10 | 1 | 2 |
| 11, 11 | 2 | 1 |
| 11, 12 | 1 | 2 |
| 11, 13 | 2 | 1 |
| 11, 14 | 1 | 2 |
| 12, 1 | 2 | 1 |
| 12, 2 | 1 | 2 |
| 12, 3 | 2 | 1 |
| 12, 4 | 1 | 2 |
| 12, 5 | 2 | 1 |
| 12, 6 | 1 | 2 |
| 12, 7 | 2 | 1 |
| 12, 8 | 1 | 2 |
| 12, 9 | 2 | 1 |
| 12, 10 | 1 | 2 |
| 12, 11 | 2 | 1 |
| 12, 12 | 1 | 2 |
| 12, 13 | 2 | 1 |
| 12, 14 | 1 | 2 |
| 13, 1 | 2 | 1 |
| 13, 2 | 1 | 2 |
| 13, 3 | 2 | 1 |
| 13, 4 | 1 | 2 |
| 13, 5 | 2 | 1 |
| 13, 6 | 1 | 2 |
| 13, 7 | 2 | 1 |
| 13, 8 | 1 | 2 |
| 13, 9 | 2 | 1 |
| 13, 10 | 1 | 2 |
| 13, 11 | 2 | 1 |
| 13, 12 | 1 | 2 |
| 13, 13 | 2 | 1 |
| 13, 14 | 1 | 2 |

HOT SPARE COVERAGE:

The following arrays are not protected: 1, 3, 4, 2

Total hot spare drives: 0

Standby: 0

In use: 0

DETAILS

Drive at Enclosure 0, Slot 1

Drive port: 1, Channel: 4, ID: 0/0xEF

Drive port: 2, Channel: 3, ID: 0/0xEF

Drive path redundancy: OK

Status: Optimal

Raw capacity: 33.902 GB

Usable capacity: 33.402 GB

Current data rate: 2 Gbps

Product ID: ST336753FC F

Firmware version: B954

Serial number: 3HX2MY9X00007502BDZ0

Vendor: IBM-ESXS

Date of manufacture: July 20, 2004

World-wide name: 20:00:00:0c:50:45:ab:3b

Drive type: Fibre Channel

Speed: 15015 RPM

Mode: Assigned

Associated array: 1

Drive at Enclosure 0, Slot 2

Drive port: 1, Channel: 3, ID: 1/0xE8
Drive port: 2, Channel: 4, ID: 1/0xE8
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MA2Z00007502CX0Q
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a8:e9
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 3

Drive port: 1, Channel: 4, ID: 2/0xE4
Drive port: 2, Channel: 3, ID: 2/0xE4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N4SF00007502BTJF
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:6f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 4

Drive port: 1, Channel: 3, ID: 3/0xE2
Drive port: 2, Channel: 4, ID: 3/0xE4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEQP00007502BU2L
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:ab
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 5

Drive port: 1, Channel: 4, ID: 4/0xE1
Drive port: 2, Channel: 3, ID: 4/0xE1
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F

Firmware version: B954
Serial number: 3HX2NEJW000075029SKF
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:da
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 6

Drive port: 1, Channel: 3, ID: 5/0xE0
Drive port: 2, Channel: 4, ID: 5/0xE0
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E1RV00007447BW2F
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:d9:9e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 7

Drive port: 1, Channel: 4, ID: 6/0xDC
Drive port: 2, Channel: 3, ID: 6/0xDC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E9JB00007447RB4B
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d9:54
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 8

Drive port: 1, Channel: 3, ID: 7/0xDA
Drive port: 2, Channel: 4, ID: 7/0xDA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2ED8C00007447BW2D
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d7:b3
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 9

Drive port: 1, Channel: 4, ID: 80/0x55
Drive port: 2, Channel: 3, ID: 80/0x55
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E2F600007447RBJB
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:db:1f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 10
Drive port: 1, Channel: 3, ID: 96/0x3A
Drive port: 2, Channel: 4, ID: 96/0x3A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2M8JX00007502CWMZ
Vendor: IBM-ESXS
Date of manufacture: July 22, 2004
World-wide name: 20:00:00:0c:50:45:b8:68
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 11
Drive port: 1, Channel: 4, ID: 64/0x72
Drive port: 2, Channel: 3, ID: 64/0x72
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EAKC00007447RBP5
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:c1
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 12
Drive port: 1, Channel: 3, ID: 72/0x67
Drive port: 2, Channel: 4, ID: 72/0x67
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EEGY00007447RB4C
Vendor: IBM-ESXS

Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:d0:09
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 13
Drive port: 1, Channel: 4, ID: 88/0x4B
Drive port: 2, Channel: 3, ID: 88/0x4B
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DWS900007447QT7M
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:1d
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 14
Drive port: 1, Channel: 3, ID: 104/0x2E
Drive port: 2, Channel: 4, ID: 104/0x2E
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EAM400007447AY35
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d8:3e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 1
Drive port: 1, Channel: 4, ID: 8/0xD9
Drive port: 2, Channel: 3, ID: 8/0xD9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EEE00000744780V7
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d1:50
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 2
Drive port: 1, Channel: 3, ID: 9/0xD6
Drive port: 2, Channel: 4, ID: 9/0xD6
Drive path redundancy: OK

Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EC0900007447AYC9
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:e3
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 3
Drive port: 1, Channel: 4, ID: 10/0xD5
Drive port: 2, Channel: 3, ID: 10/0xD5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX27B7S00007440DNUK
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:fc:da
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 4
Drive port: 1, Channel: 3, ID: 11/0xD4
Drive port: 2, Channel: 4, ID: 11/0xD4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX27NAT00007440DPAN
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:fd:cd
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 5
Drive port: 1, Channel: 4, ID: 12/0xD3
Drive port: 2, Channel: 3, ID: 12/0xD3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DWS600007447PX1X
Vendor: IBM-ESXS
Date of manufacture: May 25, 2004
World-wide name: 20:00:00:0c:50:d6:ac:eb
Drive type: Fibre Channel

Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 6
Drive port: 1, Channel: 3, ID: 13/0xD2
Drive port: 2, Channel: 4, ID: 13/0xD2
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EC0600007447AYCV
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:df
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 7
Drive port: 1, Channel: 4, ID: 14/0xD1
Drive port: 2, Channel: 3, ID: 14/0xD1
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N0Z000007502AZC9
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:17
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 8
Drive port: 1, Channel: 3, ID: 15/0xCE
Drive port: 2, Channel: 4, ID: 15/0xCE
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DDAV00007447NFCN
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d0:3d
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 9
Drive port: 1, Channel: 4, ID: 81/0x54
Drive port: 2, Channel: 3, ID: 81/0x54
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB

Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DMTB000074478XAF
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d0:53
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 10
Drive port: 1, Channel: 3, ID: 97/0x39
Drive port: 2, Channel: 4, ID: 97/0x39
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DXN600007447KUQ0
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:95
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 11
Drive port: 1, Channel: 4, ID: 65/0x71
Drive port: 2, Channel: 3, ID: 65/0x71
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MRKW00007502AZ51
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:ca
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 12
Drive port: 1, Channel: 3, ID: 73/0x66
Drive port: 2, Channel: 4, ID: 73/0x66
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N4RR00007502CWF7
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:9e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 13
Drive port: 1, Channel: 4, ID: 89/0x4A
Drive port: 2, Channel: 3, ID: 89/0x4A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DD730000744780WR
Vendor: IBM-ESXS
Date of manufacture: May 25, 2004
World-wide name: 20:00:00:0c:50:d6:d0:83
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 14
Drive port: 1, Channel: 3, ID: 105/0x2D
Drive port: 2, Channel: 4, ID: 105/0x2D
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DJE200007447PXHG
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d0:57
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 2, Slot 1
Drive port: 1, Channel: 4, ID: 16/0xCD
Drive port: 2, Channel: 3, ID: 16/0xCD
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DXBS00007447AYBZ
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:f9
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 2
Drive port: 1, Channel: 3, ID: 17/0xCC
Drive port: 2, Channel: 4, ID: 17/0xCC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954

Serial number: 3HX2EEAS00007447AYFB
Vendor: IBM-ESXS
Date of manufacture: May 25, 2004
World-wide name: 20:00:00:0c:50:d6:cf:eb
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 3

Drive port: 1, Channel: 4, ID: 18/0xCB
Drive port: 2, Channel: 3, ID: 18/0xCB
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DYMN00007447NG3Q
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d0:3f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 4

Drive port: 1, Channel: 3, ID: 19/0xCA
Drive port: 2, Channel: 4, ID: 19/0xCA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX281EC00007440BYFC
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:fd:55
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 5

Drive port: 1, Channel: 4, ID: 20/0xC9
Drive port: 2, Channel: 3, ID: 20/0xC9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX282PP00007440DNLE
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:fd:3f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 6

Drive port: 1, Channel: 3, ID: 21/0xC7

Drive port: 2, Channel: 4, ID: 21/0xC7
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EG7Y000074478XM2
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:f2:8c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 7

Drive port: 1, Channel: 4, ID: 22/0xC6
Drive port: 2, Channel: 3, ID: 22/0xC6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MS0400007502B0NV
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:0a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 8

Drive port: 1, Channel: 3, ID: 23/0xC5
Drive port: 2, Channel: 4, ID: 23/0xC5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N2NL00007502AZWM
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:bb
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 9

Drive port: 1, Channel: 4, ID: 82/0x53
Drive port: 2, Channel: 3, ID: 82/0x53
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MSBC00007502AZCW
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004

World-wide name: 20:00:00:0c:50:45:a9:71
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 10

Drive port: 1, Channel: 3, ID: 98/0x36
Drive port: 2, Channel: 4, ID: 98/0x36
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MGMW00007502CJ9N
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a8:38
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 11

Drive port: 1, Channel: 4, ID: 66/0x6E
Drive port: 2, Channel: 3, ID: 66/0x6E
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NF2J00007502A7RA
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:75
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 12

Drive port: 1, Channel: 3, ID: 74/0x65
Drive port: 2, Channel: 4, ID: 74/0x65
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DNEH00007447AYFJ
Vendor: IBM-ESXS
Date of manufacture: May 25, 2004
World-wide name: 20:00:00:0c:50:d6:cf:dc
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 13

Drive port: 1, Channel: 4, ID: 90/0x49
Drive port: 2, Channel: 3, ID: 90/0x49
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DN4100007447LWHN
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:ce:9d
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 14

Drive port: 1, Channel: 3, ID: 106/0x2C
Drive port: 2, Channel: 4, ID: 106/0x2C
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DTHN00007447RB7R
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:40
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 1

Drive port: 1, Channel: 4, ID: 24/0xC3
Drive port: 2, Channel: 3, ID: 24/0xC3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX27MP700007440BF4F
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:fd:7e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 2

Drive port: 1, Channel: 3, ID: 25/0xBC
Drive port: 2, Channel: 4, ID: 25/0xBC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX27CBH00007440CCH1
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:fd:81
Drive type: Fibre Channel
Speed: 15015 RPM

Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 3

Drive port: 1, Channel: 4, ID: 26/0xBA
Drive port: 2, Channel: 3, ID: 26/0xBA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX283PR00007440DLWA
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:fd:20
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 4

Drive port: 1, Channel: 3, ID: 27/0xB9
Drive port: 2, Channel: 4, ID: 27/0xB9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N4BV00007502BERV
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a7:d1
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 5

Drive port: 1, Channel: 4, ID: 28/0xB6
Drive port: 2, Channel: 3, ID: 28/0xB6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NESE00007502AZS3
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:60
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 6

Drive port: 1, Channel: 3, ID: 29/0xB5
Drive port: 2, Channel: 4, ID: 29/0xB5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps

Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2760N00007440P665
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:fd:75
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 7

Drive port: 1, Channel: 4, ID: 30/0xB4
Drive port: 2, Channel: 3, ID: 30/0xB4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX27MY900007440AT0U
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:fd:a2
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 8

Drive port: 1, Channel: 3, ID: 31/0xB3
Drive port: 2, Channel: 4, ID: 31/0xB3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX27ML000007440P64G
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:fd:76
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 9

Drive port: 1, Channel: 4, ID: 83/0x52
Drive port: 2, Channel: 3, ID: 83/0x52
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2D5BF00007441HJ9M
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:ce:b8
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 10
Drive port: 1, Channel: 3, ID: 99/0x35
Drive port: 2, Channel: 4, ID: 99/0x35
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DY6Z00007447NFA4
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d0:a5
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 11
Drive port: 1, Channel: 4, ID: 67/0x6D
Drive port: 2, Channel: 3, ID: 67/0x6D
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E1RC00007447RBST
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d0:3a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 12
Drive port: 1, Channel: 3, ID: 75/0x63
Drive port: 2, Channel: 4, ID: 75/0x63
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EAND00007442LPAD
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:a2
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 13
Drive port: 1, Channel: 4, ID: 91/0x47
Drive port: 2, Channel: 3, ID: 91/0x47
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EE5W000074478X91

Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d0:49
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 14
Drive port: 1, Channel: 3, ID: 107/0x2B
Drive port: 2, Channel: 4, ID: 107/0x2B
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX262SF00007440BFL4
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:fd:b4
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 10, Slot 1
Drive port: 1, Channel: 2, ID: 0/0xEF
Drive port: 2, Channel: 1, ID: 0/0xEF
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EA8P00007447RBKV
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:db:44
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 2
Drive port: 1, Channel: 1, ID: 1/0xE8
Drive port: 2, Channel: 2, ID: 1/0xE8
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N4WV000075029SYK
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:d6:ac:18
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 3
Drive port: 1, Channel: 2, ID: 2/0xE4
Drive port: 2, Channel: 1, ID: 2/0xE4

Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N4Y000007502CJ7G
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a8:3c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 4
Drive port: 1, Channel: 1, ID: 3/0xE2
Drive port: 2, Channel: 2, ID: 3/0xE2
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MVW9000075029SK7
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:33
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 5
Drive port: 1, Channel: 2, ID: 4/0xE1
Drive port: 2, Channel: 1, ID: 4/0xE1
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MPLW00007502BESD
Vendor: IBM-ESXS
Date of manufacture: July 22, 2004
World-wide name: 20:00:00:0c:50:45:b8:5b
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 6
Drive port: 1, Channel: 1, ID: 5/0xE0
Drive port: 2, Channel: 2, ID: 5/0xE0
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2D91K00007447RBJU
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:f3

Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 7
Drive port: 1, Channel: 2, ID: 6/0xDC
Drive port: 2, Channel: 1, ID: 6/0xDC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEPY00007502BE1L
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:43
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 8
Drive port: 1, Channel: 1, ID: 7/0xDA
Drive port: 2, Channel: 2, ID: 7/0xDA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N39600007502BTPM
Vendor: IBM-ESXS
Date of manufacture: July 22, 2004
World-wide name: 20:00:00:0c:50:45:b9:40
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 9
Drive port: 1, Channel: 2, ID: 8/0x55
Drive port: 2, Channel: 1, ID: 8/0x55
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2KS7A00007502CJ0S
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a9:64
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 10
Drive port: 1, Channel: 1, ID: 96/0x3A
Drive port: 2, Channel: 2, ID: 96/0x3A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB

Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E9KN00007447NFH4
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:74
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 11
Drive port: 1, Channel: 2, ID: 64/0x72
Drive port: 2, Channel: 1, ID: 64/0x72
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N1GG000075029T0P
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ab:f1
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 12
Drive port: 1, Channel: 1, ID: 72/0x67
Drive port: 2, Channel: 2, ID: 72/0x67
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MF8400007502CWD5
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a7:d5
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 13
Drive port: 1, Channel: 2, ID: 88/0x4B
Drive port: 2, Channel: 1, ID: 88/0x4B
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEWF00007502BEWP
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:76
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned

Associated array: 3

Drive at Enclosure 10, Slot 14
Drive port: 1, Channel: 1, ID: 104/0x2E
Drive port: 2, Channel: 2, ID: 104/0x2E
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEHX000075029T3U
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:2a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 1
Drive port: 1, Channel: 2, ID: 8/0xD9
Drive port: 2, Channel: 1, ID: 8/0xD9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEQF000075029SD6
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:09
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 2
Drive port: 1, Channel: 1, ID: 9/0xD6
Drive port: 2, Channel: 2, ID: 9/0xD6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DEKT00007447NFXE
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:2a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 3
Drive port: 1, Channel: 2, ID: 10/0xD5
Drive port: 2, Channel: 1, ID: 10/0xD5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F

Firmware version: B954
Serial number: 3HX2N55700007502AZ2V
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:7d
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 4

Drive port: 1, Channel: 1, ID: 11/0xD4
Drive port: 2, Channel: 2, ID: 11/0xD4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps

Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DKQC00007443EZ5G
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:6c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 5

Drive port: 1, Channel: 2, ID: 12/0xD3
Drive port: 2, Channel: 1, ID: 12/0xD3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps

Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N4FV00007502BFB8
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:e3
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 6

Drive port: 1, Channel: 1, ID: 13/0xD2
Drive port: 2, Channel: 2, ID: 13/0xD2
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps

Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2M2C500007502AZW9
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ab:5f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 7

Drive port: 1, Channel: 2, ID: 14/0xD1
Drive port: 2, Channel: 1, ID: 14/0xD1
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E3AY00007447LWHH
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d8:e3
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 8

Drive port: 1, Channel: 1, ID: 15/0xCE
Drive port: 2, Channel: 2, ID: 15/0xCE
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F

Firmware version: B954
Serial number: 3HX2EE7400007447BVSF
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:c8
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 9

Drive port: 1, Channel: 2, ID: 81/0x54
Drive port: 2, Channel: 1, ID: 81/0x54
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F

Firmware version: B954
Serial number: 3HX2DMTF00007447QT8E
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:ad
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 10

Drive port: 1, Channel: 1, ID: 97/0x39
Drive port: 2, Channel: 2, ID: 97/0x39
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F

Firmware version: B954
Serial number: 3HX2M64N00007502CVVD
Vendor: IBM-ESXS

Date of manufacture: July 22, 2004
World-wide name: 20:00:00:0c:50:45:b8:24
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 11

Drive port: 1, Channel: 2, ID: 65/0x71
Drive port: 2, Channel: 1, ID: 65/0x71
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E9Q9000074464197
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:0c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 12

Drive port: 1, Channel: 1, ID: 73/0x66
Drive port: 2, Channel: 2, ID: 73/0x66
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DCVQ00007447KV8E
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:da:2c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 13

Drive port: 1, Channel: 2, ID: 89/0x4A
Drive port: 2, Channel: 1, ID: 89/0x4A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EAHV00007447RBJF
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:f5
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 14

Drive port: 1, Channel: 1, ID: 105/0x2D
Drive port: 2, Channel: 2, ID: 105/0x2D
Drive path redundancy: OK

Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NF1Y00007502J0YA
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:de
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 12, Slot 1

Drive port: 1, Channel: 2, ID: 16/0xCD
Drive port: 2, Channel: 1, ID: 16/0xCD
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2ME0C00007502BTJH
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:69
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 2

Drive port: 1, Channel: 1, ID: 17/0xCC
Drive port: 2, Channel: 2, ID: 17/0xCC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MEDG00007502BEHY
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:cd
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 3

Drive port: 1, Channel: 2, ID: 18/0xCB
Drive port: 2, Channel: 1, ID: 18/0xCB
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MY8600007502CWM2
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:aa:eb
Drive type: Fibre Channel

Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 4

Drive port: 1, Channel: 1, ID: 19/0xCA
Drive port: 2, Channel: 2, ID: 19/0xCA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MY53000075029RG9
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:aa:95
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 5

Drive port: 1, Channel: 2, ID: 20/0xC9
Drive port: 2, Channel: 1, ID: 20/0xC9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MWH800007502BE66
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ac:08
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 6

Drive port: 1, Channel: 1, ID: 21/0xC7
Drive port: 2, Channel: 2, ID: 21/0xC7
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DVQN00007447RB0L
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d8:cc
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 7

Drive port: 1, Channel: 2, ID: 22/0xC6
Drive port: 2, Channel: 1, ID: 22/0xC6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB

Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DNCQ00007447AYH5
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:db
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 8

Drive port: 1, Channel: 1, ID: 23/0xC5
Drive port: 2, Channel: 2, ID: 23/0xC5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EAE800007447RBPY
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:eb
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 9

Drive port: 1, Channel: 2, ID: 82/0x53
Drive port: 2, Channel: 1, ID: 82/0x53
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MY8F00007502A7RV
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:aa:aa
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 10

Drive port: 1, Channel: 1, ID: 98/0x36
Drive port: 2, Channel: 2, ID: 98/0x36
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MDYX00007502BTGD
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:74
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 11
Drive port: 1, Channel: 2, ID: 66/0x6E
Drive port: 2, Channel: 1, ID: 66/0x6E
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E1MM0000744781ZV
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d8:68
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 12
Drive port: 1, Channel: 1, ID: 74/0x65
Drive port: 2, Channel: 2, ID: 74/0x65
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NET300007502A7UQ
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:1a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 13
Drive port: 1, Channel: 2, ID: 90/0x49
Drive port: 2, Channel: 1, ID: 90/0x49
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2KZKW00007453EM8S
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ac:d7
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 14
Drive port: 1, Channel: 1, ID: 106/0x2C
Drive port: 2, Channel: 2, ID: 106/0x2C
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954

Serial number: 3HX2DPBZ00007447QTDP
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:bf
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 1
Drive port: 1, Channel: 2, ID: 24/0xC3
Drive port: 2, Channel: 1, ID: 24/0xC3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DSWX0000744780X5
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d8:d8
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 2
Drive port: 1, Channel: 1, ID: 25/0xBC
Drive port: 2, Channel: 2, ID: 25/0xBC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N1H200007502CWE2
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:aa:89
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 3
Drive port: 1, Channel: 2, ID: 26/0xBA
Drive port: 2, Channel: 1, ID: 26/0xBA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EC0Y000074411AR3
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d1:4e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 4
Drive port: 1, Channel: 1, ID: 27/0xB9

Drive port: 2, Channel: 2, ID: 27/0xB9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2CX6B00007447AYG8
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:aa
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 5

Drive port: 1, Channel: 2, ID: 28/0xB6
Drive port: 2, Channel: 1, ID: 28/0xB6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EB3D00007447NFK7
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:76
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 6

Drive port: 1, Channel: 1, ID: 29/0xB5
Drive port: 2, Channel: 2, ID: 29/0xB5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DZKD00007447829E
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:da:6a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 7

Drive port: 1, Channel: 2, ID: 30/0xB4
Drive port: 2, Channel: 1, ID: 30/0xB4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EADW00007447RAZW
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004

World-wide name: 20:00:00:0c:50:d6:da:be
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 8

Drive port: 1, Channel: 1, ID: 31/0xB3
Drive port: 2, Channel: 2, ID: 31/0xB3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DP7E00007447NFLF
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:b2:08
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 9

Drive port: 1, Channel: 2, ID: 83/0x52
Drive port: 2, Channel: 1, ID: 83/0x52
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EDZM00007447BW5X
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:4c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 10

Drive port: 1, Channel: 1, ID: 99/0x35
Drive port: 2, Channel: 2, ID: 99/0x35
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NDWP00007502E1AH
Vendor: IBM-ESXS
Date of manufacture: July 22, 2004
World-wide name: 20:00:00:0c:50:45:b9:76
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 11

Drive port: 1, Channel: 2, ID: 67/0x6D
Drive port: 2, Channel: 1, ID: 67/0x6D
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MY9H00007502BECN
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:65
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 12
Drive port: 1, Channel: 1, ID: 75/0x63
Drive port: 2, Channel: 2, ID: 75/0x63
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MPTD00007502CJ8B
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a8:36
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 13
Drive port: 1, Channel: 2, ID: 91/0x47
Drive port: 2, Channel: 1, ID: 91/0x47
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2ME4800007502BT7R
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:9c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 14
Drive port: 1, Channel: 1, ID: 107/0x2B
Drive port: 2, Channel: 2, ID: 107/0x2B
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEVK00007502B0QZ
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:39
Drive type: Fibre Channel
Speed: 15015 RPM

Mode: Assigned
Associated array: 4

Rack 3

PROFILE FOR STORAGE SUBSYSTEM: RACK3 (3/21/05 3:44:04 PM)

SUMMARY-----

Number of controllers: 2
Number of arrays: 4
Total number of logical drives (includes an access logical drive): 5 of 2048 used

Number of standard logical drives: 4

Number of access logical drives: 1

Number of drives: 112

Supported drive types: Fibre (112)

Total hot spare drives: 0

Standby: 0

In use: 0

Access logical drive: None mapped

Default host type: Windows 2000/Server 2003 Non-Clustered (Host type index 2)

Current configuration

Firmware version: 06.10.06.00

NVSRAM version: N1742F900R910V02

Pending configuration

Staged firmware download supported?: Yes

Firmware version: None

NVSRAM version: None

Transferred on: None

NVSRAM configured for batteries?: Yes

Start cache flushing at (in percentage): 80

Stop cache flushing at (in percentage): 80

Cache block size (in KB): 4

Media scan frequency (in days): Disabled

Failover alert delay (in minutes): 5

Feature enable identifier: 3139313537000000000000041AF4222

Storage Subsystem worldwide name (ID):
600A0B8000133E5B0000000041B48482

CONTROLLERS-----

Number of controllers: 2

Controller in Slot A

Status: Online

Current configuration

Firmware version: 06.10.06.00

Appware version: 06.10.06.00

Bootware version: 06.10.01.00

NVSRAM version: N1742F900R910V02

Pending configuration

Firmware version: None

Appware version: None

Bootware version: None

NVSRAM version: None

Transferred on: None

Board ID: 5884

Product ID: 1742-900

Product revision: 0520

Serial number: 1T35119157

Date of manufacture: July 23, 2004

Cache/processor size (MB): 1024/128

Date/Time: Mon Mar 21 15:42:13 EST 2005

Associated Logical Drives (* = Preferred Owner):

RACK3_0*, RACK3_1*

Ethernet port: 1

MAC address: 00:a0:b8:13:3e:5b

Host name: RACK3_A

Network configuration: Static
IP address: 192.168.128.104
Subnet mask: 255.255.255.0
Gateway: 0.0.0.0
Remote login: Disabled

Drive interface: Fibre
Channel: 1
Current ID: 125/0x1
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Drive interface: Fibre
Channel: 2
Current ID: 125/0x1
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Drive interface: Fibre
Channel: 3
Current ID: 125/0x1
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Drive interface: Fibre
Channel: 4
Current ID: 125/0x1
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Host interface: Fibre
Port: 1
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 126/0x0
NL-Port ID: 0x010800
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up
Topology: Fabric Attach
World-wide port name: 20:02:00:a0:b8:13:3e:5c
World-wide node name: 20:02:00:a0:b8:13:3e:5b
Part type: HPFC-5400 revision 6

Host interface: Fibre
Port: 2
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 1/0xE8
NL-Port ID: 0x000000
Maximum data rate: 2 Gbps
Current data rate: 1 Gbps
Data rate control: Switch
Link status: Down
Topology: Not available
World-wide port name: 20:02:00:a0:b8:13:3e:5d
World-wide node name: 20:02:00:a0:b8:13:3e:5b
Part type: HPFC-5400 revision 6

Controller in Slot B
Status: Online
Current configuration
Firmware version: 06.10.06.00
Appware version: 06.10.06.00
Bootware version: 06.10.01.00
NVS RAM version: N1742F900R910V02

Pending configuration
Firmware version: None
Appware version: None
Bootware version: None
NVS RAM version: None
Transferred on: None
Board ID: 5884
Product ID: 1742-900
Product revision: 0520
Serial number: 1T35119187
Date of manufacture: December 26, 2003
Cache/processor size (MB): 1024/128
Date/Time: Mon Mar 21 15:42:39 EST 2005
Associated Logical Drives (* = Preferred Owner):
RACK3_2*, RACK3_3*
Ethernet port: 1
MAC address: 00:a0:b8:13:3e:79
Host name: RACK3_B
Network configuration: Static
IP address: 192.168.128.105
Subnet mask: 255.255.255.0
Gateway: 0.0.0.0
Remote login: Disabled

Drive interface: Fibre
Channel: 1
Current ID: 124/0x2
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Drive interface: Fibre
Channel: 2
Current ID: 124/0x2
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Drive interface: Fibre
Channel: 3
Current ID: 124/0x2
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Drive interface: Fibre
Channel: 4
Current ID: 124/0x2
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up

Host interface: Fibre
Port: 1
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 126/0x0
NL-Port ID: 0x010900
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up
Topology: Fabric Attach
World-wide port name: 20:03:00:a0:b8:13:3e:5c
World-wide node name: 20:03:00:a0:b8:13:3e:5b
Part type: HPFC-5400 revision 6
Host interface: Fibre
Port: 2
Current ID: Not applicable/0xFFFFFFFF

Preferred ID: 3/0xE2
NL-Port ID: 0x000000
Maximum data rate: 2 Gbps
Current data rate: 1 Gbps
Data rate control: Switch
Link status: Down
Topology: Not available
World-wide port name: 20:03:00:a0:b8:13:3e:5d
World-wide node name: 20:03:00:a0:b8:13:3e:5b
Part type: HPFC-5400 revision 6

ARRAYS-----

Number of arrays: 4

Array 1 (RAID 0)

Status: Online
Drive type: Fibre Channel
Enclosure loss protection: No
Current owner: Controller in slot A
Associated logical drives and free capacities:
RACK3_0 (935.238 GB)

Associated drives (in piece order):

Drive at Enclosure 0, Slot 1
Drive at Enclosure 0, Slot 2
Drive at Enclosure 0, Slot 3
Drive at Enclosure 0, Slot 4
Drive at Enclosure 0, Slot 5
Drive at Enclosure 0, Slot 6
Drive at Enclosure 0, Slot 7
Drive at Enclosure 0, Slot 8
Drive at Enclosure 0, Slot 9
Drive at Enclosure 0, Slot 10
Drive at Enclosure 0, Slot 11
Drive at Enclosure 0, Slot 12
Drive at Enclosure 0, Slot 13
Drive at Enclosure 0, Slot 14
Drive at Enclosure 1, Slot 1
Drive at Enclosure 1, Slot 2
Drive at Enclosure 1, Slot 3
Drive at Enclosure 1, Slot 4
Drive at Enclosure 1, Slot 5
Drive at Enclosure 1, Slot 6
Drive at Enclosure 1, Slot 7
Drive at Enclosure 1, Slot 8
Drive at Enclosure 1, Slot 9
Drive at Enclosure 1, Slot 10
Drive at Enclosure 1, Slot 11
Drive at Enclosure 1, Slot 12
Drive at Enclosure 1, Slot 13
Drive at Enclosure 1, Slot 14

Array 2 (RAID 0)

Status: Online
Drive type: Fibre Channel
Enclosure loss protection: No
Current owner: Controller in slot A
Associated logical drives and free capacities:
RACK3_1 (935.238 GB)

Associated drives (in piece order):

Drive at Enclosure 2, Slot 1
Drive at Enclosure 2, Slot 2
Drive at Enclosure 2, Slot 3
Drive at Enclosure 2, Slot 4
Drive at Enclosure 2, Slot 5
Drive at Enclosure 2, Slot 6
Drive at Enclosure 2, Slot 7
Drive at Enclosure 2, Slot 8
Drive at Enclosure 2, Slot 9
Drive at Enclosure 2, Slot 10

Drive at Enclosure 2, Slot 11
Drive at Enclosure 2, Slot 12
Drive at Enclosure 2, Slot 13
Drive at Enclosure 2, Slot 14
Drive at Enclosure 3, Slot 1
Drive at Enclosure 3, Slot 2
Drive at Enclosure 3, Slot 3
Drive at Enclosure 3, Slot 4
Drive at Enclosure 3, Slot 5
Drive at Enclosure 3, Slot 6
Drive at Enclosure 3, Slot 7
Drive at Enclosure 3, Slot 8
Drive at Enclosure 3, Slot 9
Drive at Enclosure 3, Slot 10
Drive at Enclosure 3, Slot 11
Drive at Enclosure 3, Slot 12
Drive at Enclosure 3, Slot 13
Drive at Enclosure 3, Slot 14

Array 3 (RAID 0)

Status: Online
Drive type: Fibre Channel
Enclosure loss protection: No
Current owner: Controller in slot B
Associated logical drives and free capacities:
RACK3_2 (935.238 GB)

Associated drives (in piece order):

Drive at Enclosure 10, Slot 1
Drive at Enclosure 10, Slot 2
Drive at Enclosure 10, Slot 3
Drive at Enclosure 10, Slot 4
Drive at Enclosure 10, Slot 5
Drive at Enclosure 10, Slot 6
Drive at Enclosure 10, Slot 7
Drive at Enclosure 10, Slot 8
Drive at Enclosure 10, Slot 9
Drive at Enclosure 10, Slot 10
Drive at Enclosure 10, Slot 11
Drive at Enclosure 10, Slot 12
Drive at Enclosure 10, Slot 13
Drive at Enclosure 10, Slot 14
Drive at Enclosure 11, Slot 1
Drive at Enclosure 11, Slot 2
Drive at Enclosure 11, Slot 3
Drive at Enclosure 11, Slot 4
Drive at Enclosure 11, Slot 5
Drive at Enclosure 11, Slot 6
Drive at Enclosure 11, Slot 7
Drive at Enclosure 11, Slot 8
Drive at Enclosure 11, Slot 9
Drive at Enclosure 11, Slot 10
Drive at Enclosure 11, Slot 11
Drive at Enclosure 11, Slot 12
Drive at Enclosure 11, Slot 13
Drive at Enclosure 11, Slot 14

Array 4 (RAID 0)

Status: Online
Drive type: Fibre Channel
Enclosure loss protection: No
Current owner: Controller in slot B
Associated logical drives and free capacities:
RACK3_3 (935.238 GB)

Associated drives (in piece order):

Drive at Enclosure 13, Slot 1
Drive at Enclosure 13, Slot 2
Drive at Enclosure 13, Slot 3
Drive at Enclosure 13, Slot 4
Drive at Enclosure 13, Slot 5
Drive at Enclosure 13, Slot 6

Drive at Enclosure 13, Slot 7
 Drive at Enclosure 13, Slot 8
 Drive at Enclosure 13, Slot 9
 Drive at Enclosure 13, Slot 10
 Drive at Enclosure 13, Slot 11
 Drive at Enclosure 13, Slot 12
 Drive at Enclosure 13, Slot 13
 Drive at Enclosure 13, Slot 14
 Drive at Enclosure 12, Slot 1
 Drive at Enclosure 12, Slot 2
 Drive at Enclosure 12, Slot 3
 Drive at Enclosure 12, Slot 4
 Drive at Enclosure 12, Slot 5
 Drive at Enclosure 12, Slot 6
 Drive at Enclosure 12, Slot 7
 Drive at Enclosure 12, Slot 8
 Drive at Enclosure 12, Slot 9
 Drive at Enclosure 12, Slot 10
 Drive at Enclosure 12, Slot 11
 Drive at Enclosure 12, Slot 12
 Drive at Enclosure 12, Slot 13
 Drive at Enclosure 12, Slot 14

STANDARD LOGICAL DRIVES-----

SUMMARY

Number of standard logical drives: 4
 See other Logical Drives sub-tabs for premium feature information.

| NAME | STATUS | CAPACITY | RAID LEVEL | ARRAY |
|---------|---------|------------|------------|-------|
| RACK3_0 | Optimal | 935.238 GB | 0 | 1 |
| RACK3_1 | Optimal | 935.238 GB | 0 | 2 |
| RACK3_2 | Optimal | 935.238 GB | 0 | 3 |
| RACK3_3 | Optimal | 935.238 GB | 0 | 4 |

DETAILS

Logical Drive name: RACK3_0
 Logical Drive ID: 60:0a:0b:80:00:13:3e:5b:00:00:00:03:41:bd:56:aa
 Subsystem ID (SSID): 0
 Status: Optimal
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Preferred owner: Controller in slot A
 Current owner: Controller in slot A
 Capacity: 935.238 GB
 RAID level: 0
 Segment size: 64 KB
 Modification priority: Highest
 Associated array: 1
 Read cache: Enabled
 Write cache: Disabled
 Write cache without batteries: Disabled
 Write cache with mirroring: Disabled
 Flush write cache after (in seconds): 10.00
 Cache read ahead multiplier: 1
 Enable background media scan: Disabled
 Media scan with redundancy check: Disabled

Logical Drive name: RACK3_1
 Logical Drive ID: 60:0a:0b:80:00:13:3e:5b:00:00:00:05:41:bd:56:e2
 Subsystem ID (SSID): 1
 Status: Optimal
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Preferred owner: Controller in slot A
 Current owner: Controller in slot A
 Capacity: 935.238 GB
 RAID level: 0

Segment size: 64 KB
 Modification priority: Highest
 Associated array: 2
 Read cache: Enabled
 Write cache: Disabled
 Write cache without batteries: Disabled
 Write cache with mirroring: Disabled
 Flush write cache after (in seconds): 10.00
 Cache read ahead multiplier: 1
 Enable background media scan: Disabled
 Media scan with redundancy check: Disabled

Logical Drive name: RACK3_2
 Logical Drive ID: 60:0a:0b:80:00:13:3e:79:00:00:00:01:41:bd:57:6b
 Subsystem ID (SSID): 2
 Status: Optimal
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Preferred owner: Controller in slot B
 Current owner: Controller in slot B
 Capacity: 935.238 GB
 RAID level: 0
 Segment size: 64 KB
 Modification priority: Highest
 Associated array: 3
 Read cache: Enabled
 Write cache: Disabled
 Write cache without batteries: Disabled
 Write cache with mirroring: Disabled
 Flush write cache after (in seconds): 10.00
 Cache read ahead multiplier: 1
 Enable background media scan: Disabled
 Media scan with redundancy check: Disabled

Logical Drive name: RACK3_3
 Logical Drive ID: 60:0a:0b:80:00:13:3e:79:00:00:00:00:42:3e:c4:c9
 Subsystem ID (SSID): 3
 Status: Optimal
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Preferred owner: Controller in slot B
 Current owner: Controller in slot B
 Capacity: 935.238 GB
 RAID level: 0
 Segment size: 64 KB
 Modification priority: High
 Associated array: 4
 Read cache: Enabled
 Write cache: Disabled
 Write cache without batteries: Disabled
 Write cache with mirroring: Disabled
 Flush write cache after (in seconds): 10.00
 Cache read ahead multiplier: 1
 Enable background media scan: Disabled
 Media scan with redundancy check: Disabled

MISSING LOGICAL DRIVES-----

Number of missing logical drives: 0

DRIVES-----

SUMMARY

Number of drives: 112
 Supported drive types: Fibre (112)

BASIC:

| TRAY, SLOT | STATUS | CAPACITY | CURRENT DATA RATE | PRODUCT ID | FIRMWARE VERSION |
|------------|---------|-----------|-------------------|------------|------------------|
| 0, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 0, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 0, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 0, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 0, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 0, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 0, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 0, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 0, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 0, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 0, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 0, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 0, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 0, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 1, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 2, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 3, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 10, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 10, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 10, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 10, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 10, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 10, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 10, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 10, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 10, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 10, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |
| 10, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F B954 |

| | | | | | | |
|--------|---------|-----------|--------|------------|---|------|
| 10, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 10, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 11, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 12, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 1 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 2 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 3 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |

| | | | | | | |
|--------|---------|-----------|--------|------------|---|------|
| 13, 4 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 5 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 6 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 7 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 8 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 9 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 10 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 11 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 12 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 13 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |
| 13, 14 | Optimal | 33.902 GB | 2 Gbps | ST336753FC | F | B954 |

DRIVE CHANNELS:

| TRAY, SLOT | PREFERRED CHANNEL | REDUNDANT CHANNEL |
|------------|-------------------|-------------------|
| 0, 1 | 4 | 3 |
| 0, 2 | 3 | 4 |
| 0, 3 | 4 | 3 |
| 0, 4 | 3 | 4 |
| 0, 5 | 4 | 3 |
| 0, 6 | 3 | 4 |
| 0, 7 | 4 | 3 |
| 0, 8 | 3 | 4 |
| 0, 9 | 4 | 3 |
| 0, 10 | 3 | 4 |
| 0, 11 | 4 | 3 |
| 0, 12 | 3 | 4 |
| 0, 13 | 4 | 3 |
| 0, 14 | 3 | 4 |
| 1, 1 | 4 | 3 |
| 1, 2 | 3 | 4 |
| 1, 3 | 4 | 3 |
| 1, 4 | 3 | 4 |
| 1, 5 | 4 | 3 |
| 1, 6 | 3 | 4 |
| 1, 7 | 4 | 3 |
| 1, 8 | 3 | 4 |
| 1, 9 | 4 | 3 |
| 1, 10 | 3 | 4 |
| 1, 11 | 4 | 3 |
| 1, 12 | 3 | 4 |
| 1, 13 | 4 | 3 |
| 1, 14 | 3 | 4 |
| 2, 1 | 4 | 3 |
| 2, 2 | 3 | 4 |
| 2, 3 | 4 | 3 |
| 2, 4 | 3 | 4 |
| 2, 5 | 4 | 3 |
| 2, 6 | 3 | 4 |
| 2, 7 | 4 | 3 |
| 2, 8 | 3 | 4 |
| 2, 9 | 4 | 3 |
| 2, 10 | 3 | 4 |
| 2, 11 | 4 | 3 |
| 2, 12 | 3 | 4 |
| 2, 13 | 4 | 3 |
| 2, 14 | 3 | 4 |

| | | |
|--------|---|---|
| 3, 1 | 4 | 3 |
| 3, 2 | 3 | 4 |
| 3, 3 | 4 | 3 |
| 3, 4 | 3 | 4 |
| 3, 5 | 4 | 3 |
| 3, 6 | 3 | 4 |
| 3, 7 | 4 | 3 |
| 3, 8 | 3 | 4 |
| 3, 9 | 4 | 3 |
| 3, 10 | 3 | 4 |
| 3, 11 | 4 | 3 |
| 3, 12 | 3 | 4 |
| 3, 13 | 4 | 3 |
| 3, 14 | 3 | 4 |
| 10, 1 | 2 | 1 |
| 10, 2 | 1 | 2 |
| 10, 3 | 2 | 1 |
| 10, 4 | 1 | 2 |
| 10, 5 | 2 | 1 |
| 10, 6 | 1 | 2 |
| 10, 7 | 2 | 1 |
| 10, 8 | 1 | 2 |
| 10, 9 | 2 | 1 |
| 10, 10 | 1 | 2 |
| 10, 11 | 2 | 1 |
| 10, 12 | 1 | 2 |
| 10, 13 | 2 | 1 |
| 10, 14 | 1 | 2 |
| 11, 1 | 2 | 1 |
| 11, 2 | 1 | 2 |
| 11, 3 | 2 | 1 |
| 11, 4 | 1 | 2 |
| 11, 5 | 2 | 1 |
| 11, 6 | 1 | 2 |
| 11, 7 | 2 | 1 |
| 11, 8 | 1 | 2 |
| 11, 9 | 2 | 1 |
| 11, 10 | 1 | 2 |
| 11, 11 | 2 | 1 |
| 11, 12 | 1 | 2 |
| 11, 13 | 2 | 1 |
| 11, 14 | 1 | 2 |
| 12, 1 | 2 | 1 |
| 12, 2 | 1 | 2 |
| 12, 3 | 2 | 1 |
| 12, 4 | 1 | 2 |
| 12, 5 | 2 | 1 |
| 12, 6 | 1 | 2 |
| 12, 7 | 2 | 1 |
| 12, 8 | 1 | 2 |
| 12, 9 | 2 | 1 |
| 12, 10 | 1 | 2 |
| 12, 11 | 2 | 1 |
| 12, 12 | 1 | 2 |
| 12, 13 | 2 | 1 |
| 12, 14 | 1 | 2 |
| 13, 1 | 2 | 1 |
| 13, 2 | 1 | 2 |
| 13, 3 | 2 | 1 |
| 13, 4 | 1 | 2 |
| 13, 5 | 2 | 1 |
| 13, 6 | 1 | 2 |
| 13, 7 | 2 | 1 |
| 13, 8 | 1 | 2 |
| 13, 9 | 2 | 1 |
| 13, 10 | 1 | 2 |
| 13, 11 | 2 | 1 |
| 13, 12 | 1 | 2 |

| | | |
|--------|---|---|
| 13, 13 | 2 | 1 |
| 13, 14 | 1 | 2 |

HOT SPARE COVERAGE:

The following arrays are not protected: 2, 1, 3, 4

Total hot spare drives: 0

Standby: 0

In use: 0

DETAILS

Drive at Enclosure 0, Slot 1

Drive port: 1, Channel: 4, ID: 0/0xEF

Drive port: 2, Channel: 3, ID: 0/0xEF

Drive path redundancy: OK

Status: Optimal

Raw capacity: 33.902 GB

Usable capacity: 33.402 GB

Current data rate: 2 Gbps

Product ID: ST336753FC F

Firmware version: B954

Serial number: 3HX2NEQR00007453V8Y0

Vendor: IBM-ESXS

Date of manufacture: July 20, 2004

World-wide name: 20:00:00:0c:50:45:ab:1f

Drive type: Fibre Channel

Speed: 15015 RPM

Mode: Assigned

Associated array: 1

Drive at Enclosure 0, Slot 2

Drive port: 1, Channel: 3, ID: 1/0xE8

Drive port: 2, Channel: 4, ID: 1/0xE8

Drive path redundancy: OK

Status: Optimal

Raw capacity: 33.902 GB

Usable capacity: 33.402 GB

Current data rate: 2 Gbps

Product ID: ST336753FC F

Firmware version: B954

Serial number: 3HX2EEDY00007447PXCXN

Vendor: IBM-ESXS

Date of manufacture: May 26, 2004

World-wide name: 20:00:00:0c:50:d6:d0:ba

Drive type: Fibre Channel

Speed: 15015 RPM

Mode: Assigned

Associated array: 1

Drive at Enclosure 0, Slot 3

Drive port: 1, Channel: 4, ID: 2/0xE4

Drive port: 2, Channel: 3, ID: 2/0xE4

Drive path redundancy: OK

Status: Optimal

Raw capacity: 33.902 GB

Usable capacity: 33.402 GB

Current data rate: 2 Gbps

Product ID: ST336753FC F

Firmware version: B954

Serial number: 3HX2DJ3R000074411AY6

Vendor: IBM-ESXS

Date of manufacture: May 25, 2004

World-wide name: 20:00:00:0c:50:d6:cf:7b

Drive type: Fibre Channel

Speed: 15015 RPM

Mode: Assigned

Associated array: 1

Drive at Enclosure 0, Slot 4
Drive port: 1, Channel: 3, ID: 3/0xE2
Drive port: 2, Channel: 4, ID: 3/0xE2
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NF0600007502CX5N
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:1b
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 5
Drive port: 1, Channel: 4, ID: 4/0xE1
Drive port: 2, Channel: 3, ID: 4/0xE1
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2L4MX00007502CJK4
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a8:e6
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 6
Drive port: 1, Channel: 3, ID: 5/0xE0
Drive port: 2, Channel: 4, ID: 5/0xE0
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NF1L00007502AZCD
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:d1
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 7
Drive port: 1, Channel: 4, ID: 6/0xDC
Drive port: 2, Channel: 3, ID: 6/0xDC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NF3T00007502BEN2

Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ab:e6
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 8
Drive port: 1, Channel: 3, ID: 7/0xDA
Drive port: 2, Channel: 4, ID: 7/0xDA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2KVHY00007502BTKM
Vendor: IBM-ESXS
Date of manufacture: July 22, 2004
World-wide name: 20:00:00:0c:50:45:b9:4c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 9
Drive port: 1, Channel: 4, ID: 80/0x55
Drive port: 2, Channel: 3, ID: 80/0x55
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MNRH00007502CJA9
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a8:5d
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 10
Drive port: 1, Channel: 3, ID: 96/0x3A
Drive port: 2, Channel: 4, ID: 96/0x3A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2BA9E00007447BW69
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d8:cd
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 11
Drive port: 1, Channel: 4, ID: 64/0x72
Drive port: 2, Channel: 3, ID: 64/0x72

Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MYAD000075029SB4
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:ca
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 12
Drive port: 1, Channel: 3, ID: 72/0x67
Drive port: 2, Channel: 4, ID: 72/0x67
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2BWTN00007445P58U
Vendor: IBM-ESXS
Date of manufacture: May 11, 2004
World-wide name: 20:00:00:0c:50:d6:16:85
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 13
Drive port: 1, Channel: 4, ID: 88/0x4B
Drive port: 2, Channel: 3, ID: 88/0x4B
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MWYY0000745266DN
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:aa
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 0, Slot 14
Drive port: 1, Channel: 3, ID: 104/0x2E
Drive port: 2, Channel: 4, ID: 104/0x2E
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEXB000075029RTG
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:93

Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 1
Drive port: 1, Channel: 4, ID: 8/0xD9
Drive port: 2, Channel: 3, ID: 8/0xD9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEM100007502CX3W
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:73
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 2
Drive port: 1, Channel: 3, ID: 9/0xD6
Drive port: 2, Channel: 4, ID: 9/0xD6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EBTE00007416J0K6
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:d9:ca
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 3
Drive port: 1, Channel: 4, ID: 10/0xD5
Drive port: 2, Channel: 3, ID: 10/0xD5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DX400007447BVTT
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:db:21
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 4
Drive port: 1, Channel: 3, ID: 11/0xD4
Drive port: 2, Channel: 4, ID: 11/0xD4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB

Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MYEK00007502CWAU
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:d6
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 5
Drive port: 1, Channel: 4, ID: 12/0xD3
Drive port: 2, Channel: 3, ID: 12/0xD3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MZY900007502CJJN
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a8:e8
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 6
Drive port: 1, Channel: 3, ID: 13/0xD2
Drive port: 2, Channel: 4, ID: 13/0xD2
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2LA5500007453W8BR
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:45
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 7
Drive port: 1, Channel: 4, ID: 14/0xD1
Drive port: 2, Channel: 3, ID: 14/0xD1
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NF1E00007502A7ER
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:42
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned

Associated array: 1

Drive at Enclosure 1, Slot 8
Drive port: 1, Channel: 3, ID: 15/0xCE
Drive port: 2, Channel: 4, ID: 15/0xCE
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2C8XY00007447NFL0
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d8:97
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 9
Drive port: 1, Channel: 4, ID: 81/0x54
Drive port: 2, Channel: 3, ID: 81/0x54
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DJ9F00007447KUAW
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:ba
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 10
Drive port: 1, Channel: 3, ID: 97/0x39
Drive port: 2, Channel: 4, ID: 97/0x39
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX23V8J000074478XPS
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:db:71
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 11
Drive port: 1, Channel: 4, ID: 65/0x71
Drive port: 2, Channel: 3, ID: 65/0x71
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F

Firmware version: B954
Serial number: 3HX2MN5H00007502CWZP
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a8:a3
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 12

Drive port: 1, Channel: 3, ID: 73/0x66
Drive port: 2, Channel: 4, ID: 73/0x66
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MYA90000750293H5
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:5a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 13

Drive port: 1, Channel: 4, ID: 89/0x4A
Drive port: 2, Channel: 3, ID: 89/0x4A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX1ZG02000074293GC1
Vendor: IBM-ESXS
Date of manufacture: May 11, 2004
World-wide name: 20:00:00:0c:50:d6:14:d6
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 1, Slot 14

Drive port: 1, Channel: 3, ID: 105/0x2D
Drive port: 2, Channel: 4, ID: 105/0x2D
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N0TZ000075014MH5
Vendor: IBM-ESXS
Date of manufacture: July 22, 2004
World-wide name: 20:00:00:0c:50:45:b8:46
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 1

Drive at Enclosure 2, Slot 1

Drive port: 1, Channel: 4, ID: 16/0xCD
Drive port: 2, Channel: 3, ID: 16/0xCD
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MN9E00007502CJAG
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a8:2f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 2

Drive port: 1, Channel: 3, ID: 17/0xCC
Drive port: 2, Channel: 4, ID: 17/0xCC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DKDP00007447AY97
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:f7
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 3

Drive port: 1, Channel: 4, ID: 18/0xCB
Drive port: 2, Channel: 3, ID: 18/0xCB
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N3QR00007502CJMC
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:2b
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 4

Drive port: 1, Channel: 3, ID: 19/0xCA
Drive port: 2, Channel: 4, ID: 19/0xCA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2D6X300007447BW5H
Vendor: IBM-ESXS

Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d9:2b
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 5

Drive port: 1, Channel: 4, ID: 20/0xC9
Drive port: 2, Channel: 3, ID: 20/0xC9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NF7M00007453RMEY
Vendor: IBM-ESXS
Date of manufacture: July 22, 2004
World-wide name: 20:00:00:0c:50:45:b8:7e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 6

Drive port: 1, Channel: 3, ID: 21/0xC7
Drive port: 2, Channel: 4, ID: 21/0xC7
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DGHH00007447KU4Q
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:82
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 7

Drive port: 1, Channel: 4, ID: 22/0xC6
Drive port: 2, Channel: 3, ID: 22/0xC6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N52Y00007502BF4S
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ac:94
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 8

Drive port: 1, Channel: 3, ID: 23/0xC5
Drive port: 2, Channel: 4, ID: 23/0xC5
Drive path redundancy: OK

Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MN4800007502BF1N
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a8:2c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 9

Drive port: 1, Channel: 4, ID: 82/0x53
Drive port: 2, Channel: 3, ID: 82/0x53
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DNAM0000744781JZ
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:6b
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 10

Drive port: 1, Channel: 3, ID: 98/0x36
Drive port: 2, Channel: 4, ID: 98/0x36
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E1FA00007447AY6A
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d8:e6
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 11

Drive port: 1, Channel: 4, ID: 66/0x6E
Drive port: 2, Channel: 3, ID: 66/0x6E
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NDJ600007453RKTR
Vendor: IBM-ESXS
Date of manufacture: July 22, 2004
World-wide name: 20:00:00:0c:50:45:b8:79
Drive type: Fibre Channel

Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 12

Drive port: 1, Channel: 3, ID: 74/0x65
Drive port: 2, Channel: 4, ID: 74/0x65
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N2T100007502BTEU
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:b0
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 13

Drive port: 1, Channel: 4, ID: 90/0x49
Drive port: 2, Channel: 3, ID: 90/0x49
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N3EZ0000750294GA
Vendor: IBM-ESXS
Date of manufacture: July 22, 2004
World-wide name: 20:00:00:0c:50:45:b8:45
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 2, Slot 14

Drive port: 1, Channel: 3, ID: 106/0x2C
Drive port: 2, Channel: 4, ID: 106/0x2C
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E2NA00007447KURO
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d0:b4
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 1

Drive port: 1, Channel: 4, ID: 24/0xC3
Drive port: 2, Channel: 3, ID: 24/0xC3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB

Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MZGL00007502J101
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a9:68
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 2

Drive port: 1, Channel: 3, ID: 25/0xBC
Drive port: 2, Channel: 4, ID: 25/0xBC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2KQ5Z00007502CWL B
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a9:25
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 3

Drive port: 1, Channel: 4, ID: 26/0xBA
Drive port: 2, Channel: 3, ID: 26/0xBA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N04600007502BU14
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:1e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 4

Drive port: 1, Channel: 3, ID: 27/0xB9
Drive port: 2, Channel: 4, ID: 27/0xB9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX281RE00007441T2LT
Vendor: IBM-ESXS
Date of manufacture: May 31, 2004
World-wide name: 20:00:00:0c:50:d6:f0:71
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 5
Drive port: 1, Channel: 4, ID: 28/0xB6
Drive port: 2, Channel: 3, ID: 28/0xB6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MZQE00007502AZ3Y
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a7:d4
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 6
Drive port: 1, Channel: 3, ID: 29/0xB5
Drive port: 2, Channel: 4, ID: 29/0xB5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MRXM00007502AZPG
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:29
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 7
Drive port: 1, Channel: 4, ID: 30/0xB4
Drive port: 2, Channel: 3, ID: 30/0xB4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MY2Z00007502CJ9V
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:aa:fe
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 8
Drive port: 1, Channel: 3, ID: 31/0xB3
Drive port: 2, Channel: 4, ID: 31/0xB3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954

Serial number: 3HX2DRS600007447KUAC
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:92
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 9
Drive port: 1, Channel: 4, ID: 83/0x52
Drive port: 2, Channel: 3, ID: 83/0x52
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E02M00007447PXCP
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d0:93
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 10
Drive port: 1, Channel: 3, ID: 99/0x35
Drive port: 2, Channel: 4, ID: 99/0x35
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DY8900007447KTU5
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:b0:84
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 11
Drive port: 1, Channel: 4, ID: 67/0x6D
Drive port: 2, Channel: 3, ID: 67/0x6D
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EANN00007447AYHA
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:e6
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 12
Drive port: 1, Channel: 3, ID: 75/0x63

Drive port: 2, Channel: 4, ID: 75/0x63
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NERY00007502BE45
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:8c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 13

Drive port: 1, Channel: 4, ID: 91/0x47
Drive port: 2, Channel: 3, ID: 91/0x47
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E9H900007447RB36
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:bd
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 3, Slot 14

Drive port: 1, Channel: 3, ID: 107/0x2B
Drive port: 2, Channel: 4, ID: 107/0x2B
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DNCV0000744779K3
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:ce:b6
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 2

Drive at Enclosure 10, Slot 1

Drive port: 1, Channel: 2, ID: 0/0xEF
Drive port: 2, Channel: 1, ID: 0/0xEF
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEJ5000075029T02
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004

World-wide name: 20:00:00:0c:50:45:ab:d9
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 2

Drive port: 1, Channel: 1, ID: 1/0xE8
Drive port: 2, Channel: 2, ID: 1/0xE8
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2CZHE00007447QTG5
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:da:55
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 3

Drive port: 1, Channel: 2, ID: 2/0xE4
Drive port: 2, Channel: 1, ID: 2/0xE4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEWP00007502AZJQ
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:aa:d4
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 4

Drive port: 1, Channel: 1, ID: 3/0xE2
Drive port: 2, Channel: 2, ID: 3/0xE2
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NF3B00007502CJDD
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:b8
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 5

Drive port: 1, Channel: 2, ID: 4/0xE1
Drive port: 2, Channel: 1, ID: 4/0xE1
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EBVT00007447KU2Y
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d8:5c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 6
Drive port: 1, Channel: 1, ID: 5/0xE0
Drive port: 2, Channel: 2, ID: 5/0xE0
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEFY00007502BT8V
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:c0
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 7
Drive port: 1, Channel: 2, ID: 6/0xDC
Drive port: 2, Channel: 1, ID: 6/0xDC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MT900007502BER0
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ab:70
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 8
Drive port: 1, Channel: 1, ID: 7/0xDA
Drive port: 2, Channel: 2, ID: 7/0xDA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E1YG00007447RAYV
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d8:75
Drive type: Fibre Channel
Speed: 15015 RPM

Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 9
Drive port: 1, Channel: 2, ID: 80/0x55
Drive port: 2, Channel: 1, ID: 80/0x55
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2CMQZ00007447QTDH
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:56
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 10
Drive port: 1, Channel: 1, ID: 96/0x3A
Drive port: 2, Channel: 2, ID: 96/0x3A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEZV00007502AZVP
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:c7
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 11
Drive port: 1, Channel: 2, ID: 64/0x72
Drive port: 2, Channel: 1, ID: 64/0x72
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E1F000007447KUF5
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:da:31
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 12
Drive port: 1, Channel: 1, ID: 72/0x67
Drive port: 2, Channel: 2, ID: 72/0x67
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps

Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EAQ000007447AY51
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:cf:97
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 13
Drive port: 1, Channel: 2, ID: 88/0x4B
Drive port: 2, Channel: 1, ID: 88/0x4B
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MDXZ00007502BEHJ
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:5b
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 10, Slot 14
Drive port: 1, Channel: 1, ID: 104/0x2E
Drive port: 2, Channel: 2, ID: 104/0x2E
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEQG00007502CJKZ
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:d9
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 1
Drive port: 1, Channel: 2, ID: 8/0xD9
Drive port: 2, Channel: 1, ID: 8/0xD9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DP4700007446XLD9
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:cf:d4
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 2
Drive port: 1, Channel: 1, ID: 9/0xD6
Drive port: 2, Channel: 2, ID: 9/0xD6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E8Y200007447BWAP
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:d7
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 3
Drive port: 1, Channel: 2, ID: 10/0xD5
Drive port: 2, Channel: 1, ID: 10/0xD5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N4S500007502CX47
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a8:a2
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 4
Drive port: 1, Channel: 1, ID: 11/0xD4
Drive port: 2, Channel: 2, ID: 11/0xD4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NELE00007502B0UF
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:1a
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 5
Drive port: 1, Channel: 2, ID: 12/0xD3
Drive port: 2, Channel: 1, ID: 12/0xD3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2FTTB00007453RLB3

Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ae:f9
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 6

Drive port: 1, Channel: 1, ID: 13/0xD2
Drive port: 2, Channel: 2, ID: 13/0xD2
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2CDYP00007447KTZT
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:84
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 7

Drive port: 1, Channel: 2, ID: 14/0xD1
Drive port: 2, Channel: 1, ID: 14/0xD1
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EB5100007447BWC2
Vendor: IBM-ESXS
Date of manufacture: May 29, 2004
World-wide name: 20:00:00:0c:50:d6:da:ca
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 8

Drive port: 1, Channel: 1, ID: 15/0xCE
Drive port: 2, Channel: 2, ID: 15/0xCE
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MKBH00007502AZDH
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:4f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 9

Drive port: 1, Channel: 2, ID: 81/0x54
Drive port: 2, Channel: 1, ID: 81/0x54

Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N4BN00007453V8M8
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ac:2e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 10

Drive port: 1, Channel: 1, ID: 97/0x39
Drive port: 2, Channel: 2, ID: 97/0x39
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MHAL00007502AZ90
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a9:69
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 11

Drive port: 1, Channel: 2, ID: 65/0x71
Drive port: 2, Channel: 1, ID: 65/0x71
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MY3T00007502A7QQ
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:21
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 12

Drive port: 1, Channel: 1, ID: 73/0x66
Drive port: 2, Channel: 2, ID: 73/0x66
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2E0FP00007447NFHM
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d8:85

Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 13

Drive port: 1, Channel: 2, ID: 89/0x4A
Drive port: 2, Channel: 1, ID: 89/0x4A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NF2100007502BEZM
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:3d
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 11, Slot 14

Drive port: 1, Channel: 1, ID: 105/0x2D
Drive port: 2, Channel: 2, ID: 105/0x2D
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2ME0R00007453RLTE
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:53
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 3

Drive at Enclosure 12, Slot 1

Drive port: 1, Channel: 2, ID: 16/0xCD
Drive port: 2, Channel: 1, ID: 16/0xCD
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2ME4700007502CJGG
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:b0:0b
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 2

Drive port: 1, Channel: 1, ID: 17/0xCC
Drive port: 2, Channel: 2, ID: 17/0xCC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB

Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MY6G0000745266ES
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:aa:fa
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 3

Drive port: 1, Channel: 2, ID: 18/0xCB
Drive port: 2, Channel: 1, ID: 18/0xCB
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEKG00007502CJDK
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:aa:a4
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 4

Drive port: 1, Channel: 1, ID: 19/0xCA
Drive port: 2, Channel: 2, ID: 19/0xCA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MY6000007502CX2F
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:aa:af
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 5

Drive port: 1, Channel: 2, ID: 20/0xC9
Drive port: 2, Channel: 1, ID: 20/0xC9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEK1000075029T3J
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:61
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned

Associated array: 4

Drive at Enclosure 12, Slot 6

Drive port: 1, Channel: 1, ID: 21/0xC7
Drive port: 2, Channel: 2, ID: 21/0xC7
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MVNZ00007502AZX1
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:aa:ef
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 7

Drive port: 1, Channel: 2, ID: 22/0xC6
Drive port: 2, Channel: 1, ID: 22/0xC6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEJJ000075014L68
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:aa:d5
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 8

Drive port: 1, Channel: 1, ID: 23/0xC5
Drive port: 2, Channel: 2, ID: 23/0xC5
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEMJ00007502BDUC
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:02
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 9

Drive port: 1, Channel: 2, ID: 82/0x53
Drive port: 2, Channel: 1, ID: 82/0x53
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F

Firmware version: B954
Serial number: 3HX2N54Y00007502AZLM
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ac:14
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 10

Drive port: 1, Channel: 1, ID: 98/0x36
Drive port: 2, Channel: 2, ID: 98/0x36
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MY5R00007502J0KW
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:aa:9e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 11

Drive port: 1, Channel: 2, ID: 66/0x6E
Drive port: 2, Channel: 1, ID: 66/0x6E
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX33NP500007516LQTX
Vendor: IBM-ESXS
Date of manufacture: October 25, 2004
World-wide name: 20:00:00:11:c6:23:90:9f
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 12

Drive port: 1, Channel: 1, ID: 74/0x65
Drive port: 2, Channel: 2, ID: 74/0x65
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MSQC00007502BTEB
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:b4
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 13

Drive port: 1, Channel: 2, ID: 90/0x49
Drive port: 2, Channel: 1, ID: 90/0x49
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MWZ000007502BU27
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:72
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 12, Slot 14

Drive port: 1, Channel: 1, ID: 106/0x2C
Drive port: 2, Channel: 2, ID: 106/0x2C
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MZY000007502CJ8F
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:a8:35
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 1

Drive port: 1, Channel: 2, ID: 24/0xC3
Drive port: 2, Channel: 1, ID: 24/0xC3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2DDA90000744781AV
Vendor: IBM-ESXS
Date of manufacture: May 26, 2004
World-wide name: 20:00:00:0c:50:d6:d8:dd
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 2

Drive port: 1, Channel: 1, ID: 25/0xBC
Drive port: 2, Channel: 2, ID: 25/0xBC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MY8C00007502CX1T
Vendor: IBM-ESXS

Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:10
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 3

Drive port: 1, Channel: 2, ID: 26/0xBA
Drive port: 2, Channel: 1, ID: 26/0xBA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N3L700007502A7SY
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:7c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 4

Drive port: 1, Channel: 1, ID: 27/0xB9
Drive port: 2, Channel: 2, ID: 27/0xB9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N3CY00007502AZYS
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ad:c5
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 5

Drive port: 1, Channel: 2, ID: 28/0xB6
Drive port: 2, Channel: 1, ID: 28/0xB6
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEDQ0000750294N1
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:82
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 6

Drive port: 1, Channel: 1, ID: 29/0xB5
Drive port: 2, Channel: 2, ID: 29/0xB5
Drive path redundancy: OK

Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N61B00007502CWX5
Vendor: IBM-ESXS
Date of manufacture: July 22, 2004
World-wide name: 20:00:00:0c:50:45:b8:87
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 7
Drive port: 1, Channel: 2, ID: 30/0xB4
Drive port: 2, Channel: 1, ID: 30/0xB4
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2MV9600007502CX3B
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:af:04
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 8
Drive port: 1, Channel: 1, ID: 31/0xB3
Drive port: 2, Channel: 2, ID: 31/0xB3
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2NEMP00007502AZ9N
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ab:e7
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 9
Drive port: 1, Channel: 2, ID: 83/0x52
Drive port: 2, Channel: 1, ID: 83/0x52
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N10M000075029S2S
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:aa:f0
Drive type: Fibre Channel

Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 10
Drive port: 1, Channel: 1, ID: 99/0x35
Drive port: 2, Channel: 2, ID: 99/0x35
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2N4AE00007502BEQ9
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ab:e4
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 11
Drive port: 1, Channel: 2, ID: 67/0x6D
Drive port: 2, Channel: 1, ID: 67/0x6D
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX33N7A00007517B1WL
Vendor: IBM-ESXS
Date of manufacture: October 25, 2004
World-wide name: 20:00:00:11:c6:23:90:87
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 12
Drive port: 1, Channel: 1, ID: 75/0x63
Drive port: 2, Channel: 2, ID: 75/0x63
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B954
Serial number: 3HX2EDQP00007447NFFE
Vendor: IBM-ESXS
Date of manufacture: May 27, 2004
World-wide name: 20:00:00:0c:50:d6:d9:78
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 13
Drive port: 1, Channel: 2, ID: 91/0x47
Drive port: 2, Channel: 1, ID: 91/0x47
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB

Current data rate: 2 Gbps
 Product ID: ST336753FC F
 Firmware version: B954
 Serial number: 3HX2ME0L00007502CWGE
 Vendor: IBM-ESXS
 Date of manufacture: July 21, 2004
 World-wide name: 20:00:00:0c:50:45:af:a6
 Drive type: Fibre Channel
 Speed: 15015 RPM
 Mode: Assigned
 Associated array: 4

Drive at Enclosure 13, Slot 14
 Drive port: 1, Channel: 1, ID: 107/0x2B
 Drive port: 2, Channel: 2, ID: 107/0x2B
 Drive path redundancy: OK
 Status: Optimal
 Raw capacity: 33.902 GB
 Usable capacity: 33.402 GB
 Current data rate: 2 Gbps
 Product ID: ST336753FC F
 Firmware version: B954
 Serial number: 3HX2NEPG00007502J0WM
 Vendor: IBM-ESXS
 Date of manufacture: July 20, 2004
 World-wide name: 20:00:00:0c:50:45:ab:a9
 Drive type: Fibre Channel
 Speed: 15015 RPM
 Mode: Assigned
 Associated array: 4

Client Configuration

Microsoft Windows 2000 Client System Information Report

Following is the system information report for Client 1. The system information report for clients 2-6 are identical to this one.

System Information report written at: 03/21/2005 04:25:57 PM
 [System Information]

[Following are sub-categories of this main category]

[System Summary]

| Item | Value |
|---------------------------|---|
| OS Name | Microsoft Windows 2000 Server |
| Version | 5.0.2195 Service Pack 4 Build 2195 |
| OS Manufacturer | Microsoft Corporation |
| System Name | CLIENT20 |
| System Manufacturer | IBM |
| System Model | IBM eServer x226-[86482BU]- |
| System Type | X86-based PC |
| Processor | x86 Family 15 Model 3 Stepping 4 GenuineIntel ~3200 Mhz |
| Processor | x86 Family 15 Model 4 Stepping 1 GenuineIntel ~3200 Mhz |
| Processor | x86 Family 15 Model 3 Stepping 4 GenuineIntel ~3200 Mhz |
| Processor | x86 Family 15 Model 4 Stepping 1 GenuineIntel ~3200 Mhz |
| BIOS Version | PhoenixBIOS 4.0 Release 6.1.U |
| Windows Directory | C:\WINNT |
| System Directory | C:\WINNT\system32 |
| Boot Device | \Device\Harddisk0\Partition1 |
| Locale | United States |
| User Name | CLIENT20\Administrator |
| Time Zone | Eastern Standard Time |
| Total Physical Memory | 2,620,316 KB |
| Available Physical Memory | 2,364,456 KB |

Total Virtual Memory 7,176,064 KB
 Available Virtual Memory 6,805,400 KB
 Page File Space 4,555,748 KB
 Page File C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

| Resource | Device |
|----------|--|
| IRQ 16 | Intel(R) E7525/E7520/E7320 PCI Express Root Port A0 - 3595 |
| IRQ 16 | Broadcom NetXtreme Gigabit Ethernet |
| IRQ 16 | Intel(R) E7525/E7520/E7320 PCI Express Root Port A1 - 3596 |
| IRQ 16 | Intel(R) E7525/E7520 PCI Express Root Port B0 - 3597 |
| IRQ 16 | Intel(R) 82801EB USB Universal Host Controller - 24D2 |
| IRQ 16 | Intel(R) 82801EB USB Universal Host Controller - 24DE |

[DMA]

| Channel | Device | Status |
|---------|---------------------------------|--------|
| 4 | Direct memory access controller | OK |
| 2 | Standard floppy disk controller | OK |
| 3 | ECP Printer Port (LPT1) | OK |

[Forced Hardware]

| Device | PNP Device ID |
|--------------------|---------------|
| No Forced Hardware | |

[I/O]

| Address Range | Device | Status |
|---------------|--|--------|
| 0x0000-0x0CF7 | PCI bus | OK |
| 0x0000-0x0CF7 | Direct memory access controller | OK |
| 0x0D00-0xFFFF | PCI bus | OK |
| 0x2000-0x4FFF | Intel(R) E7525/E7520/E7320 PCI Express Root Port A1 - 3596 | OK |
| 0x2000-0x4FFF | Intel(R) 6700PXH PCI Express-to-PCI Bridge A - 0329 | OK |
| 0x2000-0x4FFF | Adaptec AIC-7902B - Ultra320 SCSI | OK |
| 0x3000-0x303F | Intel(R) PRO/1000 MT Dual Port Server Adapter | |
| 0x3040-0x307F | Intel(R) PRO/1000 MT Dual Port Server Adapter #2 | |
| 0x3080-0x30BF | Intel(R) PRO/1000 MT Dual Port Server Adapter #3 | |
| 0x30C0-0x30FF | Intel(R) PRO/1000 MT Dual Port Server Adapter #4 | |
| 0x2400-0x24FF | Adaptec AIC-7902B - Ultra320 SCSI | OK |
| 0x2C00-0x2CFF | Adaptec AIC-7902B - Ultra320 SCSI | OK |
| 0x2800-0x28FF | Adaptec AIC-7902B - Ultra320 SCSI | OK |
| 0x4000-0x4FFF | Intel(R) 6700PXH PCI Express-to-PCI Bridge B - 032A | OK |
| 0x4000-0x4FFF | Intel(R) PRO/1000 MT Dual Port Server Adapter #5 | |
| 0x4040-0x407F | Intel(R) PRO/1000 MT Dual Port Server Adapter #6 | |
| 0x1400-0x141F | Intel(R) 82801EB USB Universal Host Controller - 24D2 | |
| 0x1420-0x143F | Intel(R) 82801EB USB Universal Host Controller - 24D4 | |
| 0x1440-0x145F | Intel(R) 82801EB USB Universal Host Controller - 24D7 | |
| 0x1460-0x147F | Intel(R) 82801EB USB Universal Host Controller - 24DE | |
| 0x5000-0x50FF | RADEON 7000M (on board) | OK |

| | | |
|---------------|---|----|
| 0x03B0-0x03BB | RADEON 7000M (on board) | OK |
| 0x03C0-0x03DF | RADEON 7000M (on board) | OK |
| 0x0A79-0x0A79 | ISAPNP Read Data Port | OK |
| 0x0279-0x0279 | ISAPNP Read Data Port | OK |
| 0x0274-0x0277 | ISAPNP Read Data Port | OK |
| 0x0010-0x001F | Motherboard resources | OK |
| 0x0024-0x0025 | Motherboard resources | OK |
| 0x0028-0x0029 | Motherboard resources | OK |
| 0x002C-0x002D | Motherboard resources | OK |
| 0x002E-0x002F | Motherboard resources | OK |
| 0x0030-0x0031 | Motherboard resources | OK |
| 0x0034-0x0035 | Motherboard resources | OK |
| 0x0038-0x0039 | Motherboard resources | OK |
| 0x003C-0x003D | Motherboard resources | OK |
| 0x0050-0x0053 | Motherboard resources | OK |
| 0x0072-0x0077 | Motherboard resources | OK |
| 0x0080-0x0080 | Motherboard resources | OK |
| 0x0090-0x009F | Motherboard resources | OK |
| 0x00A4-0x00A5 | Motherboard resources | OK |
| 0x00A8-0x00A9 | Motherboard resources | OK |
| 0x00AC-0x00AD | Motherboard resources | OK |
| 0x00B0-0x00B5 | Motherboard resources | OK |
| 0x00B8-0x00B9 | Motherboard resources | OK |
| 0x00BC-0x00BD | Motherboard resources | OK |
| 0x04D0-0x04D1 | Motherboard resources | OK |
| 0x1000-0x107F | Motherboard resources | OK |
| 0x1180-0x118C | Motherboard resources | OK |
| 0x118D-0x118D | Motherboard resources | OK |
| 0x118F-0x118F | Motherboard resources | OK |
| 0x1190-0x11BF | Motherboard resources | OK |
| 0x0800-0x082F | Motherboard resources | OK |
| 0xFE00-0xFE00 | Motherboard resources | OK |
| 0x0081-0x008F | Direct memory access controller | OK |
| 0x00C0-0x00DF | Direct memory access controller | OK |
| 0x00F0-0x00FE | Numeric data processor | OK |
| 0x0020-0x0021 | Programmable interrupt controller | OK |
| 0x00A0-0x00A1 | Programmable interrupt controller | OK |
| 0x0070-0x0071 | System CMOS/real time clock | OK |
| 0x0061-0x0061 | System speaker | OK |
| 0x0040-0x0043 | System timer | OK |
| 0x118E-0x118E | Not Available | OK |
| 0x03F0-0x03F5 | Standard floppy disk controller | OK |
| 0x03F7-0x03F7 | Standard floppy disk controller | OK |
| 0x0378-0x037F | ECP Printer Port (LPT1) | OK |
| 0x0778-0x077F | ECP Printer Port (LPT1) | OK |
| 0x03F8-0x03FF | Communications Port (COM1) | OK |
| 0x02F8-0x02FF | Communications Port (COM2) | OK |
| 0x0060-0x0060 | Standard 101/102-Key or Microsoft Natural PS/2 Keyboard | OK |
| 0x0064-0x0064 | Standard 101/102-Key or Microsoft Natural PS/2 Keyboard | OK |
| 0x14A0-0x14AF | Intel(R) 82801EB Ultra ATA Storage Controllers | OK |
| 0x01F0-0x01F7 | Primary IDE Channel | OK |
| 0x03F6-0x03F6 | Primary IDE Channel | OK |
| 0x0170-0x0177 | Secondary IDE Channel | OK |
| 0x0376-0x0376 | Secondary IDE Channel | OK |
| 0x1100-0x111F | Intel(R) 82801EB SMBus Controller - 24D3 | OK |

[IRQs]

| IRQ Number | Device |
|------------|--|
| 9 | Microsoft ACPI-Compliant System |
| 16 | Intel(R) E7525/E7520/E7320 PCI Express Root Port A0 - 3595 |
| 16 | Broadcom NetXtreme Gigabit Ethernet |
| 16 | Intel(R) E7525/E7520/E7320 PCI Express Root Port A1 - 3596 |
| 16 | Intel(R) E7525/E7520 PCI Express Root Port B0 - 3597 |
| 16 | Intel(R) 82801EB USB Universal Host Controller - 24D2 |

| | |
|----|---|
| 16 | Intel(R) 82801EB USB Universal Host Controller - 24DE |
| 28 | Intel(R) PRO/1000 MT Dual Port Server Adapter |
| 29 | Intel(R) PRO/1000 MT Dual Port Server Adapter #2 |
| 24 | Intel(R) PRO/1000 MT Dual Port Server Adapter #3 |
| 25 | Intel(R) PRO/1000 MT Dual Port Server Adapter #4 |
| 30 | Adaptec AIC-7902B - Ultra320 SCSI |
| 31 | Adaptec AIC-7902B - Ultra320 SCSI |
| 48 | Intel(R) PRO/1000 MT Dual Port Server Adapter #5 |
| 49 | Intel(R) PRO/1000 MT Dual Port Server Adapter #6 |
| 19 | Intel(R) 82801EB USB Universal Host Controller - 24D4 |
| 18 | Intel(R) 82801EB USB Universal Host Controller - 24D7 |
| 23 | Intel(R) 82801EB USB2 Enhanced Host Controller - 24DD |
| 22 | RADEON 7000M (on board) |
| 13 | Numeric data processor |
| 8 | System CMOS/real time clock |
| 6 | Standard floppy disk controller |
| 4 | Communications Port (COM1) |
| 3 | Communications Port (COM2) |
| 12 | PS/2 Compatible Mouse |
| 1 | Standard 101/102-Key or Microsoft Natural PS/2 Keyboard |
| 14 | Primary IDE Channel |
| 15 | Secondary IDE Channel |
| 10 | Intel(R) 82801EB SMBus Controller - 24D3 |

[Memory]

| Range | Device | Status |
|-----------------------|--|--------|
| 0xA0000-0xBFFFF | PCI bus | OK |
| 0xA0000-0xBFFFF | RADEON 7000M (on board) | OK |
| 0xD8000-0xDBFFF | PCI bus | OK |
| 0xDC000-0xDFFFF | PCI bus | OK |
| 0xA0000000-0xFEBFFFFF | PCI bus | OK |
| 0xD0100000-0xD01FFFFF | Intel(R) E7525/E7520/E7320 PCI Express Root Port A0 - 3595 | OK |
| 0xD0100000-0xD01FFFFF | Broadcom NetXtreme Gigabit Ethernet | OK |
| 0xD0200000-0xD05FFFFF | Intel(R) E7525/E7520/E7320 PCI Express Root Port A1 - 3596 | OK |
| 0xD0200000-0xD05FFFFF | Intel(R) 6700PXH I/OxAPIC Interrupt Controller A - 0326 | OK |
| 0xD0300000-0xD04FFFFF | Intel(R) 6700PXH PCI Express-to-PCI Bridge A - 0329 | OK |
| 0xD0300000-0xD04FFFFF | Adaptec AIC-7902B - Ultra320 SCSI | OK |
| 0xD0320000-0xD033FFFF | Intel(R) PRO/1000 MT Dual Port Server Adapter | OK |
| 0xD0340000-0xD037FFFF | Intel(R) PRO/1000 MT Dual Port Server Adapter | OK |
| 0xD0380000-0xD039FFFF | Intel(R) PRO/1000 MT Dual Port Server Adapter #2 | OK |
| 0xD03A0000-0xD03BFFFF | Intel(R) PRO/1000 MT Dual Port Server Adapter #3 | OK |
| 0xD03C0000-0xD03FFFFF | Intel(R) PRO/1000 MT Dual Port Server Adapter #3 | OK |
| 0xD0400000-0xD041FFFF | Intel(R) PRO/1000 MT Dual Port Server Adapter #4 | OK |
| 0xD0302000-0xD0303FFF | Adaptec AIC-7902B - Ultra320 SCSI | OK |
| 0xD0500000-0xD05FFFFF | Intel(R) 6700PXH PCI Express-to-PCI Bridge B - 032A | OK |
| 0xD0500000-0xD05FFFFF | Intel(R) PRO/1000 MT Dual Port Server Adapter #5 | OK |
| 0xD0540000-0xD055FFFF | Intel(R) PRO/1000 MT Dual Port Server Adapter #5 | OK |
| 0xD0560000-0xD057FFFF | Intel(R) PRO/1000 MT Dual Port Server Adapter #6 | OK |
| 0xD0201000-0xD0201FFF | Intel(R) 6700PXH I/OxAPIC Interrupt Controller B - 0327 | OK |

```

0xD0000000-0xD00003FF Intel(R) 82801EB USB2 Enhanced Host
Controller - 24DD OK
0xD8000000-0xDFFFFFFF RADEON 7000M (on board) OK
0xD0600000-0xD060FFFF RADEON 7000M (on board) OK
0xE0000000-0xEFFFFFFF Motherboard resources OK
0xFEE00000-0xFEE0FFFF Motherboard resources OK
0xFEC81000-0xFEC81FFF Motherboard resources OK
0xFEC81400-0xFEC823FF Motherboard resources OK
0xFEBF0000-0xFEBFFFFF Intel(R) 82801EB Ultra ATA Storage
Controllers OK

```

[Components]

[Following are sub-categories of this main category]

[Multimedia]

[Following are sub-categories of this main category]

[Audio Codecs]

| Codec Version | Manufacturer Size | Description Creation Date | Status | File |
|--------------------------------|--------------------------------|---------------------------|-----------|--|
| c:\winnt\system32\lhacm.acm | Microsoft Corporation | | | |
| OK | C:\WINNT\system32\LHACM.ACM | 4.4.3385 | 33.27 KB | (34,064 bytes) 11/3/2004 1:58:47 PM |
| c:\winnt\system32\msadp32.acm | Microsoft Corporation | | | |
| OK | C:\WINNT\system32\MSADP32.ACM | 5.00.2134.1 | | 14.77 KB (15,120 bytes) 12/7/1999 7:00:00 AM |
| c:\winnt\system32\iac25_32.ax | Intel Corporation | Indeo® audio software | | |
| OK | C:\WINNT\system32\IAC25_32.AX | 2.05.53 | 195.00 KB | (199,680 bytes) 12/7/1999 7:00:00 AM |
| c:\winnt\system32\imaadp32.acm | Microsoft Corporation | | | |
| OK | C:\WINNT\system32\IMAADP32.ACM | 5.00.2195.6612 | | 16.27 KB (16,656 bytes) 11/3/2004 3:02:37 PM |
| c:\winnt\system32\tssoft32.acm | DSP GROUP, INC. | | OK | |
| C:\WINNT\system32\TSSOFT32.ACM | 1.01 | 9.27 KB (9,488 bytes) | | 12/7/1999 7:00:00 AM |
| c:\winnt\system32\msg723.acm | Microsoft Corporation | | | |
| OK | C:\WINNT\system32\MSG723.ACM | 4.4.3385 | 106.77 KB | (109,328 bytes) 11/3/2004 1:58:46 PM |
| c:\winnt\system32\msg711.acm | Microsoft Corporation | | | |
| OK | C:\WINNT\system32\MSG711.ACM | 5.00.2134.1 | | 10.27 KB (10,512 bytes) 12/7/1999 7:00:00 AM |
| c:\winnt\system32\msgsm32.acm | Microsoft Corporation | | | |
| OK | C:\WINNT\system32\MSGSM32.ACM | 5.00.2134.1 | | 22.27 KB (22,800 bytes) 12/7/1999 7:00:00 AM |

[Video Codecs]

| Codec Version | Manufacturer Size | Description Creation Date | Status | File |
|--------------------------------|--------------------------------|---------------------------|-----------|--|
| c:\winnt\system32\ir50_32.dll | Intel Corporation | Indeo® video 5.10 | | |
| OK | C:\WINNT\system32\IR50_32.DLL | R.5.10.15.2.55 | | 737.50 KB (755,200 bytes) 12/7/1999 7:00:00 AM |
| c:\winnt\system32\msh261.drv | Microsoft Corporation | | | |
| OK | C:\WINNT\system32\MSH261.DRV | 4.4.3385 | 163.77 KB | (167,696 bytes) 11/3/2004 1:58:46 PM |
| c:\winnt\system32\msh263.drv | Microsoft Corporation | | | |
| OK | C:\WINNT\system32\MSH263.DRV | 4.4.3385 | 252.27 KB | (258,320 bytes) 11/3/2004 1:58:27 PM |
| c:\winnt\system32\msvidc32.dll | Microsoft Corporation | | | |
| OK | C:\WINNT\system32\MSVIDC32.DLL | 5.00.2134.1 | | 27.27 KB (27,920 bytes) 12/7/1999 7:00:00 AM |
| c:\winnt\system32\iccvid.dll | Radius Inc. | | OK | |
| C:\WINNT\system32\ICCVID.DLL | 1.10.0.6 | 108.00 KB (110,592 bytes) | | 12/7/1999 7:00:00 AM |

```

c:\winnt\system32\msrle32.dll Microsoft Corporation
OK C:\WINNT\system32\MSRLE32.DLL 5.00.2195.6612
10.77 KB (11,024 bytes) 11/3/2004 3:02:45 PM
c:\winnt\system32\ir32_32.dll Intel(R) Corporation OK
C:\WINNT\system32\IR32_32.DLL Not Available 194.50 KB
(199,168 bytes) 12/7/1999 7:00:00 AM

```

[CD-ROM]

| Item | Value |
|------------------------------------|---------------------------|
| Drive D: | |
| Description | CD-ROM Drive |
| Media Loaded | False |
| Media Type | CD-ROM |
| Name | HL-DT-ST CD-ROM GCR-8482B |
| Manufacturer | (Standard CD-ROM drives) |
| Status | OK |
| Transfer Rate | Not Available |
| SCSI Target ID | 0 |
| PNP Device ID | |
| IDE\CDROMHL-DT-ST_CD-ROM_GCR-8482B | 1.04 \5&25B98AF5&0&0.0.0 |

[Sound Device]

| Item | Value |
|------------------|-------|
| No sound devices | |

[Display]

| Item | Value |
|---------------------|---|
| Name | RADEON 7000M (on board) |
| PNP Device ID | PCI\VEN_1002&DEV_5159&SUBSYS_02C81014&REV_004&3A321F38&0&20F0 |
| Adapter Type | RADEON 7000 (0x5159), ATI Technologies Inc. compatible |
| Adapter Description | RADEON 7000M (on board) |
| Adapter RAM | 16.00 MB (16,777,216 bytes) |
| Installed Drivers | ati2dvag.dll |
| Driver Version | 5.2.3790.2 |
| INF File | oem6.inf (ati2mtag_RV100 section) |
| Color Planes | 1 |
| Color Table Entries | 4294967296 |
| Resolution | 800 x 600 x 60 hertz |
| Bits/Pixel | 32 |

[Infrared]

| Item | Value |
|---------------------|-------|
| No infrared devices | |

[Input]

[Following are sub-categories of this main category]

[Keyboard]

| Item | Value |
|----------------------|---|
| Description | Standard 101/102-Key or Microsoft Natural PS/2 Keyboard |
| Name | Enhanced (101- or 102-key) |
| Layout | 00000409 |
| PNP Device ID | ACPI\PNP0303\5&9583612&0 |
| NumberOfFunctionKeys | 12 |

[Pointing Device]

Item Value
Hardware Type PS/2 Compatible Mouse
Number of Buttons 5
Status OK
PNP Device ID ACPI\PNP0F13\5&9583612&0
Power Management Supported False
Double Click Threshold 6
Handedness Right Handed Operation

[Modem]

Item Value
No modems

[Network]

[Following are sub-categories of this main category]

[Adapter]

Item Value
Name [00000000] RAS Async Adapter
Adapter Type Not Available
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset 3/21/2005 6:29:28 AM
Index 0
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Not Available

Name [00000001] WAN Miniport (L2TP)
Adapter Type Not Available
Product Name WAN Miniport (L2TP)
Installed True
PNP Device ID ROOT\MS_L2TPMINIPOINT\0000
Last Reset 3/21/2005 6:29:28 AM
Index 1
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Rasl2tp
Driver c:\winnt\system32\drivers\rasl2tp.sys (52112, 5.00.2195.6655)

Name [00000002] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Name WAN Miniport (PPTP)
Installed True
PNP Device ID ROOT\MS_PPTPMINIPOINT\0000
Last Reset 3/21/2005 6:29:28 AM

Index 2
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Service Name PptpMiniport
Driver c:\winnt\system32\drivers\rasptp.sys (48464, 5.00.2195.6711)

Name [00000003] Direct Parallel
Adapter Type Not Available
Product Name Direct Parallel
Installed True
PNP Device ID ROOT\MS_PTMINIPOINT\0000
Last Reset 3/21/2005 6:29:28 AM

Index 3
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Raspti
Driver c:\winnt\system32\drivers\raspti.sys (16880, 5.00.2146.1)

Name [00000004] WAN Miniport (IP)
Adapter Type Not Available
Product Name WAN Miniport (IP)
Installed True
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 3/21/2005 6:29:28 AM

Index 4
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name NdisWan
Driver c:\winnt\system32\drivers\ndiswan.sys (93360, 5.00.2195.6699)

Name [00000005] Intel(R) PRO/1000 MT Dual Port Server Adapter
Adapter Type Ethernet 802.3
Product Name Intel(R) PRO/1000 MT Dual Port Server Adapter
Installed True
PNP Device ID PCI\VEN_8086&DEV_1079&SUBSYS_117A8086&REV_03\5&1EA3B137&0&080018
Last Reset 3/21/2005 6:29:28 AM

Index 5
Service Name E1000
IP Address 192.168.123.10
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available

MAC Address 00:0E:0C:36:78:EC
Service Name E1000
IRQ Number 28
I/O Port 0x3000-0x303F
Driver c:\winnt\system32\drivers\e1000nt5.sys (170496, 8.4.21.0 built by: WinDDK)

Name [00000006] Intel(R) PRO/1000 MT Dual Port Server Adapter
Adapter Type Ethernet 802.3
Product Name Intel(R) PRO/1000 MT Dual Port Server Adapter
Installed True
PNP Device ID
PCI\VEN_8086&DEV_1079&SUBSYS_117A8086&REV_03\5&1EA3B137&0&090018
Last Reset 3/21/2005 6:29:28 AM
Index 6
Service Name E1000
IP Address 192.168.124.10
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:0E:0C:36:78:ED
Service Name E1000
IRQ Number 29
I/O Port 0x3040-0x307F
Driver c:\winnt\system32\drivers\e1000nt5.sys (170496, 8.4.21.0 built by: WinDDK)

Name [00000007] Intel(R) PRO/1000 MT Dual Port Server Adapter
Adapter Type Ethernet 802.3
Product Name Intel(R) PRO/1000 MT Dual Port Server Adapter
Installed True
PNP Device ID
PCI\VEN_8086&DEV_1079&SUBSYS_117A8086&REV_03\5&1EA3B137&0&100018
Last Reset 3/21/2005 6:29:28 AM
Index 7
Service Name E1000
IP Address 192.168.125.10
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:0E:0C:36:7B:60
Service Name E1000
IRQ Number 24
I/O Port 0x3080-0x30BF
Driver c:\winnt\system32\drivers\e1000nt5.sys (170496, 8.4.21.0 built by: WinDDK)

Name [00000008] Intel(R) PRO/1000 MT Dual Port Server Adapter
Adapter Type Ethernet 802.3
Product Name Intel(R) PRO/1000 MT Dual Port Server Adapter
Installed True
PNP Device ID
PCI\VEN_8086&DEV_1079&SUBSYS_117A8086&REV_03\5&1EA3B137&0&110018
Last Reset 3/21/2005 6:29:28 AM
Index 8
Service Name E1000
IP Address 192.168.50.225
IP Subnet 255.255.255.0
Default IP Gateway Not Available

DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:0E:0C:36:7B:61
Service Name E1000
IRQ Number 25
I/O Port 0x30C0-0x30FF
Driver c:\winnt\system32\drivers\e1000nt5.sys (170496, 8.4.21.0 built by: WinDDK)

Name [00000009] Intel(R) PRO/1000 MT Dual Port Server Adapter
Adapter Type Ethernet 802.3
Product Name Intel(R) PRO/1000 MT Dual Port Server Adapter
Installed True
PNP Device ID
PCI\VEN_8086&DEV_1079&SUBSYS_117A8086&REV_03\5&A4D5A19&0&080218
Last Reset 3/21/2005 6:29:28 AM
Index 9
Service Name E1000
IP Address 192.168.121.10
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:0E:0C:36:78:B6
Service Name E1000
IRQ Number 48
I/O Port 0x4000-0x4FFF
Driver c:\winnt\system32\drivers\e1000nt5.sys (170496, 8.4.21.0 built by: WinDDK)

Name [00000010] Intel(R) PRO/1000 MT Dual Port Server Adapter
Adapter Type Ethernet 802.3
Product Name Intel(R) PRO/1000 MT Dual Port Server Adapter
Installed True
PNP Device ID
PCI\VEN_8086&DEV_1079&SUBSYS_117A8086&REV_03\5&A4D5A19&0&090218
Last Reset 3/21/2005 6:29:28 AM
Index 10
Service Name E1000
IP Address 192.168.222.10
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:0E:0C:36:78:B7
Service Name E1000
IRQ Number 49
I/O Port 0x4040-0x407F
Driver c:\winnt\system32\drivers\e1000nt5.sys (170496, 8.4.21.0 built by: WinDDK)

Name [00000011] Broadcom NetXtreme Gigabit Ethernet
Adapter Type Ethernet 802.3
Product Name Broadcom NetXtreme Gigabit Ethernet
Installed True
PNP Device ID
PCI\VEN_14E4&DEV_1659&SUBSYS_02C61014&REV_01\4&1855300&0&0010
Last Reset 3/21/2005 6:29:28 AM
Index 11

Service Name b57w2k
 IP Address 192.168.122.20
 IP Subnet 255.255.255.0
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:0D:60:15:18:A3
 Service Name b57w2k
 IRQ Number 16
 Driver c:\winnt\system32\drivers\b57w2k.sys (192215, 7.80.0.0)

[Protocol]

Item Value
 Name MSAFD Tcpip [TCP/IP]
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 16 bytes
 MaximumMessageSize 0 bytes
 MessageOriented False
 MinimumAddressSize 16 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData True
 SupportsGracefulClosing True
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD Tcpip [UDP/IP]
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 16 bytes
 MaximumMessageSize 65467 bytes
 MessageOriented True
 MinimumAddressSize 16 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting True

Name RSVP UDP Service Provider
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 16 bytes
 MaximumMessageSize 65467 bytes
 MessageOriented True
 MinimumAddressSize 16 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption True
 SupportsExpeditedData False
 SupportsGracefulClosing False

SupportsGuaranteedBandwidth False
 SupportsMulticasting True
 Name RSVP TCP Service Provider
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 16 bytes
 MaximumMessageSize 0 bytes
 MessageOriented False
 MinimumAddressSize 16 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption True
 SupportsExpeditedData True
 SupportsGracefulClosing True
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{0A116012-8A70-4813-A60F-1179D8F7AE88}]
 SEQPACKET 8
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize 20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{0A116012-8A70-4813-A60F-1179D8F7AE88}]
 DATAGRAM 8
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize 20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{29FD1170-E61B-4B40-A9F0-4BCEDBDF8A33}]
 SEQPACKET 7
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 20 bytes

MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{29FD1170-E61B-4B40-A9F0-4BCEDEDBF8A3}]
 DATAGRAM 7
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{49CD30FF-A43F-4429-B4C6-DC5F2B3F2E53}]
 SEQPACKET 6
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{49CD30FF-A43F-4429-B4C6-DC5F2B3F2E53}]
 DATAGRAM 6
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False

SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{7B118AA5-DF50-4EB6-9CD2-D4FF625CDACB}]
 SEQPACKET 5
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{7B118AA5-DF50-4EB6-9CD2-D4FF625CDACB}]
 DATAGRAM 5
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{4FF97F97-C919-4FEB-87EA-6AF8D1BE84CF}]
 SEQPACKET 4
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{4FF97F97-C919-4FEB-87EA-6AF8D1BE84CF}]
 DATAGRAM 4

ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{72931A11-10E1-4813-9A35-A28A5575ED6F}]
 SEQPACKET 3
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{72931A11-10E1-4813-9A35-A28A5575ED6F}]
 DATAGRAM 3
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{E6D306A6-B48E-4A25-841B-F670AE1A5D60}]
 SEQPACKET 0
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False

SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{E6D306A6-B48E-4A25-841B-F670AE1A5D60}]
 DATAGRAM 0
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{19854BA0-8D07-490E-BE50-A95157216BD2}]
 SEQPACKET 1
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{19854BA0-8D07-490E-BE50-A95157216BD2}]
 DATAGRAM 1
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{E71C5B67-467D-4AAA-BFFB-3896F017D737}]
 SEQPACKET 2
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{E71C5B67-467D-4AAA-BFFB-3896F017D737}]
 DATAGRAM 2
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

[WinSock]

Item Value
 File c:\winnt\system32\winsock.dll
 Version 3.10
 Size 2.80 KB (2,864 bytes)

File c:\winnt\system32\wsock32.dll
 Version 5.00.2195.6603
 Size 21.27 KB (21,776 bytes)

[Ports]

[Following are sub-categories of this main category]

[Serial]

Item Value
 Name COM1
 Status OK
 PNP Device ID ACPI\PNP0501\1
 Maximum Input Buffer Size 0
 Maximum Output Buffer Size False
 Settable Baud Rate True
 Settable Data Bits True

Settable Flow Control True
 Settable Parity True
 Settable Parity Check True
 Settable Stop Bits True
 Settable RLSD True
 Supports RLSD True
 Supports 16 Bit Mode False
 Supports Special Characters False
 Baud Rate 9600
 Bits/Byte 8
 Stop Bits 1
 Parity None
 Busy 0
 Abort Read/Write on Error 0
 Binary Mode Enabled -1
 Continue XMit on XOff 0
 CTS Outflow Control 0
 Discard NULL Bytes 0
 DSR Outflow Control 0
 DSR Sensitivity 0
 DTR Flow Control Type Enable
 EOF Character 0
 Error Replace Character 0
 Error Replacement Enabled 0
 Event Character 0
 Parity Check Enabled 0
 RTS Flow Control Type Enable
 XOff Character 19
 XOffXMit Threshold 512
 XOn Character 17
 XOnXMit Threshold 2048
 XOnXOff InFlow Control 0
 XOnXOff OutFlow Control 0
 IRQ Number 4
 I/O Port 0x03F8-0x03FF
 Driver c:\winnt\system32\drivers\serial.sys (62736, 5.00.2195.6655)

Name COM2
 Status OK
 PNP Device ID ACPI\PNP0501\2
 Maximum Input Buffer Size 0
 Maximum Output Buffer Size False
 Settable Baud Rate True
 Settable Data Bits True
 Settable Flow Control True
 Settable Parity True
 Settable Parity Check True
 Settable Stop Bits True
 Settable RLSD True
 Supports RLSD True
 Supports 16 Bit Mode False
 Supports Special Characters False
 Baud Rate 9600
 Bits/Byte 8
 Stop Bits 1
 Parity None
 Busy 0
 Abort Read/Write on Error 0
 Binary Mode Enabled -1
 Continue XMit on XOff 0
 CTS Outflow Control 0
 Discard NULL Bytes 0
 DSR Outflow Control 0
 DSR Sensitivity 0
 DTR Flow Control Type Enable
 EOF Character 0
 Error Replace Character 0
 Error Replacement Enabled 0

Event Character 0
 Parity Check Enabled 0
 RTS Flow Control Type Enable
 XOff Character 19
 XOffXMit Threshold 512
 XOn Character 17
 XOnXMit Threshold 2048
 XOnXOff InFlow Control 0
 XOnXOff OutFlow Control 0
 IRQ Number 3
 I/O Port 0x02F8-0x02FF
 Driver c:\winnt\system32\drivers\serial.sys (62736, 5.00.2195.6655)

[Parallel]

Item Value
 Name LPT1
 PNP Device ID ACPI\PNP0401\5&9583612&0

[Storage]

[Following are sub-categories of this main category]

[Drives]

Item Value
 Drive A:
 Description 3 1/2 Inch Floppy Drive

Drive C:
 Description Local Fixed Disk
 Compressed False
 File System NTFS
 Size 33.90 GB (36,396,830,720 bytes)
 Free Space 29.95 GB (32,154,103,808 bytes)
 Volume Name
 Volume Serial Number FC4C7C1B
 Partition Disk #0, Partition #0
 Partition Size 33.90 GB (36,396,831,744 bytes)
 Starting Offset 32256 bytes
 Drive Description Disk drive
 Drive Manufacturer (Standard disk drives)
 Drive Model IBM-ESXS DTN036C3UCDY10FN SCSI Disk Device
 Drive BytesPerSector 512
 Drive MediaLoaded True
 Drive MediaType Fixed hard disk media
 Drive Partitions 1
 Drive SCSI Bus 0
 Drive SCSI Logical Unit 0
 Drive SCSI Port 3
 Drive SCSI Target ID 0
 Drive SectorsPerTrack 63
 Drive Size 36396864000 bytes
 Drive TotalCylinders 4425
 Drive TotalSectors 71087625
 Drive TotalTracks 1128375
 Drive TracksPerCylinder 255

[SCSI]

Item Value
 Name Adaptec AIC-7902B - Ultra320 SCSI
 Caption Adaptec AIC-7902B - Ultra320 SCSI
 Driver adpu320
 Status OK

PNP Device ID
 PCI\VEN_9005&DEV_801D&SUBSYS_02CC1014&REV_10\5&1EA3B137
 &0&180018
 Device ID
 PCI\VEN_9005&DEV_801D&SUBSYS_02CC1014&REV_10\5&1EA3B137
 &0&180018
 Device Map Not Available
 Index Not Available
 Max Number Controlled Not Available
 IRQ Number 30
 I/O Port 0x2400-0x24FF
 I/O Port 0x2000-0x4FFF
 Driver c:\winnt\system32\drivers\adpu320.sys (132608, 3.0.000.000 built
 by: WinDDK)

Name Adaptec AIC-7902B - Ultra320 SCSI
 Caption Adaptec AIC-7902B - Ultra320 SCSI
 Driver adpu320
 Status OK
 PNP Device ID
 PCI\VEN_9005&DEV_801D&SUBSYS_02CC1014&REV_10\5&1EA3B137
 &0&190018
 Device ID
 PCI\VEN_9005&DEV_801D&SUBSYS_02CC1014&REV_10\5&1EA3B137
 &0&190018
 Device Map Not Available
 Index Not Available
 Max Number Controlled Not Available
 IRQ Number 31
 I/O Port 0x2C00-0x2CFF
 I/O Port 0x2800-0x28FF
 Driver c:\winnt\system32\drivers\adpu320.sys (132608, 3.0.000.000 built
 by: WinDDK)

[Printing]

Name Port Name Server Name
 No printing information

[Problem Devices]

| Device | PNP Device ID | Error Code |
|---------------|---------------------------|------------|
| Not Available | ACPI\ASF0001\2&DABA3FF&0 | 28 |
| Not Available | ACPI\IBM3737\4&369939D9&0 | 28 |

[USB]

Device PNP Device ID
 Intel(R) 82801EB USB Universal Host Controller - 24D2
 PCI\VEN_8086&DEV_24D2&SUBSYS_02ED1014&REV_02\3&61AAA01&
 0&E8
 USB Root Hub USB\ROOT_HUB\4&39460DFB&0
 Intel(R) 82801EB USB Universal Host Controller - 24D4
 PCI\VEN_8086&DEV_24D4&SUBSYS_02ED1014&REV_02\3&61AAA01&
 0&E9
 USB Root Hub USB\ROOT_HUB\4&31D97CBA&0
 Intel(R) 82801EB USB Universal Host Controller - 24D7
 PCI\VEN_8086&DEV_24D7&SUBSYS_02ED1014&REV_02\3&61AAA01&
 0&EA
 USB Root Hub USB\ROOT_HUB\4&206D9F09&0
 Intel(R) 82801EB USB Universal Host Controller - 24DE
 PCI\VEN_8086&DEV_24DE&SUBSYS_02ED1014&REV_02\3&61AAA01
 &0&EB
 USB Root Hub USB\ROOT_HUB\4&CAEE98E&0
 Intel(R) 82801EB USB2 Enhanced Host Controller - 24DD
 PCI\VEN_8086&DEV_24DD&SUBSYS_02ED1014&REV_02\3&61AAA01
 &0&EF

USB 2.0 Root Hub USB\ROOT_HUB20\4&34B50607&0

[Software Environment]

[Following are sub-categories of this main category]

[Drivers]

| Name | Description | File | Type | Started | Start Mode | State | Status | Error Control | Accept Pause | Accept Stop |
|----------|--|--|---------------|---------|------------|----------|---------|---------------|--------------|-------------|
| abiosdsk | Abiosdsk | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Ignore | False |
| abp480n5 | abp480n5 | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| acpi | Microsoft ACPI Driver | c:\winnt\system32\drivers\acpi.sys | Kernel Driver | True | Normal | Running | OK | Boot | Running | OK |
| acpiec | ACPIEC | c:\winnt\system32\drivers\acpiec.sys | Kernel Driver | False | Normal | Stopped | OK | Disabled | Stopped | OK |
| adpu160m | adpu160m | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| adpu320 | adpu320 | c:\winnt\system32\drivers\adpu320.sys | Kernel Driver | True | Normal | Running | OK | Boot | Running | OK |
| afd | AFD Networking Support Environment | c:\winnt\system32\drivers\afd.sys | Kernel Driver | True | Auto | Running | OK | Normal | False | True |
| aha154x | Aha154x | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| aic116x | aic116x | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| aic78u2 | aic78u2 | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| aic78xx | aic78xx | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| ami0nt | ami0nt | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| amsint | amsint | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| asc | asc | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| asc3350p | asc3350p | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| asc3550 | asc3550 | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| asynmac | RAS Asynchronous Media Driver | c:\winnt\system32\drivers\asynmac.sys | Kernel Driver | False | False | Manual | Stopped | OK | Normal | False |
| atapi | Standard IDE/ESDI Hard Disk Controller | c:\winnt\system32\drivers\atapi.sys | Kernel Driver | True | True | Boot | Running | OK | Normal | False |
| atdisk | Atdisk | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Ignore | False |
| ati2mtag | ati2mtag | c:\winnt\system32\drivers\ati2mtag.sys | Kernel Driver | True | Ignore | True | True | Manual | Running | OK |
| atmarpc | ATM ARP Client Protocol | c:\winnt\system32\drivers\atmarpc.sys | Kernel Driver | False | False | Manual | Stopped | OK | Normal | False |
| audstub | Audio Stub Driver | c:\winnt\system32\drivers\audstub.sys | Kernel Driver | True | Normal | Running | OK | Manual | Running | OK |
| b57w2k | Broadcom NetXtreme Gigabit Ethernet | c:\winnt\system32\drivers\b57w2k.sys | Kernel Driver | True | True | Manual | Running | OK | Normal | False |

| | | | | | | | | | | |
|-----------|---|--|--------------------|-------|----------|----------|---------|--------|--------|-------|
| beep | Beep | c:\winnt\system32\drivers\beep.sys | Kernel Driver | True | System | Running | OK | Normal | False | |
| buslogic | BusLogic | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| cd20xrnt | cd20xrnt | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| cdaudio | Cdaudio | c:\winnt\system32\drivers\cdaudio.sys | Kernel Driver | False | System | Stopped | OK | Ignore | False | |
| cdfs | Cdfs | c:\winnt\system32\drivers\cdfs.sys | File System Driver | True | Disabled | Running | OK | Normal | False | |
| cdrom | CD-ROM Driver | c:\winnt\system32\drivers\cdrom.sys | Kernel Driver | True | System | Running | OK | Normal | False | |
| changer | Changer | Not Available | Kernel Driver | False | False | System | Stopped | OK | Ignore | False |
| cpqarray | Cpqarray | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| cpqarray2 | cpqarray2 | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| cpqfcalm | cpqfcalm | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| cpqfws2e | cpqfws2e | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| dac960nt | dac960nt | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| deckzpsx | deckzpsx | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| dfsdriver | DfsDriver | c:\winnt\system32\drivers\dfs.sys | File System Driver | True | Boot | Running | OK | Normal | False | |
| disk | Disk Driver | c:\winnt\system32\drivers\disk.sys | Kernel Driver | True | Boot | Running | OK | Normal | False | |
| diskperf | Diskperf | c:\winnt\system32\drivers\diskperf.sys | Kernel Driver | True | Boot | Running | OK | Normal | False | |
| dmboot | dmboot | c:\winnt\system32\drivers\dmboot.sys | Kernel Driver | False | Disabled | Stopped | OK | Normal | False | |
| dmio | Logical Disk Manager Driver | c:\winnt\system32\drivers\dmio.sys | Kernel Driver | True | Boot | Running | OK | Normal | False | |
| dmload | dmload | c:\winnt\system32\drivers\dmload.sys | Kernel Driver | True | Boot | Running | OK | Normal | False | |
| e1000 | Intel(R) PRO/1000 Network Connection Driver | c:\winnt\system32\drivers\e1000nt5.sys | Kernel Driver | True | Manual | Running | OK | Normal | False | |
| e100b | Intel(R) PRO Adapter Driver | c:\winnt\system32\drivers\e100bnt5.sys | Kernel Driver | False | Manual | Stopped | OK | Normal | False | |
| efs | EFS | c:\winnt\system32\drivers\efs.sys | File System Driver | True | Disabled | Running | OK | Normal | False | |
| fastfat | Fastfat | c:\winnt\system32\drivers\fastfat.sys | File System Driver | True | Disabled | Running | OK | Normal | False | |
| fd16_700 | Fd16_700 | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |
| fdc | Floppy Disk Controller Driver | c:\winnt\system32\drivers\fdc.sys | Kernel Driver | True | Manual | Running | OK | Normal | False | |
| fips | Fips | c:\winnt\system32\drivers\fips.sys | Kernel Driver | True | Auto | Running | OK | Normal | False | |
| fireport | fireport | Not Available | Kernel Driver | False | False | Disabled | Stopped | OK | Normal | False |

| | | | | | | | | | | | | | | | |
|--|---|--|---------------|---------|--------|--------|--|--|--|---|--|---------------|-------|--------|-------|
| flashpnt | flashpnt | Not Available | Kernel Driver | False | | | | | msfs | Msfs | c:\winnt\system32\drivers\msfs.sys | File System | | | |
| Disabled | Stopped | OK | Normal | False | False | | | | Driver | True | System | Running | OK | Normal | False |
| flpydisk | Floppy Disk Driver | c:\winnt\system32\drivers\flpydisk.sys | | | | | | | True | | | | | | |
| Kernel Driver | True | Manual | Running | OK | Normal | | | | mkserv | Microsoft Streaming Service Proxy | | | | | |
| False | True | | | | | | | | c:\winnt\system32\drivers\mkserv.sys | Kernel Driver | | | | | False |
| ftdisk | Volume Manager Driver | | | | | | | | Manual | Stopped | OK | Normal | False | False | |
| c:\winnt\system32\drivers\ftdisk.sys | | | | | | | | | mspclock | Microsoft Streaming Clock Proxy | | | | | |
| Kernel Driver | | | | | | | | | c:\winnt\system32\drivers\mspclock.sys | Kernel Driver | | | | | False |
| Boot | Running | OK | Normal | False | True | | | | Manual | Stopped | OK | Normal | False | False | |
| gpc | Generic Packet Classifier | | | | | | | | mspqm | Microsoft Streaming Quality Manager Proxy | | | | | |
| c:\winnt\system32\drivers\msgpc.sys | | | | | | | | | c:\winnt\system32\drivers\mspqm.sys | Kernel Driver | | | | | False |
| Kernel Driver | | | | | | | | | Manual | Stopped | OK | Normal | False | False | |
| Manual | Running | OK | Normal | False | True | | | | mup | Mup | c:\winnt\system32\drivers\mup.sys | File System | | | |
| i8042prt | i8042 Keyboard and PS/2 Mouse Port Driver | | | | | | | | Driver | True | Boot | Running | OK | Normal | False |
| c:\winnt\system32\drivers\i8042prt.sys | | | | | | | | | True | | | | | | |
| Kernel Driver | | | | | | | | | nal | Nal Service | \\?\c:\winnt\system32\drivers\iqvw32.sys | | | | |
| System | Running | OK | Normal | False | True | | | | Kernel Driver | False | Manual | Stopped | OK | Normal | |
| ibmfe | IBM 10/100 Ethernet PCI Adapter NT Driver | | | | | | | | False | | | | | | |
| c:\winnt\system32\drivers\ibmfnt5.sys | | | | | | | | | False | | | | | | |
| Kernel Driver | | | | | | | | | ncrc710 | Nrc710 | Not Available | Kernel Driver | | | False |
| Manual | Stopped | OK | Normal | False | False | | | | Disabled | Stopped | OK | Normal | False | False | |
| ini910u | ini910u | Not Available | Kernel Driver | | | | | | ndis | NDIS System Driver | c:\winnt\system32\drivers\ndis.sys | | | | |
| Disabled | Stopped | OK | Normal | False | False | | | | Kernel Driver | True | Boot | Running | OK | Normal | |
| intelide | IntelIde | Not Available | Kernel Driver | | | | | | False | True | | | | | |
| Disabled | Stopped | OK | Normal | False | False | | | | ndistapi | Remote Access NDIS TAPI Driver | | | | | |
| ipfilterdriver | IP Traffic Filter Driver | | | | | | | | c:\winnt\system32\drivers\ndistapi.sys | Kernel Driver | | | | | True |
| c:\winnt\system32\drivers\ipfltdrv.sys | | | | | | | | | Manual | Running | OK | Normal | False | True | |
| Kernel Driver | | | | | | | | | ndisui0 | NDIS Usermode I/O Protocol | | | | | |
| Manual | Stopped | OK | Normal | False | False | | | | c:\winnt\system32\drivers\ndisui0.sys | Kernel Driver | | | | | False |
| ipinip | IP in IP Tunnel Driver | c:\winnt\system32\drivers\ipinip.sys | | | | | | | Manual | Stopped | OK | Normal | False | False | |
| Kernel Driver | | False | Manual | Stopped | OK | Normal | | | ndiswan | Remote Access NDIS WAN Driver | | | | | |
| False | False | | | | | | | | c:\winnt\system32\drivers\ndiswan.sys | Kernel Driver | | | | | True |
| ipnat | IP Network Address Translator | c:\winnt\system32\drivers\ipnat.sys | | | | | | | Manual | Running | OK | Normal | False | True | |
| Kernel Driver | | False | Manual | Stopped | OK | Normal | | | ndproxy | NDIS Proxy | c:\winnt\system32\drivers\ndproxy.sys | | | | |
| False | False | | | | | | | | Kernel Driver | True | Manual | Running | OK | Normal | |
| ipsec | IPSEC driver | c:\winnt\system32\drivers\ipsec.sys | | | | | | | False | True | | | | | |
| Kernel Driver | | True | Manual | Running | OK | Normal | | | netbios | NetBIOS Interface | c:\winnt\system32\drivers\netbios.sys | | | | |
| False | True | | | | | | | | File System Driver | True | System | Running | OK | Normal | |
| ipsraidn | ipsraidn | Not Available | Kernel Driver | | | | | | False | True | | | | | |
| Disabled | Stopped | OK | Normal | False | False | | | | netbt | NetBios over Tcpip | c:\winnt\system32\drivers\netbt.sys | | | | |
| irenum | IR Enumerator Service | | | | | | | | Kernel Driver | True | System | Running | OK | Normal | |
| c:\winnt\system32\drivers\irenum.sys | | | | | | | | | False | True | | | | | |
| Kernel Driver | | | | | | | | | netdetect | NetDetect | c:\winnt\system32\drivers\netdect.sys | Kernel | | | |
| Manual | Stopped | OK | Normal | False | False | | | | Driver | False | Manual | Stopped | OK | Normal | False |
| isapnp | PnP ISA/EISA Bus Driver | | | | | | | | False | | | | | | |
| c:\winnt\system32\drivers\isapnp.sys | | | | | | | | | npfs | Npfs | c:\winnt\system32\drivers\npfs.sys | File System | | | |
| Kernel Driver | | | | | | | | | Driver | True | System | Running | OK | Normal | False |
| Boot | Running | OK | Critical | False | True | | | | True | | | | | | |
| kbdclass | Keyboard Class Driver | | | | | | | | ntfs | Ntfs | c:\winnt\system32\drivers\ntfs.sys | File System | | | |
| c:\winnt\system32\drivers\kbdclass.sys | | | | | | | | | Driver | True | Disabled | Running | OK | Normal | False |
| Kernel Driver | | | | | | | | | True | | | | | | |
| System | Running | OK | Normal | False | True | | | | null | Null | c:\winnt\system32\drivers\null.sys | Kernel | | | |
| System | Running | OK | Normal | False | True | | | | Driver | True | System | Running | OK | Normal | False |
| ksecdd | KSecDD | c:\winnt\system32\drivers\ksecdd.sys | | | | | | | True | | | | | | |
| Driver | | True | Boot | Running | OK | Normal | | | nwlnkflt | IPX Traffic Filter Driver | | | | | |
| True | | | | | | | | | c:\winnt\system32\drivers\nwlnkflt.sys | Kernel Driver | | | | | False |
| lbrtdc | lbrtdc | Not Available | Kernel Driver | | | | | | Manual | Stopped | OK | Normal | False | False | |
| System | Stopped | OK | Ignore | False | False | | | | nwlnkfld | IPX Traffic Forwarder Driver | | | | | |
| System | Stopped | OK | Ignore | False | False | | | | c:\winnt\system32\drivers\nwlnkfld.sys | Kernel Driver | | | | | False |
| lp6nds35 | lp6nds35 | Not Available | Kernel Driver | | | | | | Manual | Stopped | OK | Normal | False | False | |
| Disabled | Stopped | OK | Normal | False | False | | | | parallel | Parallel class driver | c:\winnt\system32\drivers\parallel.sys | | | | |
| Kernel Driver | | | | | | | | | Kernel Driver | True | Manual | Running | OK | Normal | |
| mnmd | mnmd | c:\winnt\system32\drivers\mnmd.sys | | | | | | | False | True | | | | | |
| Driver | | True | System | Running | OK | Ignore | | | parport | Parallel port driver | c:\winnt\system32\drivers\parport.sys | | | | |
| True | | | | | | | | | Kernel Driver | True | System | Running | OK | Ignore | |
| modem | Modem | c:\winnt\system32\drivers\modem.sys | | | | | | | False | True | | | | | |
| Driver | | False | Manual | Stopped | OK | Ignore | | | partmgr | PartMgr | c:\winnt\system32\drivers\partmgr.sys | Kernel | | | |
| False | | | | | | | | | Driver | True | Boot | Running | OK | Normal | False |
| False | | | | | | | | | True | | | | | | |
| mouclass | Mouse Class Driver | c:\winnt\system32\drivers\mouclass.sys | | | | | | | | | | | | | |
| Kernel Driver | | True | System | Running | OK | Normal | | | | | | | | | |
| False | True | | | | | | | | | | | | | | |
| mountmgr | MountMgr | c:\winnt\system32\drivers\mountmgr.sys | | | | | | | | | | | | | |
| Driver | | True | Boot | Running | OK | Normal | | | | | | | | | |
| True | | | | | | | | | | | | | | | |
| mraid35x | mraid35x | Not Available | Kernel Driver | | | | | | | | | | | | |
| Disabled | Stopped | OK | Normal | False | False | | | | | | | | | | |
| mrxsmb | MRXSMB | c:\winnt\system32\drivers\mrxsmb.sys | | | | | | | | | | | | | |
| Driver | | True | System | Running | OK | Normal | | | | | | | | | |
| True | | | | | | | | | | | | | | | |


```

usbhub20 USB 2.0 Root Hub Support
c:\winnt\system32\drivers\usbhub20.sys Kernel Driver True
Manual Running OK Normal False True
vgasave VgaSave c:\winnt\system32\drivers\vga.sys Kernel
Driver True System Running OK Ignore False
True
wanarp Remote Access IP ARP Driver
c:\winnt\system32\drivers\wanarp.sys Kernel Driver True
Manual Running OK Normal False True
wdica WDICA Not Available Kernel Driver False
Manual Stopped OK Ignore False False

```

[Environment Variables]

```

Variable Value User Name
CLASSPATH
.;C:\SQLLIB\java\db2java.zip;C:\SQLLIB\java\db2jcc.jar;C:\SQLLIB\java\db2
jcc_license_cu.jar;C:\SQLLIB\bin <SYSTEM>
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
DB2INSTANCE DB2 <SYSTEM>
DB2TEMPDIR C:\SQLLIB\ <SYSTEM>
NUMBER_OF_PROCESSORS 4 <SYSTEM>
OS Windows_NT <SYSTEM>
Os2LibPath %SystemRoot%\system32\os2\dll; <SYSTEM>
Path
%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;
C:\SQLLIB\BIN;C:\SQLLIB\FUNCTION;c:\tools;c:\tools\util;C:\Program
Files\Inte\NDMIX <SYSTEM>
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH <SYSTEM>
PROCESSOR_ARCHITECTURE x86 <SYSTEM>
PROCESSOR_IDENTIFIER x86 Family 15 Model 3 Stepping 4,
GenuineIntel <SYSTEM>
PROCESSOR_LEVEL 15 <SYSTEM>
PROCESSOR_REVISION 0304 <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
windir %SystemRoot% <SYSTEM>
include C:\Program Files\Microsoft Visual
Studio\VC98\atl\include;C:\Program Files\Microsoft Visual
Studio\VC98\mfc\include;C:\Program Files\Microsoft Visual
Studio\VC98\include CLIENT20\Administrator
lib C:\Program Files\Microsoft Visual Studio\VC98\mfc\lib;C:\Program
Files\Microsoft Visual Studio\VC98\lib CLIENT20\Administrator
MSDevDirC:\Program Files\Microsoft Visual Studio\Common\MSDev98
CLIENT20\Administrator
path C:\Program Files\Microsoft Visual
Studio\Common\Tools\WinNT;C:\Program Files\Microsoft Visual
Studio\Common\MSDev98\Bin;C:\Program Files\Microsoft Visual
Studio\Common\Tools;C:\Program Files\Microsoft Visual Studio\VC98\bin
CLIENT20\Administrator
TEMP %USERPROFILE%\Local Settings\Temp
CLIENT20\Administrator
TMP %USERPROFILE%\Local Settings\Temp
CLIENT20\Administrator

```

[Jobs]

[Following are sub-categories of this main category]

[Print]

```

Document Size Owner Notify Status Time Submitted
Start Time Until Time Elapsed Time Pages Printed Job ID
Priority Parameters Driver Name Print Processor Host Print
Queue Data Type Name

```

No print jobs

[Network Connections]

| Local Name | Remote Name | Type | Status | User Name |
|------------|-----------------|------|--------|-----------|
| E: | \\fsserv\ddrive | Disk | OK | |

[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 | 0 | Not |
| Available | Not Available | Not Available | Unknown | Unknown | Unknown |
| Unknown | | | | | |
| system | Not Available | 8 | 8 | 0 | 1413120 |
| Not Available | Unknown | Unknown | Unknown | Unknown | |
| smss.exe | c:\winnt\system32\smss.exe | 192 | 11 | 204800 | |
| 1413120 | 3/21/2005 11:29:48 AM | 5.00.2195.6601 | | 44.77 KB | |
| (45,840 bytes) | 12/7/1999 7:00:00 AM | | | | |
| csrss.exe | Not Available | 216 | 13 | Not Available | |
| Not Available | 3/21/2005 11:29:51 AM | | | Unknown | Unknown |
| Unknown | | | | | |
| winlogon.exe | c:\winnt\system32\winlogon.exe | 212 | 13 | | |
| 204800 | 1413120 3/21/2005 11:29:52 AM | 5.00.2195.6714 | | | |
| 176.77 KB (181,008 bytes) | 11/3/2004 3:03:01 PM | | | | |
| services.exe | c:\winnt\system32\services.exe | 268 | 9 | | |
| 204800 | 1413120 3/21/2005 11:29:53 AM | 5.00.2195.6700 | | | |
| 87.27 KB (89,360 bytes) | 12/7/1999 7:00:00 AM | | | | |
| lsass.exe | c:\winnt\system32\lsass.exe | 280 | 9 | 204800 | |
| 1413120 | 3/21/2005 11:29:53 AM | 5.00.2195.6695 | | 32.77 KB | |
| (33,552 bytes) | 12/7/1999 7:00:00 AM | | | | |
| termsrv.exe | c:\winnt\system32\termsrv.exe | 388 | 10 | | |
| 204800 | 1413120 3/21/2005 11:29:54 AM | 5.00.2195.6696 | | | |
| 139.27 KB (142,608 bytes) | 11/3/2004 3:02:58 PM | | | | |
| svchost.exe | c:\winnt\system32\svchost.exe | 492 | 8 | | |
| 204800 | 1413120 3/21/2005 11:29:55 AM | 5.00.2134.1 | | | |
| 7.77 KB (7,952 bytes) | 12/7/1999 7:00:00 AM | | | | |
| spoolsv.exe | c:\winnt\system32\spoolsv.exe | 528 | 8 | | |
| 204800 | 1413120 3/21/2005 11:29:55 AM | 5.00.2195.6659 | | | |
| 44.27 KB (45,328 bytes) | 11/3/2004 8:42:04 AM | | | | |
| msdtc.exe | c:\winnt\system32\msdtc.exe | 556 | 8 | 204800 | |
| 1413120 | 3/21/2005 11:29:55 AM | 1999.9.3421.3 | | 6.77 KB | |
| (6,928 bytes) | 11/3/2004 8:55:08 AM | | | | |
| db2sec.exe | c:\sql\bin\db2sec.exe | 720 | 8 | 204800 | |
| 1413120 | 3/21/2005 11:29:56 AM | 8.1.7.447 | | 24.06 KB (24,638 | |
| bytes) | 8/15/2004 9:33:16 PM | | | | |
| svchost.exe | c:\winnt\system32\svchost.exe | 736 | 8 | | |
| 204800 | 1413120 3/21/2005 11:29:57 AM | 5.00.2134.1 | | | |
| 7.77 KB (7,952 bytes) | 12/7/1999 7:00:00 AM | | | | |
| llssrv.exe | c:\winnt\system32\llssrv.exe | 764 | 9 | 204800 | |
| 1413120 | 3/21/2005 11:29:57 AM | 5.00.2195.6697 | | 81.77 KB | |
| (83,728 bytes) | 6/19/2003 1:05:04 PM | | | | |
| regsvc.exe | c:\winnt\system32\regsvc.exe | 840 | 8 | 204800 | |
| 1413120 | 3/21/2005 11:29:58 AM | 5.00.2195.6701 | | 66.77 KB | |
| (68,368 bytes) | 11/3/2004 3:02:53 PM | | | | |
| mstask.exe | c:\winnt\system32\mstask.exe | 968 | 8 | 204800 | |
| 1413120 | 3/21/2005 11:30:13 AM | 4.71.2195.6704 | | 116.77 KB | |
| (119,568 bytes) | 11/3/2004 3:02:46 PM | | | | |
| tcpvcs.exe | c:\winnt\system32\tcpvcs.exe | 1052 | 8 | | |
| 204800 | 1413120 3/21/2005 11:30:13 AM | 5.00.2134.1 | | | |
| 24.77 KB (25,360 bytes) | 12/7/1999 7:00:00 AM | | | | |
| winmgmt.exe | c:\winnt\system32\wbem\winmgmt.exe | 1096 | 8 | | |
| 204800 | 1413120 3/21/2005 11:30:13 AM | 1.50.1085.0100 | | | |
| 192.10 KB (196,706 bytes) | 11/3/2004 3:03:07 PM | | | | |

| | | | |
|--------------------------------|--------------------------------------|---------------------------|----------------------|
| netplwiz.dll | 5.00.2195.6601 | 169.77 KB (173,840 bytes) | |
| 11/3/2004 3:02:48 PM | Microsoft Corporation | | |
| c:\winnt\system32\netplwiz.dll | | | |
| netmsg.dll | 5.00.2137.1 | 152.50 KB (156,160 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\netmsg.dll | | | |
| netui2.dll | 5.00.2134.1 | 280.27 KB (286,992 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\netui2.dll | | | |
| mprui.dll | 5.00.2195.6601 | 54.77 KB (56,080 bytes) | 11/3/2004 |
| 3:02:42 PM | Microsoft Corporation | | |
| c:\winnt\system32\mprui.dll | | | |
| hhsetup.dll | 5.2.3644.0 | 37.00 KB (37,888 bytes) | 11/3/2004 3:02:37 PM |
| Microsoft Corporation | c:\winnt\system32\hhsetup.dll | | |
| mmschext.dll | 5.00.2153.1 | 24.27 KB (24,848 bytes) | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\mmschext.dll | | | |
| rdpclip.exe | 5.00.2174.1 | 39.77 KB (40,720 bytes) | 11/3/2004 |
| 8:55:10 AM | Microsoft Corporation | | |
| c:\winnt\system32\rdpclip.exe | | | |
| mcsms.dll | 5.00.2180.1 | 68.27 KB (69,904 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\mcsms.dll | | | |
| printui.dll | 5.00.2195.6702 | 372.77 KB (381,712 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\printui.dll | | | |
| h323.tsp | 5.00.2195.6699 | 248.77 KB (254,736 bytes) | 11/3/2004 |
| 3:02:36 PM | Microsoft Corporation | | |
| c:\winnt\system32\h323.tsp | | | |
| ipconf.tsp | 5.00.2143.1 | 10.77 KB (11,024 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\ipconf.tsp | | | |
| ndptsp.tsp | 5.00.2143.1 | 38.27 KB (39,184 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\ndptsp.tsp | | | |
| kmddsp.tsp | 5.00.2150.1 | 17.77 KB (18,192 bytes) | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\kmddsp.tsp | | | |
| uniplat.dll | 5.00.2195.6601 | 14.27 KB (14,608 bytes) | 11/3/2004 |
| 3:02:59 PM | Microsoft Corporation | | |
| c:\winnt\system32\uniplat.dll | | | |
| unimdm.tsp | 5.00.2195.6601 | 199.27 KB (204,048 bytes) | |
| 11/3/2004 3:02:59 PM | Microsoft Corporation | | |
| c:\winnt\system32\unimdm.tsp | | | |
| tapisrv.dll | 5.00.2195.6666 | 169.27 KB (173,328 bytes) | 11/3/2004 |
| 3:02:58 PM | Microsoft Corporation | | |
| c:\winnt\system32\tapisrv.dll | | | |
| mdm.exe | 6.00.8424 | 121.29 KB (124,200 bytes) | 11/3/2004 8:56:48 AM |
| Microsoft Corporation | c:\winnt\system32\mdm.exe | | |
| faxshell.dll | 5.00.2134.1 | 8.27 KB (8,464 bytes) | 12/7/1999 7:00:00 AM |
| Microsoft Corporation | c:\winnt\system32\faxshell.dll | | |
| msacm32.dll | 5.00.2134.1 | 65.27 KB (66,832 bytes) | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\msacm32.dll | | | |
| avifil32.dll | 5.00.2195.6612 | 76.77 KB (78,608 bytes) | 11/3/2004 |
| 3:02:27 PM | Microsoft Corporation | | |
| c:\winnt\system32\avifil32.dll | | | |
| msvfw32.dll | 5.00.2195.6612 | 113.77 KB (116,496 bytes) | |
| 11/3/2004 3:02:46 PM | Microsoft Corporation | | |
| c:\winnt\system32\msvfw32.dll | | | |
| docprop2.dll | 5.00.2178.1 | 297.77 KB (304,912 bytes) | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\docprop2.dll | | | |
| imgutil.dll | 5.00.3700.6682 | 30.77 KB (31,504 bytes) | 11/3/2004 |
| 3:02:37 PM | Microsoft Corporation | | |
| c:\winnt\system32\imgutil.dll | | | |
| mydocs.dll | 5.00.3502.6601 | 55.77 KB (57,104 bytes) | 11/3/2004 |
| 3:02:47 PM | Microsoft Corporation | | |
| c:\winnt\system32\mydocs.dll | | | |
| webvw.dll | 5.00.2920.0000 | 1.06 MB (1,115,408 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\webvw.dll | | | |
| imm32.dll | 5.00.2195.6655 | 94.27 KB (96,528 bytes) | 11/3/2004 |
| 3:02:37 PM | Microsoft Corporation | | |
| c:\winnt\system32\imm32.dll | | | |
| msls31.dll | 3.10.337.0 | 145.27 KB (148,752 bytes) | 12/7/1999 7:00:00 AM |
| Microsoft Corporation | c:\winnt\system32\msls31.dll | | |
| msdbg.dll | 6.00.8424 | 67.50 KB (69,120 bytes) | 11/3/2004 8:56:48 AM |
| Microsoft Corporation | c:\winnt\system32\msdbg.dll | | |
| shdoclc.dll | 5.00.3700.6668 | 324.50 KB (332,288 bytes) | 11/3/2004 |
| 3:02:55 PM | Microsoft Corporation | | |
| c:\winnt\system32\shdoclc.dll | | | |
| pdm.dll | 6.00.8424 | 179.27 KB (183,574 bytes) | 11/3/2004 8:56:49 AM |
| Microsoft Corporation | c:\winnt\system32\pdm.dll | | |
| mshtml.dll | 5.00.3700.6699 | 2.24 MB (2,353,936 bytes) | 11/3/2004 |
| 3:02:43 PM | Microsoft Corporation | | |
| c:\winnt\system32\mshtml.dll | | | |
| mlang.dll | 5.00.3700.6655 | 510.77 KB (523,024 bytes) | 11/3/2004 |
| 3:02:41 PM | Microsoft Corporation | | |
| c:\winnt\system32\mlang.dll | | | |
| urlmon.dll | 5.00.3700.6705 | 442.77 KB (453,392 bytes) | 11/3/2004 |
| 3:02:59 PM | Microsoft Corporation | | |
| c:\winnt\system32\urlmon.dll | | | |
| powrprof.dll | 5.00.3502.6601 | 13.27 KB (13,584 bytes) | |
| 11/3/2004 3:02:52 PM | Microsoft Corporation | | |
| c:\winnt\system32\powrprof.dll | | | |
| batmeter.dll | 5.00.3502.6601 | 20.27 KB (20,752 bytes) | |
| 11/3/2004 3:02:28 PM | Microsoft Corporation | | |
| c:\winnt\system32\batmeter.dll | | | |
| stobjct.dll | 5.00.2195.6601 | 79.27 KB (81,168 bytes) | 11/3/2004 |
| 3:02:57 PM | Microsoft Corporation | | |
| c:\winnt\system32\stobjct.dll | | | |
| msi.dll | 2.0.2600.1183 | 1.92 MB (2,017,792 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\msi.dll | | | |
| webcheck.dll | 5.00.3502.6601 | 251.77 KB (257,808 bytes) | |
| 11/3/2004 3:03:00 PM | Microsoft Corporation | | |
| c:\winnt\system32\webcheck.dll | | | |
| browsecl.dll | 5.00.3700.6661 | 34.50 KB (35,328 bytes) | |
| 11/3/2004 3:02:28 PM | Microsoft Corporation | | |
| c:\winnt\system32\browsecl.dll | | | |
| ntshrui.dll | 5.00.2134.1 | 46.77 KB (47,888 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\ntshrui.dll | | | |
| linkinfo.dll | 5.00.2134.1 | 15.77 KB (16,144 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\linkinfo.dll | | | |
| browseui.dll | 5.00.3700.6661 | 789.27 KB (808,208 bytes) | |
| 11/3/2004 3:02:28 PM | Microsoft Corporation | | |
| c:\winnt\system32\browseui.dll | | | |
| shdocvw.dll | 5.00.3700.6668 | 1.06 MB (1,107,728 bytes) | |
| 11/3/2004 3:02:55 PM | Microsoft Corporation | | |
| c:\winnt\system32\shdocvw.dll | | | |
| explorer.exe | 5.00.3700.6690 | 237.77 KB (243,472 bytes) | |
| 11/3/2004 3:03:02 PM | Microsoft Corporation | | |
| c:\winnt\explorer.exe | | | |
| dfssvc.exe | 5.00.2195.6664 | 88.77 KB (90,896 bytes) | 11/3/2004 |
| 3:02:32 PM | Microsoft Corporation | | |
| c:\winnt\system32\dfssvc.exe | | | |
| iislog.dll | 5.00.0984 | 75.27 KB (77,072 bytes) | 11/3/2004 3:03:21 PM |
| Microsoft Corporation | c:\winnt\system32\inetsrv\iislog.dll | | |
| ilsdbx.dll | 5.00.0984 | 56.27 KB (57,616 bytes) | 3/15/2005 3:51:03 PM |
| Microsoft Corporation | c:\winnt\system32\inetsrv\ilsdbx.dll | | |

| | | | | | | | | | | |
|--|----------------|-----------------------------|---------------------------|-----------------------|--|----------------|---------------------------|---------------------------|-----------------------|-----------------------|
| msrd3x40.dll | 4.00.6508.0 | 308.27 KB (315,664 bytes) | 11/3/2004 3:02:45 PM | Microsoft Corporation | w3svc.dll | 5.00.0984 | 338.27 KB (346,384 bytes) | 11/3/2004 3:03:23 PM | Microsoft Corporation | |
| c:\winnt\system32\msrd3x40.dll | | | | | c:\winnt\system32\inetsrv\w3svc.dll | | | | | |
| odbccp32.dll | 3.520.7713.0 | 92.00 KB (94,208 bytes) | 11/3/2004 5:27:34 PM | Microsoft Corporation | staxmem.dll | 5.00.0984 | 8.27 KB (8,464 bytes) | 11/3/2004 3:02:57 PM | Microsoft Corporation | |
| c:\winnt\system32\odbccp32.dll | | | | | c:\winnt\system32\staxmem.dll | | | | | |
| mtxdm.dll | 2000.2.3504.0 | 22.77 KB (23,312 bytes) | 3:02:47 PM | Microsoft Corporation | extrace.dll | 5.00.0984 | 13.77 KB (14,096 bytes) | 11/3/2004 8:55:28 AM | Microsoft Corporation | |
| c:\winnt\system32\mtxdm.dll | | | | | c:\winnt\system32\extrace.dll | | | | | |
| odbj32.dll | 4.0.6200.0 | 52.27 KB (53,520 bytes) | 3:02:50 PM | Microsoft Corporation | iisfecnv.dll | 5.00.0984 | 7.27 KB (7,440 bytes) | 11/3/2004 8:55:27 AM | Microsoft Corporation | |
| c:\winnt\system32\odbj32.dll | | | | | c:\winnt\system32\inetsrv\iisfecnv.dll | | | | | |
| odbj32.dll | 4.0.6200.0 | 264.27 KB (270,608 bytes) | 3:02:50 PM | Microsoft Corporation | isatq.dll | 5.00.0984 | 61.27 KB (62,736 bytes) | 11/3/2004 3:03:22 PM | Microsoft Corporation | |
| c:\winnt\system32\odbj32.dll | | | | | c:\winnt\system32\inetsrv\isatq.dll | | | | | |
| msdasqlr.dll | 2.70.7713.0 | built by: Lab06_N(dagbuild) | 16.00 KB (16,384 bytes) | 11/3/2004 5:27:34 PM | Microsoft Corporation | infocomm.dll | 5.00.0984 | 242.27 KB (248,080 bytes) | 3:03:22 PM | Microsoft Corporation |
| c:\program files\common files\system\ole db\msdasqlr.dll | | | | | c:\winnt\system32\inetsrv\infocomm.dll | | | | | |
| msdatl3.dll | 2.70.7713.0 | built by: Lab06_N(dagbuild) | 84.00 KB (86,016 bytes) | 11/3/2004 5:27:34 PM | Microsoft Corporation | ldapsvcx.dll | 5.00.0984 | 126.77 KB (129,808 bytes) | 3:03:33 PM | Microsoft Corporation |
| c:\program files\common files\system\ole db\msdatl3.dll | | | | | c:\winnt\system32\inetsrv\ldapsvcx.dll | | | | | |
| msdasql.dll | 2.70.7713.0 | built by: Lab06_N(dagbuild) | 296.00 KB (303,104 bytes) | 11/3/2004 5:27:34 PM | Microsoft Corporation | security.dll | 5.00.2154.1 | 5.77 KB (5,904 bytes) | 12/7/1999 7:00:00 AM | Microsoft Corporation |
| c:\program files\common files\system\ole db\msdasql.dll | | | | | c:\winnt\system32\security.dll | | | | | |
| ldapdbx.dll | 5.00.0984 | 73.27 KB (75,024 bytes) | 3:51:03 PM | Microsoft Corporation | svccxt.dll | 5.00.0984 | 39.77 KB (40,720 bytes) | 11/3/2004 3:03:23 PM | Microsoft Corporation | |
| c:\winnt\system32\inetsrv\ldapdbx.dll | | | | | c:\winnt\system32\inetsrv\svccxt.dll | | | | | |
| dscomobx.dll | 5.00.0984 | 188.77 KB (193,296 bytes) | 3:51:03 PM | Microsoft Corporation | admexs.dll | 5.00.0984 | 27.77 KB (28,432 bytes) | 11/3/2004 3:03:20 PM | Microsoft Corporation | |
| c:\winnt\system32\inetsrv\dscomobx.dll | | | | | c:\winnt\system32\inetsrv\admexs.dll | | | | | |
| httpext.dll | 5.00.0984 | 240.77 KB (246,544 bytes) | 11/3/2004 3:03:21 PM | Microsoft Corporation | wamreg.dll | 5.00.0984 | 45.77 KB (46,864 bytes) | 11/3/2004 3:03:24 PM | Microsoft Corporation | |
| c:\winnt\system32\inetsrv\httpext.dll | | | | | c:\winnt\system32\inetsrv\wamreg.dll | | | | | |
| rpeproxy.dll | 5.00.2195.6701 | 16.27 KB (16,656 bytes) | 11/3/2004 3:04:06 PM | Microsoft Corporation | metadata.dll | 5.00.0984 | 68.77 KB (70,416 bytes) | 3:03:23 PM | Microsoft Corporation | |
| c:\winnt\system32\rpeproxy.dll | | | | | c:\winnt\system32\inetsrv\metadata.dll | | | | | |
| fpexedll.dll | 4.0.2.7523 | 20.06 KB (20,541 bytes) | 3:03:13 PM | Microsoft Corporation | iismap.dll | 5.00.0984 | 56.27 KB (57,616 bytes) | 11/3/2004 3:02:37 PM | Microsoft Corporation | |
| c:\program files\common files\microsoft shared\web server extensions\40\bin\fpexedll.dll | | | | | c:\winnt\system32\iismap.dll | | | | | |
| md5filt.dll | 5.00.0984 | 32.77 KB (33,552 bytes) | 11/3/2004 3:03:23 PM | Microsoft Corporation | nsepm.dll | 5.00.0984 | 43.27 KB (44,304 bytes) | 11/3/2004 3:03:23 PM | Microsoft Corporation | |
| c:\winnt\system32\inetsrv\md5filt.dll | | | | | c:\winnt\system32\inetsrv\nsepm.dll | | | | | |
| gzip.dll | 5.00.0984 | 30.27 KB (30,992 bytes) | 11/3/2004 3:03:21 PM | Microsoft Corporation | admwprox.dll | 5.00.0984 | 31.77 KB (32,528 bytes) | 11/3/2004 8:55:27 AM | Microsoft Corporation | |
| c:\winnt\system32\inetsrv\gzip.dll | | | | | c:\winnt\system32\admwprox.dll | | | | | |
| compfilt.dll | 5.00.0984 | 22.77 KB (23,312 bytes) | 3:03:21 PM | Microsoft Corporation | coadmin.dll | 5.00.0984 | 39.77 KB (40,720 bytes) | 11/3/2004 3:03:21 PM | Microsoft Corporation | |
| c:\winnt\system32\inetsrv\compfilt.dll | | | | | c:\winnt\system32\inetsrv\coadmin.dll | | | | | |
| odbcint.dll | 3.520.7713.0 | 88.00 KB (90,112 bytes) | 5:27:33 PM | Microsoft Corporation | iisadmin.dll | 5.00.0984 | 15.77 KB (16,144 bytes) | 3:03:21 PM | Microsoft Corporation | |
| c:\winnt\system32\odbcint.dll | | | | | c:\winnt\system32\inetsrv\iisadmin.dll | | | | | |
| odbc32.dll | 3.520.7713.0 | 196.00 KB (200,704 bytes) | 5:27:34 PM | Microsoft Corporation | rpref.dll | 5.00.0984 | 4.27 KB (4,368 bytes) | 11/3/2004 3:03:23 PM | Microsoft Corporation | |
| c:\winnt\system32\odbc32.dll | | | | | c:\winnt\system32\inetsrv\rpref.dll | | | | | |
| ldapacx.dll | 5.00.0984 | 8.27 KB (8,464 bytes) | 3:15:03 PM | Microsoft Corporation | iisrtl.dll | 5.00.0984 | 121.27 KB (124,176 bytes) | 11/3/2004 3:02:37 PM | Microsoft Corporation | |
| c:\winnt\system32\inetsrv\ldapacx.dll | | | | | c:\winnt\system32\inetsrv\iisrtl.dll | | | | | |
| storedbx.dll | 5.00.0984 | 251.27 KB (257,296 bytes) | 3:03:26 PM | Microsoft Corporation | inetinfo.exe | 5.00.0984 | 14.27 KB (14,608 bytes) | 11/3/2004 3:03:22 PM | Microsoft Corporation | |
| c:\winnt\system32\inetsrv\storedbx.dll | | | | | c:\winnt\system32\inetsrv\inetinfo.exe | | | | | |
| ladminx.dll | 5.00.0984 | 61.27 KB (62,736 bytes) | 3:04:05 PM | Microsoft Corporation | winhttp.dll | 5.1.2600.1188 | (xpsp2.030318-2132) | 303.50 KB (310,784 bytes) | 11/3/2004 3:03:20 PM | Microsoft Corporation |
| c:\winnt\system32\inetsrv\ladminx.dll | | | | | c:\winnt\system32\winhttp.dll | | | | | |
| sspifilt.dll | 5.00.0984 | 42.77 KB (43,792 bytes) | 11/3/2004 3:03:23 PM | Microsoft Corporation | wininet.dll | 5.00.3700.6713 | 455.77 KB (466,704 bytes) | 11/3/2004 3:03:01 PM | Microsoft Corporation | |
| c:\winnt\system32\inetsrv\sspifilt.dll | | | | | c:\winnt\system32\wininet.dll | | | | | |
| iscomlog.dll | 5.00.0984 | 24.27 KB (24,848 bytes) | 3:03:22 PM | Microsoft Corporation | utildll.dll | 5.00.2195.6701 | 25.77 KB (26,384 bytes) | 11/3/2004 3:02:59 PM | Microsoft Corporation | |
| c:\winnt\system32\inetsrv\iscomlog.dll | | | | | c:\winnt\system32\utildll.dll | | | | | |
| lonsint.dll | 5.00.0984 | 11.77 KB (12,048 bytes) | 11/3/2004 3:03:22 PM | Microsoft Corporation | wsapi32.dll | 5.00.2134.1 | 14.27 KB (14,608 bytes) | 12/7/1999 7:00:00 AM | Microsoft Corporation | |
| c:\winnt\system32\inetsrv\lonsint.dll | | | | | c:\winnt\system32\wsapi32.dll | | | | | |
| inetsloc.dll | 5.00.0984 | 20.27 KB (20,752 bytes) | 11/3/2004 3:02:38 PM | Microsoft Corporation | advpack.dll | 5.00.3502.6601 | 86.77 KB (88,848 bytes) | 11/3/2004 3:02:26 PM | Microsoft Corporation | |
| c:\winnt\system32\inetsloc.dll | | | | | c:\winnt\system32\advpack.dll | | | | | |
| | | | | | wuaueng.dll | 5.4.3630.2554 | built by: lab04_n | 188.00 KB (192,512 bytes) | 11/3/2004 3:03:20 PM | Microsoft Corporation |
| | | | | | c:\winnt\system32\wuaueng.dll | | | | | |

| | | | | | | | |
|-------------------------------------|---------------------------------|---------------------------|----------------------|--------------------------------|-----------------------|---------------------------|------------------------------|
| wuau serv.dll | 5.4.3630.2554 built by: lab04_n | 9.00 KB (9,216 bytes) | | llsrv.exe | 5.00.2195.6697 | 81.77 KB (83,728 bytes) | 6/19/2003 |
| 11/3/2004 3:03:20 PM | Microsoft Corporation | | | 1:05:04 PM | Microsoft Corporation | | |
| c:\winnt\system32\wuau serv.dll | | | | c:\winnt\system32\llsrv.exe | | | |
| netui1.dll | 5.00.2134.1 | 210.27 KB (215,312 bytes) | 12/7/1999 | ipbootp.dll | 5.00.2168.1 | 33.77 KB (34,576 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | | 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\netui1.dll | | | | c:\winnt\system32\ipbootp.dll | | | |
| netui0.dll | 5.00.2195.6601 | 70.27 KB (71,952 bytes) | 11/3/2004 | cryptui.dll | 5.131.2195.6628 | 433.27 KB (443,664 bytes) | 11/3/2004 |
| 3:02:48 PM | Microsoft Corporation | | | 3:02:31 PM | Microsoft Corporation | | |
| c:\winnt\system32\netui0.dll | | | | c:\winnt\system32\cryptui.dll | | | |
| ntlanman.dll | 5.00.2195.6601 | 35.27 KB (36,112 bytes) | | rastls.dll | 5.00.2195.6680 | 98.27 KB (100,624 bytes) | 11/3/2004 |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | | 3:02:53 PM | Microsoft Corporation | | |
| c:\winnt\system32\ntlanman.dll | | | | c:\winnt\system32\rastls.dll | | | |
| wshnetbs.dll | 5.00.2134.1 | 7.77 KB (7,952 bytes) | 12/7/1999 | raschap.dll | 5.00.2195.6663 | 59.27 KB (60,688 bytes) | 11/3/2004 |
| 7:00:00 AM | Microsoft Corporation | | | 3:02:52 PM | Microsoft Corporation | | |
| c:\winnt\system32\wshnetbs.dll | | | | c:\winnt\system32\raschap.dll | | | |
| provthrd.dll | 1.50.1085.0000 | 68.07 KB (69,708 bytes) | | rasppp.dll | 5.00.2195.6626 | 194.27 KB (198,928 bytes) | 11/3/2004 |
| 11/3/2004 1:58:37 PM | Microsoft Corporation | | | 3:02:53 PM | Microsoft Corporation | | |
| c:\winnt\system32\wbem\provthrd.dll | | | | c:\winnt\system32\rasppp.dll | | | |
| ntevt.dll | 1.50.1085.0072 | 192.06 KB (196,671 bytes) | 11/3/2004 | rastapi.dll | 5.00.2195.6604 | 52.77 KB (54,032 bytes) | 12/7/1999 |
| 3:03:06 PM | Microsoft Corporation | | | 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\wbem\ntevt.dll | | | | c:\winnt\system32\rastapi.dll | | | |
| perfos.dll | 5.00.2155.1 | 21.27 KB (21,776 bytes) | 12/7/1999 | rasdlg.dll | 5.00.2195.6625 | 516.77 KB (529,168 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | | 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\perfos.dll | | | | c:\winnt\system32\rasdlg.dll | | | |
| psapi.dll | 5.00.2134.1 | 28.27 KB (28,944 bytes) | 12/7/1999 | netcfgx.dll | 5.00.2195.6604 | 534.77 KB (547,600 bytes) | 11/3/2004 |
| 7:00:00 AM | Microsoft Corporation | | | 3:02:47 PM | Microsoft Corporation | | |
| c:\winnt\system32\psapi.dll | | | | c:\winnt\system32\netcfgx.dll | | | |
| framedyn.dll | 1.50.1085.0076 | 164.07 KB (168,009 bytes) | | rasmans.dll | 5.00.2195.6696 | 149.77 KB (153,360 bytes) | |
| 11/3/2004 3:03:06 PM | Microsoft Corporation | | | 11/3/2004 3:02:52 PM | Microsoft Corporation | | |
| c:\winnt\system32\wbem\framedyn.dll | | | | c:\winnt\system32\rasmans.dll | | | |
| cimwin32.dll | 1.50.1085.0103 | 1.04 MB (1,089,637 bytes) | | wmi.dll | 5.00.2191.1 | 6.27 KB (6,416 bytes) | 12/7/1999 7:00:00 AM |
| 11/3/2004 3:03:05 PM | Microsoft Corporation | | | Microsoft Corporation | | | c:\winnt\system32\wmi.dll |
| c:\winnt\system32\wbem\cimwin32.dll | | | | netshell.dll | 5.00.2195.6604 | 466.27 KB (477,456 bytes) | 11/3/2004 |
| wbemsvcs.dll | 1.50.1085.0007 | 40.07 KB (41,036 bytes) | | 3:02:48 PM | Microsoft Corporation | | |
| 11/3/2004 3:03:07 PM | Microsoft Corporation | | | c:\winnt\system32\netshell.dll | | | |
| c:\winnt\system32\wbem\wbemsvcs.dll | | | | netman.dll | 5.00.2195.6660 | 93.27 KB (95,504 bytes) | 11/3/2004 |
| wbemess.dll | 1.50.1085.0100 | 364.09 KB (372,825 bytes) | | 3:02:48 PM | Microsoft Corporation | | |
| 11/3/2004 3:03:07 PM | Microsoft Corporation | | | c:\winnt\system32\netman.dll | | | |
| c:\winnt\system32\wbem\wbemess.dll | | | | comsvcs.dll | 2000.2.3504.0 | 1.38 MB (1,448,208 bytes) | |
| fastprox.dll | 1.50.1085.0100 | 152.10 KB (155,749 bytes) | | 11/3/2004 3:02:30 PM | Microsoft Corporation | | |
| 11/3/2004 3:03:06 PM | Microsoft Corporation | | | c:\winnt\system32\comsvcs.dll | | | |
| c:\winnt\system32\wbem\fastprox.dll | | | | ntmsdba.dll | 5.00.2195.6655 | 169.27 KB (173,328 bytes) | |
| wbemcore.dll | 1.50.1085.0100 | 632.09 KB (647,257 bytes) | | 11/3/2004 3:02:49 PM | Microsoft Corporation | | |
| 11/3/2004 3:03:06 PM | Microsoft Corporation | | | c:\winnt\system32\ntmsdba.dll | | | |
| c:\winnt\system32\wbem\wbemcore.dll | | | | sens.dll | 5.00.2195.6627 | 37.27 KB (38,160 bytes) | 11/3/2004 |
| wbemcomn.dll | 1.50.1085.0100 | 692.09 KB (708,696 bytes) | | 3:02:54 PM | Microsoft Corporation | | |
| 11/3/2004 3:03:06 PM | Microsoft Corporation | | | c:\winnt\system32\sens.dll | | | |
| c:\winnt\system32\wbem\wbemcomn.dll | | | | iashlpr.dll | 5.00.2184.1 | 33.27 KB (34,064 bytes) | 12/7/1999 |
| winmgmt.exe | 1.50.1085.0100 | 192.10 KB (196,706 bytes) | | 7:00:00 AM | Microsoft Corporation | | |
| 11/3/2004 3:03:07 PM | Microsoft Corporation | | | c:\winnt\system32\iashlpr.dll | | | |
| c:\winnt\system32\wbem\winmgmt.exe | | | | iasacct.dll | 5.00.2195.6603 | 28.27 KB (28,944 bytes) | 11/3/2004 |
| simptcp.dll | 5.00.2134.1 | 19.27 KB (19,728 bytes) | 3/15/2005 | 3:02:37 PM | Microsoft Corporation | | |
| 3:51:03 PM | Microsoft Corporation | | | c:\winnt\system32\iasacct.dll | | | |
| c:\winnt\system32\simptcp.dll | | | | iasuser.dll | 5.00.2195.6622 | 19.77 KB (20,240 bytes) | 12/7/1999 |
| tcpvcs.exe | 5.00.2134.1 | 24.77 KB (25,360 bytes) | | 7:00:00 AM | Microsoft Corporation | | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | | c:\winnt\system32\iasuser.dll | | | |
| c:\winnt\system32\tcpvcs.exe | | | | iasnap.dll | 5.00.2195.6601 | 58.77 KB (60,176 bytes) | 11/3/2004 |
| msidle.dll | 5.00.2920.0000 | 6.27 KB (6,416 bytes) | 12/7/1999 7:00:00 AM | 3:02:37 PM | Microsoft Corporation | | |
| Microsoft Corporation | | | | c:\winnt\system32\iasnap.dll | | | |
| mstask.exe | 4.71.2195.6704 | 116.77 KB (119,568 bytes) | 11/3/2004 | iaspipe.dll | 5.00.2134.1 | 41.77 KB (42,768 bytes) | 12/7/1999 |
| 3:02:46 PM | Microsoft Corporation | | | 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\mstask.exe | | | | c:\winnt\system32\iaspipe.dll | | | |
| regsvcs.exe | 5.00.2195.6701 | 66.77 KB (68,368 bytes) | 11/3/2004 | expsrv.dll | 6.0.9589 | 372.03 KB (380,957 bytes) | 11/3/2004 3:02:35 PM |
| 3:02:53 PM | Microsoft Corporation | | | Microsoft Corporation | | | c:\winnt\system32\expsrv.dll |
| c:\winnt\system32\regsvcs.exe | | | | vbajet32.dll | 6.1.9431 | 30.03 KB (30,749 bytes) | 11/3/2004 |
| llsrpc.dll | 5.00.2195.6601 | 47.77 KB (48,912 bytes) | 12/7/1999 | 3:02:59 PM | Microsoft Corporation | | |
| 7:00:00 AM | Microsoft Corporation | | | c:\winnt\system32\vbajet32.dll | | | |
| c:\winnt\system32\llsrpc.dll | | | | | | | |

| | | | |
|--|---|---------------------------|--|
| msjtes40.dll | 4.00.7328.0 | 236.27 KB (241,936 bytes) | |
| 11/3/2004 3:02:45 PM Microsoft Corporation | | | |
| c:\winnt\system32\msjtes40.dll | | | |
| oledb32r.dll | 2.70.7713.0 built by: Lab06_N(dagbuild) | 64.00 KB (65,536 bytes) | |
| 11/3/2004 5:27:34 PM Microsoft Corporation | | | |
| c:\program files\common files\system\ole db\oledb32r.dll | | | |
| comdlg32.dll | 5.00.3700.6693 | 235.77 KB (241,424 bytes) | |
| 12/7/1999 7:00:00 AM Microsoft Corporation | | | |
| c:\winnt\system32\comdlg32.dll | | | |
| msdart.dll | 2.70.7713.0 built by: Lab06_N(dagbuild) | 124.00 KB (126,976 bytes) | |
| 11/3/2004 5:27:34 PM Microsoft Corporation | | | |
| c:\winnt\system32\msdart.dll | | | |
| oledb32.dll | 2.70.7713.0 built by: Lab06_N(dagbuild) | 404.00 KB (413,696 bytes) | |
| 11/3/2004 5:27:34 PM Microsoft Corporation | | | |
| c:\program files\common files\system\ole db\oledb32.dll | | | |
| msjint40.dll | 4.00.6508.0 | 148.27 KB (151,824 bytes) | |
| 11/3/2004 3:02:45 PM Microsoft Corporation | | | |
| c:\winnt\system32\msjint40.dll | | | |
| msjter40.dll | 4.00.6508.0 | 52.27 KB (53,520 bytes) | |
| 11/3/2004 3:02:45 PM Microsoft Corporation | | | |
| c:\winnt\system32\msjter40.dll | | | |
| mswstr10.dll | 4.00.6508.0 | 600.27 KB (614,672 bytes) | |
| 11/3/2004 3:02:47 PM Microsoft Corporation | | | |
| c:\winnt\system32\mswstr10.dll | | | |
| msjet40.dll | 4.00.7328.0 | 1.44 MB (1,507,600 bytes) | 11/3/2004 3:02:44 PM Microsoft Corporation |
| c:\winnt\system32\msjet40.dll | | | |
| msjtoledb40.dll | 4.00.6807.0 | 340.27 KB (348,432 bytes) | |
| 11/3/2004 3:02:45 PM Microsoft Corporation | | | |
| c:\winnt\system32\msjtoledb40.dll | | | |
| iasrad.dll | 5.00.2195.6601 | 94.77 KB (97,040 bytes) | 11/3/2004 3:02:37 PM Microsoft Corporation |
| c:\winnt\system32\iasrad.dll | | | |
| iassam.dll | 5.00.2195.6601 | 98.27 KB (100,624 bytes) | 11/3/2004 3:02:37 PM Microsoft Corporation |
| c:\winnt\system32\iassam.dll | | | |
| iasads.dll | 5.00.2195.6601 | 73.77 KB (75,536 bytes) | 11/3/2004 3:02:37 PM Microsoft Corporation |
| c:\winnt\system32\iasads.dll | | | |
| iaspolcy.dll | 5.00.2134.1 | 25.27 KB (25,872 bytes) | |
| 12/7/1999 7:00:00 AM Microsoft Corporation | | | |
| c:\winnt\system32\iaspolcy.dll | | | |
| iassvcs.dll | 5.00.2195.6601 | 58.77 KB (60,176 bytes) | 11/3/2004 3:02:37 PM Microsoft Corporation |
| c:\winnt\system32\iassvcs.dll | | | |
| iaspdo.dll | 5.00.2195.6601 | 263.27 KB (269,584 bytes) | 11/3/2004 3:02:37 PM Microsoft Corporation |
| c:\winnt\system32\iaspdo.dll | | | |
| ntmssvc.dll | 5.00.2195.6655 | 391.77 KB (401,168 bytes) | |
| 11/3/2004 3:02:49 PM Microsoft Corporation | | | |
| c:\winnt\system32\ntmssvc.dll | | | |
| ias.dll | 5.00.2134.1 | 7.27 KB (7,440 bytes) | 12/7/1999 7:00:00 AM Microsoft Corporation |
| c:\winnt\system32\ias.dll | | | |
| es.dll | 2000.2.3504.0 | 227.77 KB (233,232 bytes) | 11/3/2004 3:02:34 PM Microsoft Corporation |
| c:\winnt\system32\es.dll | | | |
| ntmarta.dll | 5.00.2195.6666 | 100.27 KB (102,672 bytes) | 11/3/2004 3:02:49 PM Microsoft Corporation |
| c:\winnt\system32\ntmarta.dll | | | |
| db2trcapi.dll | 8.1.7.447 | 36.07 KB (36,938 bytes) | 8/15/2004 9:30:48 PM International Business Machines Corporation |
| c:\sqlib\bin\db2trcapi.dll | | | |
| db2sec.dll | 8.1.7.447 | 32.07 KB (32,840 bytes) | 8/15/2004 9:33:28 PM International Business Machines Corporation |
| c:\sqlib\bin\db2sec.dll | | | |
| db2osse_db2.dll | 8.1.7.447 | 160.08 KB (163,917 bytes) | 8/15/2004 9:33:28 PM International Business Machines Corporation |
| c:\sqlib\bin\db2osse_db2.dll | | | |
| db2dascmn.dll | 8.1.7.447 | 84.07 KB (86,087 bytes) | 8/15/2004 9:32:24 PM International Business Machines Corporation |
| c:\sqlib\bin\db2dascmn.dll | | | |
| db2install.dll | 8.1.7.447 | 28.06 KB (28,738 bytes) | 8/15/2004 9:30:48 PM International Business Machines Corporation |
| c:\sqlib\bin\db2install.dll | | | |
| db2genreg.dll | 8.1.7.447 | 156.07 KB (159,818 bytes) | 8/15/2004 9:30:46 PM International Business Machines Corporation |
| c:\sqlib\bin\db2genreg.dll | | | |
| db2g11n.dll | 8.1.7.447 | 412.06 KB (421,951 bytes) | 8/15/2004 9:30:48 PM International Business Machines Corporation |
| c:\sqlib\bin\db2g11n.dll | | | |
| db2locale.dll | 8.1.7.447 | 48.06 KB (49,217 bytes) | 8/15/2004 9:30:48 PM International Business Machines Corporation |
| c:\sqlib\bin\db2locale.dll | | | |
| db2osse.dll | 8.1.7.447 | 1.60 MB (1,675,336 bytes) | 8/15/2004 9:33:28 PM International Business Machines Corporation |
| c:\sqlib\bin\db2osse.dll | | | |
| db2app.dll | 8.1.7.447 | 6.63 MB (6,950,974 bytes) | 8/15/2004 9:32:38 PM International Business Machines Corporation |
| c:\sqlib\bin\db2app.dll | | | |
| db2sysp.dll | 8.1.7.447 | 132.06 KB (135,232 bytes) | 8/15/2004 9:33:22 PM International Business Machines Corporation |
| c:\sqlib\bin\db2sysp.dll | | | |
| db2wint.dll | 8.1.7.447 | 48.06 KB (49,215 bytes) | 8/15/2004 9:33:24 PM International Business Machines Corporation |
| c:\sqlib\bin\db2wint.dll | | | |
| db2sys.dll | 8.1.7.447 | 1.32 MB (1,384,510 bytes) | 8/15/2004 9:33:22 PM International Business Machines Corporation |
| c:\sqlib\bin\db2sys.dll | | | |
| db2sec.exe | 8.1.7.447 | 24.06 KB (24,638 bytes) | 8/15/2004 9:33:16 PM International Business Machines Corporation |
| c:\sqlib\bin\db2sec.exe | | | |
| mtxoci.dll | 2000.2.3504.0 | 103.27 KB (105,744 bytes) | 11/3/2004 3:02:47 PM Microsoft Corporation |
| c:\winnt\system32\mtxoci.dll | | | |
| resutils.dll | 5.00.2195.6702 | 39.77 KB (40,720 bytes) | 11/3/2004 3:02:53 PM Microsoft Corporation |
| c:\winnt\system32\resutils.dll | | | |
| clusapi.dll | 5.00.2195.6683 | 54.27 KB (55,568 bytes) | 11/3/2004 3:02:30 PM Microsoft Corporation |
| c:\winnt\system32\clusapi.dll | | | |
| msvcsp50.dll | 5.00.7051 | 552.50 KB (565,760 bytes) | 12/7/1999 7:00:00 AM Microsoft Corporation |
| c:\winnt\system32\msvcsp50.dll | | | |
| xolehlp.dll | 1999.9.3421.3 | 17.27 KB (17,680 bytes) | 11/3/2004 8:55:08 AM Microsoft Corporation |
| c:\winnt\system32\xolehlp.dll | | | |
| msdtclog.dll | 2000.2.3504.0 | 86.77 KB (88,848 bytes) | |
| 11/3/2004 3:02:42 PM Microsoft Corporation | | | |
| c:\winnt\system32\msdtclog.dll | | | |
| mtxclu.dll | 2000.2.3504.0 | 51.27 KB (52,496 bytes) | 11/3/2004 3:02:47 PM Microsoft Corporation |
| c:\winnt\system32\mtxclu.dll | | | |
| msdtcprx.dll | 2000.2.3504.0 | 690.77 KB (707,344 bytes) | |
| 11/3/2004 3:02:42 PM Microsoft Corporation | | | |
| c:\winnt\system32\msdtcprx.dll | | | |
| txfaux.dll | 2000.2.3504.0 | 388.27 KB (397,584 bytes) | 11/3/2004 3:02:58 PM Microsoft Corporation |
| c:\winnt\system32\txfaux.dll | | | |
| msdtctm.dll | 2000.2.3504.0 | 1.08 MB (1,131,280 bytes) | |
| 11/3/2004 3:02:42 PM Microsoft Corporation | | | |
| c:\winnt\system32\msdtctm.dll | | | |
| msdtc.exe | 1999.9.3421.3 | 6.77 KB (6,928 bytes) | 11/3/2004 8:55:08 AM Microsoft Corporation |
| c:\winnt\system32\msdtc.exe | | | |
| inetpp.dll | 5.00.2195.6707 | 65.27 KB (66,832 bytes) | 11/3/2004 3:02:38 PM Microsoft Corporation |
| c:\winnt\system32\inetpp.dll | | | |
| win32spl.dll | 5.00.2195.6681 | 94.77 KB (97,040 bytes) | |
| 12/7/1999 7:00:00 AM Microsoft Corporation | | | |
| c:\winnt\system32\win32spl.dll | | | |

| | | | | | |
|--------------------------------|-----------------------------|----------------------|--------------------------------|------------------------------|-----------|
| usbmon.dll 5.00.2195.6684 | 11.27 KB (11,536 bytes) | 11/3/2004 | scecli.dll 5.00.2195.6704 | 111.77 KB (114,448 bytes) | 11/3/2004 |
| 3:02:59 PM | Microsoft Corporation | | 3:02:54 PM | Microsoft Corporation | |
| c:\winnt\system32\usbmon.dll | | | c:\winnt\system32\scecli.dll | | |
| tcpmon.dll 5.00.2195.6659 | 40.77 KB (41,744 bytes) | 11/3/2004 | esent.dll 6.1.3940.31 | 1.08 MB (1,135,376 bytes) | 11/3/2004 |
| 3:02:58 PM | Microsoft Corporation | | 3:02:34 PM | Microsoft Corporation | |
| c:\winnt\system32\tcpmon.dll | | | c:\winnt\system32\esent.dll | | |
| pjlmon.dll 5.00.2165.1 | 12.77 KB (13,072 bytes) | 11/30/1999 | msocket.dll 5.00.2195.6603 | 62.77 KB (64,272 bytes) | |
| 6:39:36 PM | Microsoft Corporation | | 11/3/2004 3:02:47 PM | Microsoft Corporation | |
| c:\winnt\system32\pjlmon.dll | | | c:\winnt\system32\msocket.dll | | |
| cnbjmon.dll 5.00.2134.1 | 43.77 KB (44,816 bytes) | | ntdsatq.dll 5.00.2195.6620 | 31.27 KB (32,016 bytes) | 11/3/2004 |
| 11/30/1999 6:38:48 PM | Microsoft Corporation | | 3:02:49 PM | Microsoft Corporation | |
| c:\winnt\system32\cnbjmon.dll | | | c:\winnt\system32\ntdsatq.dll | | |
| localspl.dll 5.00.2195.6714 | 253.27 KB (259,344 bytes) | 12/7/1999 | ntdsa.dll 5.00.2195.6697 | 1016.27 KB (1,040,656 bytes) | 11/3/2004 |
| 7:00:00 AM | Microsoft Corporation | | 3:02:48 PM | Microsoft Corporation | |
| c:\winnt\system32\localspl.dll | | | c:\winnt\system32\ntdsa.dll | | |
| spoolss.dll 5.00.2195.6704 | 79.77 KB (81,680 bytes) | 11/3/2004 | kdcsvc.dll 5.00.2195.6627 | 144.77 KB (148,240 bytes) | 11/3/2004 |
| 8:42:04 AM | Microsoft Corporation | | 3:02:40 PM | Microsoft Corporation | |
| c:\winnt\system32\spoolss.dll | | | c:\winnt\system32\kdcsvc.dll | | |
| spoolsv.exe 5.00.2195.6659 | 44.27 KB (45,328 bytes) | | sfmapi.dll 5.00.2134.1 | 38.77 KB (39,696 bytes) | 12/7/1999 |
| 11/3/2004 8:42:04 AM | Microsoft Corporation | | 7:00:00 AM | Microsoft Corporation | |
| c:\winnt\system32\spoolsv.exe | | | c:\winnt\system32\sfmapi.dll | | |
| rasadhlp.dll 5.00.2168.1 | 7.27 KB (7,440 bytes) | 12/7/1999 | rassfm.dll 5.00.2195.6604 | 21.27 KB (21,776 bytes) | 11/3/2004 |
| 7:00:00 AM | Microsoft Corporation | | 3:02:53 PM | Microsoft Corporation | |
| c:\winnt\system32\rasadhlp.dll | | | c:\winnt\system32\rassfm.dll | | |
| winnr.dll 5.00.2160.1 | 18.77 KB (19,216 bytes) | 12/7/1999 | rsabase.dll 5.00.2195.6619 | 129.27 KB (132,368 bytes) | 6/19/2003 |
| 7:00:00 AM | Microsoft Corporation | | 1:05:04 PM | Microsoft Corporation | |
| c:\winnt\system32\winnr.dll | | | c:\winnt\system32\rsabase.dll | | |
| rpss.dll 5.00.2195.6702 | 233.77 KB (239,376 bytes) | 11/3/2004 | schannel.dll 5.00.2195.6705 | 144.27 KB (147,728 bytes) | |
| 3:02:53 PM | Microsoft Corporation | | 12/7/1999 7:00:00 AM | Microsoft Corporation | |
| c:\winnt\system32\rpss.dll | | | c:\winnt\system32\schannel.dll | | |
| svchost.exe 5.00.2134.1 | 7.77 KB (7,952 bytes) | 12/7/1999 | netlogon.dll 5.00.2195.6695 | 363.27 KB (371,984 bytes) | |
| 7:00:00 AM | Microsoft Corporation | | 11/3/2004 3:02:48 PM | Microsoft Corporation | |
| c:\winnt\system32\svchost.exe | | | c:\winnt\system32\netlogon.dll | | |
| rdpwsx.dll 5.00.2195.6697 | 97.90 KB (100,248 bytes) | 11/3/2004 | kerberos.dll 5.00.2195.6666 | 207.77 KB (212,752 bytes) | |
| 3:02:53 PM | Microsoft Corporation | | 11/3/2004 3:02:40 PM | Microsoft Corporation | |
| c:\winnt\system32\rdpwsx.dll | | | c:\winnt\system32\kerberos.dll | | |
| ntlsapi.dll 5.00.2195.6601 | 6.77 KB (6,928 bytes) | 12/7/1999 7:00:00 AM | msprivs.dll 5.00.2195.6695 | 46.00 KB (47,104 bytes) | 11/3/2004 |
| Microsoft Corporation | | | 3:02:45 PM | Microsoft Corporation | |
| c:\winnt\system32\ntlsapi.dll | | | c:\winnt\system32\msprivs.dll | | |
| mstlsapi.dll 5.00.2195.6659 | 25.77 KB (26,384 bytes) | | samsrv.dll 5.00.2195.6697 | 380.77 KB (389,904 bytes) | 12/7/1999 |
| 11/3/2004 3:02:46 PM | Microsoft Corporation | | 7:00:00 AM | Microsoft Corporation | |
| c:\winnt\system32\mstlsapi.dll | | | c:\winnt\system32\samsrv.dll | | |
| icaapi.dll 5.00.2195.6654 | 122.77 KB (125,712 bytes) | 11/3/2004 | lsasrv.dll 5.00.2195.6695 | 506.77 KB (518,928 bytes) | 12/7/1999 |
| 3:02:37 PM | Microsoft Corporation | | 7:00:00 AM | Microsoft Corporation | |
| c:\winnt\system32\icaapi.dll | | | c:\winnt\system32\lsasrv.dll | | |
| regapi.dll 5.00.2195.6602 | 35.27 KB (36,112 bytes) | 11/3/2004 | lsass.exe 5.00.2195.6695 | 32.77 KB (33,552 bytes) | 12/7/1999 |
| 3:02:53 PM | Microsoft Corporation | | 7:00:00 AM | Microsoft Corporation | |
| c:\winnt\system32\regapi.dll | | | c:\winnt\system32\lsass.exe | | |
| termsrv.exe 5.00.2195.6696 | 139.27 KB (142,608 bytes) | | rnr20.dll 5.00.2195.6603 | 35.77 KB (36,624 bytes) | 11/3/2004 |
| 11/3/2004 3:02:58 PM | Microsoft Corporation | | 3:02:53 PM | Microsoft Corporation | |
| c:\winnt\system32\termsrv.exe | | | c:\winnt\system32\rnr20.dll | | |
| dssenh.dll 5.00.2195.6612 | 143.77 KB (147,216 bytes) | 11/3/2004 | wmicore.dll 5.00.2195.6611 | 72.77 KB (74,512 bytes) | |
| 3:03:16 PM | Microsoft Corporation | | 11/3/2004 3:03:01 PM | Microsoft Corporation | |
| c:\winnt\system32\dssenh.dll | | | c:\winnt\system32\wmicore.dll | | |
| wshtcpip.dll 5.00.2195.6601 | 17.27 KB (17,680 bytes) | | xactsrv.dll 5.00.2195.6662 | 90.27 KB (92,432 bytes) | 11/3/2004 |
| 11/3/2004 3:03:01 PM | Microsoft Corporation | | 3:03:02 PM | Microsoft Corporation | |
| c:\winnt\system32\wshtcpip.dll | | | c:\winnt\system32\xactsrv.dll | | |
| msafd.dll 5.00.2195.6602 | 106.27 KB (108,816 bytes) | 11/3/2004 | browser.dll 5.00.2195.6693 | 67.27 KB (68,880 bytes) | |
| 3:02:42 PM | Microsoft Corporation | | 11/3/2004 3:02:28 PM | Microsoft Corporation | |
| c:\winnt\system32\msafd.dll | | | c:\winnt\system32\browser.dll | | |
| oakley.dll 5.00.2195.6662 | 435.77 KB (446,224 bytes) | 11/3/2004 | alrsvc.dll 5.00.2134.1 | 17.77 KB (18,192 bytes) | 12/7/1999 |
| 3:02:49 PM | Microsoft Corporation | | 7:00:00 AM | Microsoft Corporation | |
| c:\winnt\system32\oakley.dll | | | c:\winnt\system32\alrsvc.dll | | |
| mfc42u.dll 6.00.9586.0 | 988.05 KB (1,011,764 bytes) | 11/3/2004 | trkwks.dll 5.00.2195.6623 | 88.27 KB (90,384 bytes) | 11/3/2004 |
| 3:02:41 PM | Microsoft Corporation | | 3:02:58 PM | Microsoft Corporation | |
| c:\winnt\system32\mfc42u.dll | | | c:\winnt\system32\trkwks.dll | | |
| polagent.dll 5.00.2195.6655 | 109.27 KB (111,888 bytes) | | | | |
| 11/3/2004 3:02:52 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\polagent.dll | | | | | |

| | | | |
|---------------------------------|------------------------|---------------------------|----------------------|
| seclogon.dll | 5.00.2195.6707 | 16.77 KB (17,168 bytes) | |
| 11/3/2004 3:02:54 PM | Microsoft Corporation | | |
| c:\winnt\system32\seclogon.dll | | | |
| psbase.dll | 5.00.2195.6661 | 112.77 KB (115,472 bytes) | 11/3/2004 |
| 3:02:52 PM | Microsoft Corporation | | |
| c:\winnt\system32\psbase.dll | | | |
| cryptsvc.dll | 5.00.2195.6661 | 74.27 KB (76,048 bytes) | |
| 11/3/2004 3:02:31 PM | Microsoft Corporation | | |
| c:\winnt\system32\cryptsvc.dll | | | |
| cryptdll.dll | 5.00.2195.6607 | 43.27 KB (44,304 bytes) | 11/3/2004 |
| 3:02:31 PM | Microsoft Corporation | | |
| c:\winnt\system32\cryptdll.dll | | | |
| wkssvc.dll | 5.00.2195.6692 | 95.77 KB (98,064 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\wkssvc.dll | | | |
| srvsvc.dll | 5.00.2195.6697 | 81.77 KB (83,728 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\srvsvc.dll | | | |
| cfgmgr32.dll | 5.00.2134.1 | 16.77 KB (17,168 bytes) | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\cfgmgr32.dll | | | |
| dmserver.dll | 2195.6605.297.3 | 11.77 KB (12,048 bytes) | |
| 11/3/2004 3:02:33 PM | VERITAS Software Corp. | | |
| c:\winnt\system32\dmserver.dll | | | |
| lmhsvc.dll | 5.00.2195.6601 | 9.77 KB (10,000 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\lmhsvc.dll | | | |
| dnssrslvr.dll | 5.00.2195.6663 | 90.27 KB (92,432 bytes) | |
| 11/3/2004 3:02:33 PM | Microsoft Corporation | | |
| c:\winnt\system32\dnssrslvr.dll | | | |
| tapi32.dll | 5.00.2195.6664 | 123.77 KB (126,736 bytes) | 11/3/2004 |
| 3:02:58 PM | Microsoft Corporation | | |
| c:\winnt\system32\tapi32.dll | | | |
| rasman.dll | 5.00.2195.6604 | 54.77 KB (56,080 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\rasman.dll | | | |
| rasapi32.dll | 5.00.2195.6625 | 192.77 KB (197,392 bytes) | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\rasapi32.dll | | | |
| rtutils.dll | 5.00.2168.1 | 43.77 KB (44,816 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\rtutils.dll | | | |
| adslrpc.dll | 5.00.2195.6701 | 130.77 KB (133,904 bytes) | 11/3/2004 |
| 3:02:26 PM | Microsoft Corporation | | |
| c:\winnt\system32\adslrpc.dll | | | |
| activeds.dll | 5.00.2195.6601 | 177.77 KB (182,032 bytes) | |
| 11/3/2004 3:02:22 PM | Microsoft Corporation | | |
| c:\winnt\system32\activeds.dll | | | |
| mprapi.dll | 5.00.2181.1 | 79.27 KB (81,168 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\mprapi.dll | | | |
| iphlpapi.dll | 5.00.2195.6602 | 68.27 KB (69,904 bytes) | |
| 11/3/2004 3:02:38 PM | Microsoft Corporation | | |
| c:\winnt\system32\iphlpapi.dll | | | |
| icmp.dll | 5.00.2134.1 | 7.27 KB (7,440 bytes) | 12/7/1999 7:00:00 AM |
| Microsoft Corporation | | | |
| c:\winnt\system32\icmp.dll | | | |
| dhcpcsvc.dll | 5.00.2195.6685 | 90.77 KB (92,944 bytes) | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\dhcpcsvc.dll | | | |
| eventlog.dll | 5.00.2195.6716 | 46.77 KB (47,888 bytes) | |
| 11/3/2004 3:02:35 PM | Microsoft Corporation | | |
| c:\winnt\system32\eventlog.dll | | | |
| ntdsapi.dll | 5.00.2195.6666 | 56.27 KB (57,616 bytes) | 11/3/2004 |
| 3:02:48 PM | Microsoft Corporation | | |
| c:\winnt\system32\ntdsapi.dll | | | |
| scesrv.dll | 5.00.2195.6704 | 248.77 KB (254,736 bytes) | 11/3/2004 |
| 3:02:54 PM | Microsoft Corporation | | |
| c:\winnt\system32\scesrv.dll | | | |
| umpnpgmgr.dll | 5.00.2182.1 | 86.27 KB (88,336 bytes) | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\umpnpgmgr.dll | | | |
| services.exe | 5.00.2195.6700 | 87.27 KB (89,360 bytes) | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\services.exe | | | |
| clbcatq.dll | 2000.2.3504.0 | 498.27 KB (510,224 bytes) | 11/3/2004 |
| 3:02:30 PM | Microsoft Corporation | | |
| c:\winnt\system32\clbcatq.dll | | | |
| wzcsapi.dll | 5.00.2195.6604 | 29.27 KB (29,968 bytes) | |
| 11/3/2004 3:03:18 PM | Microsoft Corporation | | |
| c:\winnt\system32\wzcsapi.dll | | | |
| oleaut32.dll | 2.40.4522 | 612.27 KB (626,960 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\oleaut32.dll | | | |
| wzcdlg.dll | 5.00.2195.6604 | 51.27 KB (52,496 bytes) | 11/3/2004 |
| 3:03:18 PM | Microsoft Corporation | | |
| c:\winnt\system32\wzcdlg.dll | | | |
| cscur.dll | 5.00.2195.6705 | 237.27 KB (242,960 bytes) | 11/3/2004 |
| 3:02:31 PM | Microsoft Corporation | | |
| c:\winnt\system32\cscur.dll | | | |
| msv1_0.dll | 5.00.2195.6680 | 114.77 KB (117,520 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\msv1_0.dll | | | |
| mpr.dll | 5.00.2195.6611 | 53.77 KB (55,056 bytes) | 11/3/2004 |
| 3:02:42 PM | Microsoft Corporation | | |
| c:\winnt\system32\mpr.dll | | | |
| winspool.drv | 5.00.2195.6659 | 111.27 KB (113,936 bytes) | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\winspool.drv | | | |
| winscard.dll | 5.00.2195.6609 | 77.27 KB (79,120 bytes) | |
| 11/3/2004 3:03:01 PM | Microsoft Corporation | | |
| c:\winnt\system32\winscard.dll | | | |
| atl.dll | 3.00.9435 | 73.06 KB (74,810 bytes) | 11/3/2004 3:02:27 PM |
| Microsoft Corporation | | | |
| c:\winnt\system32\atl.dll | | | |
| certcli.dll | 5.00.2195.6619 | 132.27 KB (135,440 bytes) | 11/3/2004 |
| 3:02:29 PM | Microsoft Corporation | | |
| c:\winnt\system32\certcli.dll | | | |
| wlnotify.dll | 5.00.2195.6706 | 56.27 KB (57,616 bytes) | |
| 11/3/2004 3:03:01 PM | Microsoft Corporation | | |
| c:\winnt\system32\wlnotify.dll | | | |
| csddl.dll | 5.00.2195.6713 | 98.77 KB (101,136 bytes) | 11/3/2004 |
| 3:02:31 PM | Microsoft Corporation | | |
| c:\winnt\system32\csddl.dll | | | |
| lz32.dll | 5.00.2195.6611 | 9.77 KB (10,000 bytes) | 11/3/2004 |
| 3:02:40 PM | Microsoft Corporation | | |
| c:\winnt\system32\lz32.dll | | | |
| version.dll | 5.00.2195.6623 | 15.77 KB (16,144 bytes) | 11/3/2004 |
| 3:03:00 PM | Microsoft Corporation | | |
| c:\winnt\system32\version.dll | | | |
| rsaenh.dll | 5.00.2195.6611 | 131.77 KB (134,928 bytes) | 11/3/2004 |
| 3:03:17 PM | Microsoft Corporation | | |
| c:\winnt\system32\rsaenh.dll | | | |
| mscat32.dll | 5.131.2134.1 | 7.77 KB (7,952 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\mscat32.dll | | | |
| ole32.dll | 5.00.2195.6692 | 972.77 KB (996,112 bytes) | 11/3/2004 |
| 3:02:51 PM | Microsoft Corporation | | |
| c:\winnt\system32\ole32.dll | | | |
| imagehlp.dll | 5.00.2195.6613 | 125.77 KB (128,784 bytes) | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\imagehlp.dll | | | |
| msasn1.dll | 5.00.2195.6666 | 51.77 KB (53,008 bytes) | 12/7/1999 |
| 7:00:00 AM | Microsoft Corporation | | |
| c:\winnt\system32\msasn1.dll | | | |
| crypt32.dll | 5.131.2195.6661 | 468.27 KB (479,504 bytes) | 11/3/2004 |
| 3:02:31 PM | Microsoft Corporation | | |
| c:\winnt\system32\crypt32.dll | | | |

| | | | | | |
|--|-----------------------|---------------------------|------------|---------------|---|
| wintrust.dll | 5.131.2195.6624 | 162.27 KB (166,160 bytes) | | | |
| 11/3/2004 3:03:01 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\wintrust.dll | | | | | |
| shlwapi.dll | 5.00.3502.6601 | 282.77 KB (289,552 bytes) | 11/3/2004 | | |
| 3:02:56 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\shlwapi.dll | | | | | |
| shell32.dll | 5.00.3700.6705 | 2.27 MB (2,383,632 bytes) | 11/3/2004 | | |
| 3:02:55 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\shell32.dll | | | | | |
| msgina.dll | 5.00.2195.6669 | 326.27 KB (334,096 bytes) | 11/3/2004 | | |
| 3:02:43 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\msgina.dll | | | | | |
| comctl32.dll | 5.81 | 537.77 KB (550,672 bytes) | 12/7/1999 | | |
| 7:00:00 AM | Microsoft Corporation | | | | |
| c:\winnt\system32\comctl32.dll | | | | | |
| setupapi.dll | 5.00.2195.6622 | 556.77 KB (570,128 bytes) | | | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | | | |
| c:\winnt\system32\setupapi.dll | | | | | |
| winmm.dll | 5.00.2161.1 | 184.77 KB (189,200 bytes) | 12/7/1999 | | |
| 7:00:00 AM | Microsoft Corporation | | | | |
| c:\winnt\system32\winmm.dll | | | | | |
| winsta.dll | 5.00.2195.6701 | 38.27 KB (39,184 bytes) | 11/3/2004 | | |
| 3:03:01 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\winsta.dll | | | | | |
| wsock32.dll | 5.00.2195.6603 | 21.27 KB (21,776 bytes) | | | |
| 11/3/2004 3:03:02 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\wsock32.dll | | | | | |
| dnsapi.dll | 5.00.2195.6680 | 131.77 KB (134,928 bytes) | 11/3/2004 | | |
| 3:02:33 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\dnsapi.dll | | | | | |
| wldap32.dll | 5.00.2195.6666 | 158.27 KB (162,064 bytes) | | | |
| 11/3/2004 3:03:01 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\wldap32.dll | | | | | |
| ws2help.dll | 5.00.2134.1 | 17.77 KB (18,192 bytes) | | | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | | | |
| c:\winnt\system32\ws2help.dll | | | | | |
| ws2_32.dll | 5.00.2195.6601 | 68.27 KB (69,904 bytes) | 11/3/2004 | | |
| 3:03:01 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\ws2_32.dll | | | | | |
| samlib.dll | 5.00.2195.6666 | 48.77 KB (49,936 bytes) | 12/7/1999 | | |
| 7:00:00 AM | Microsoft Corporation | | | | |
| c:\winnt\system32\samlib.dll | | | | | |
| netrap.dll | 5.00.2134.1 | 11.27 KB (11,536 bytes) | 12/7/1999 | | |
| 7:00:00 AM | Microsoft Corporation | | | | |
| c:\winnt\system32\netrap.dll | | | | | |
| netapi32.dll | 5.00.2195.6601 | 304.27 KB (311,568 bytes) | | | |
| 11/3/2004 3:02:47 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\netapi32.dll | | | | | |
| profmap.dll | 5.00.2195.6610 | 29.27 KB (29,968 bytes) | | | |
| 11/3/2004 3:02:52 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\profmap.dll | | | | | |
| secur32.dll | 5.00.2195.6695 | 47.77 KB (48,912 bytes) | 11/3/2004 | | |
| 3:02:54 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\secur32.dll | | | | | |
| sfc.dll | 5.00.2195.6673 | 92.80 KB (95,024 bytes) | 11/3/2004 | | |
| 3:02:54 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\sfc.dll | | | | | |
| nddeapi.dll | 5.00.2195.6661 | 15.77 KB (16,144 bytes) | 11/3/2004 | | |
| 3:02:47 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\nddeapi.dll | | | | | |
| userenv.dll | 5.00.2195.6711 | 380.77 KB (389,904 bytes) | 11/3/2004 | | |
| 3:02:59 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\userenv.dll | | | | | |
| user32.dll | 5.00.2195.6688 | 393.77 KB (403,216 bytes) | 11/3/2004 | | |
| 3:02:59 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\user32.dll | | | | | |
| gdi32.dll | 5.00.2195.6660 | 228.27 KB (233,744 bytes) | 11/3/2004 | | |
| 3:02:36 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\gdi32.dll | | | | | |
| rpct4.dll | 5.00.2195.6701 | 443.77 KB (454,416 bytes) | 11/3/2004 | | |
| 3:02:53 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\rpct4.dll | | | | | |
| advapi32.dll | 5.00.2195.6710 | 378.27 KB (387,344 bytes) | | | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | | | |
| c:\winnt\system32\advapi32.dll | | | | | |
| kernel32.dll | 5.00.2195.6688 | 725.77 KB (743,184 bytes) | | | |
| 12/7/1999 7:00:00 AM | Microsoft Corporation | | | | |
| c:\winnt\system32\kernel32.dll | | | | | |
| msvcrt.dll | 6.10.9844.0 | 280.05 KB (286,773 bytes) | 6/19/2003 | | |
| 1:05:04 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\msvcrt.dll | | | | | |
| winlogon.exe | 5.00.2195.6714 | 176.77 KB (181,008 bytes) | | | |
| 11/3/2004 3:03:01 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\winlogon.exe | | | | | |
| sfcfiles.dll | 5.00.2195.6717 | 948.27 KB (971,024 bytes) | 11/3/2004 | | |
| 3:02:55 PM | Microsoft Corporation | | | | |
| c:\winnt\system32\sfcfiles.dll | | | | | |
| ntdll.dll | 5.00.2195.6685 | 480.27 KB (491,792 bytes) | 12/7/1999 | | |
| 7:00:00 AM | Microsoft Corporation | | | | |
| c:\winnt\system32\ntdll.dll | | | | | |
| smss.exe | 5.00.2195.6601 | 44.77 KB (45,840 bytes) | 12/7/1999 | | |
| 7:00:00 AM | Microsoft Corporation | | | | |
| c:\winnt\system32\smss.exe | | | | | |
| [Services] | | | | | |
| Display Name | Name | State | Start Mode | Service Type | |
| Path | Error Control | Start Name | Tag ID | | |
| Alerter | Alerter | Running | Auto | Share Process | |
| c:\winnt\system32\services.exe | Normal | LocalSystem | 0 | | |
| Application Management | AppMgmt | Stopped | Manual | Share | |
| Process c:\winnt\system32\services.exe | Normal | LocalSystem | 0 | | 0 |
| Background Intelligent Transfer Service | BITS | Stopped | Manual | | |
| Share Process c:\winnt\system32\svchost.exe -k bitsgroup | Normal | LocalSystem | 0 | | |
| Computer Browser | Browser | Running | Auto | Share Process | |
| c:\winnt\system32\services.exe | Normal | LocalSystem | 0 | | |
| Indexing Service | cisvc | Stopped | Disabled | Share Process | |
| c:\winnt\system32\cisvc.exe | Normal | LocalSystem | 0 | | |
| ClipBook | ClipSrv | Stopped | Manual | Own Process | |
| c:\winnt\system32\clipsrv.exe | Normal | LocalSystem | 0 | | |
| DB2 JDBC Applet Server | DB2JDS | Stopped | Manual | Own | |
| Process "c:\sql\lib\bin\db2jds.exe" | Normal | LocalSystem | 0 | | 0 |
| DB2 Security Server | DB2NTSECSERVER | Running | Auto | Own | |
| Process "c:\sql\lib\bin\db2sec.exe" | Normal | LocalSystem | 0 | | 0 |
| Distributed File System | Dfs | Running | Auto | Own | |
| Process c:\winnt\system32\dfssvc.exe | Normal | LocalSystem | 0 | | 0 |
| DHCP Client | Dhcp | Running | Auto | Share Process | |
| c:\winnt\system32\services.exe | Normal | LocalSystem | 0 | | |
| Logical Disk Manager Administrative Service | dmadmin | Stopped | | | |
| Manual Share Process c:\winnt\system32\dmadmin.exe /com | Normal | LocalSystem | 0 | | |
| Logical Disk Manager | dmserver | Running | Auto | Share Process | |
| c:\winnt\system32\services.exe | Normal | LocalSystem | 0 | | |
| DNS Client | Dnscache | Running | Auto | Share Process | |
| c:\winnt\system32\services.exe | Normal | LocalSystem | 0 | | |
| Event Log | Eventlog | Running | Auto | Share Process | |
| c:\winnt\system32\services.exe | Normal | LocalSystem | 0 | | |
| COM+ Event System | EventSystem | Running | Manual | Share | |
| Process c:\winnt\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 | | |
| Fax Service | Fax | Stopped | Manual | Own Process | |
| c:\winnt\system32\faxsvc.exe | Normal | LocalSystem | 0 | | |

| | | | | | | | | | |
|--|--|--------------|---------------|---------------|--|--|-------------|-------------|---|
| Internet Authentication Service | IAS | Running | Auto | Share | Process | c:\winnt\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| IIS Admin Service | IISADMIN | Running | Auto | Share | Process | c:\winnt\system32\inetrv\inetinfo.exe | Normal | LocalSystem | 0 |
| Intersite Messaging | IsmServ | Stopped | Disabled | Own Process | Process | c:\winnt\system32\ismserv.exe | Normal | LocalSystem | 0 |
| Kerberos Key Distribution Center | kdc | Stopped | Disabled | Share Process | Process | c:\winnt\system32\lsass.exe | Normal | LocalSystem | 0 |
| Server | lanmanserver | Running | Auto | Share Process | Process | c:\winnt\system32\services.exe | Normal | LocalSystem | 0 |
| Workstation | lanmanworkstation | Running | Auto | Share | Process | c:\winnt\system32\services.exe | Normal | LocalSystem | 0 |
| Site Server ILS Service | LDAPSVCX | Running | Auto | Share Process | Process | c:\winnt\system32\inetrv\inetinfo.exe | Normal | LocalSystem | 0 |
| License Logging Service | LicenseService | Running | Auto | Own Process | Process | c:\winnt\system32\llssrv.exe | Normal | LocalSystem | 0 |
| TCP/IP NetBIOS Helper Service | LmHosts | Running | Auto | Share | Process | c:\winnt\system32\services.exe | Normal | LocalSystem | 0 |
| Messenger Messenger | Stopped | Manual | Share Process | Process | c:\winnt\system32\services.exe | Normal | LocalSystem | 0 | |
| NetMeeting Remote Desktop Sharing | mnmsrv | Stopped | Manual | Own Process | Process | c:\winnt\system32\mnmsrv.exe | Normal | LocalSystem | 0 |
| Distributed Transaction Coordinator | MSDTC | Running | Auto | Own Process | Process | c:\winnt\system32\msdtc.exe | Normal | LocalSystem | 0 |
| Windows Installer | MSIServer | Stopped | Manual | Share Process | Process | c:\winnt\system32\msiexec.exe /v | Normal | LocalSystem | 0 |
| Network DDE | NetDDE | Stopped | Manual | Share Process | Process | c:\winnt\system32\netdde.exe | Normal | LocalSystem | 0 |
| Network DDE DSDMNetDDEdsdm | Stopped | Manual | Share | Process | Process | c:\winnt\system32\netdde.exe | Normal | LocalSystem | 0 |
| Net Logon Netlogon | Stopped | Manual | Share Process | Process | c:\winnt\system32\lsass.exe | Normal | LocalSystem | 0 | |
| Network Connections Netman | Running | Manual | Share Process | Process | c:\winnt\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 | |
| File Replication | NtFrs | Stopped | Manual | Own Process | Process | c:\winnt\system32\ntfrs.exe | Ignore | LocalSystem | 0 |
| NT LM Security Support Provider | NtLmSsp | Stopped | Manual | Share Process | Process | c:\winnt\system32\lsass.exe | Normal | LocalSystem | 0 |
| Removable Storage | NtmsSvc | Running | Auto | Share Process | Process | c:\winnt\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| Plug and Play | PlugPlay | Running | Auto | Share Process | Process | c:\winnt\system32\services.exe | Normal | LocalSystem | 0 |
| IPSEC Policy Agent | PolicyAgent | Running | Auto | Share | Process | c:\winnt\system32\lsass.exe | Normal | LocalSystem | 0 |
| Protected Storage | ProtectedStorage | Running | Auto | Share | Process | c:\winnt\system32\services.exe | Normal | LocalSystem | 0 |
| Remote Access Auto Connection Manager | RasAuto | Stopped | Manual | Share Process | Process | c:\winnt\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| Remote Access Connection Manager | RasMan | Running | Manual | Share Process | Process | c:\winnt\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| Routing and Remote Access | RemoteAccess | Stopped | Disabled | Share Process | Process | c:\winnt\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| Remote Registry Service | RemoteRegistry | Running | Auto | Own Process | Process | c:\winnt\system32\regsvc.exe | Normal | LocalSystem | 0 |
| Remote Procedure Call (RPC) Locator | RpcLocator | Stopped | Manual | Own Process | Process | c:\winnt\system32\locator.exe | Normal | LocalSystem | 0 |
| Remote Procedure Call (RPC) | RpcSs | Running | Auto | Share | Process | c:\winnt\system32\svchost -k rpsvc | Normal | LocalSystem | 0 |
| QoS Admission Control (RSVP) | RSVP | Running | Auto | Own | Process | c:\winnt\system32\rsvp.exe -s | Normal | LocalSystem | 0 |
| Security Accounts Manager | SamSs | Running | Auto | Share | Process | c:\winnt\system32\lsass.exe | Normal | LocalSystem | 0 |
| Smart Card Helper | SCardDrv | Stopped | Manual | Share Process | Process | c:\winnt\system32\scardsvr.exe | Ignore | LocalSystem | 0 |
| Smart Card | SCardSvr | Stopped | Manual | Share Process | Process | c:\winnt\system32\scardsvr.exe | Ignore | LocalSystem | 0 |
| Task Scheduler | Schedule | Running | Auto | Share Process | Process | c:\winnt\system32\mstask.exe | Normal | LocalSystem | 0 |
| RunAs Service | seclogon | Running | Auto | Share Process | Process | c:\winnt\system32\services.exe | Ignore | LocalSystem | 0 |
| System Event Notification | SENS | Running | Auto | Share | Process | c:\winnt\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| Internet Connection Sharing | SharedAccess | Stopped | Manual | Share Process | Process | c:\winnt\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| Simple TCP/IP Services | SimpTcp | Running | Auto | Share | Process | c:\winnt\system32\tpsvcs.exe | Normal | LocalSystem | 0 |
| Print Spooler | Spooler | Running | Auto | Own Process | Process | c:\winnt\system32\spoolsv.exe | Normal | LocalSystem | 0 |
| Performance Logs and Alerts | SysmonLog | Stopped | Manual | Own Process | Process | c:\winnt\system32\smlogsvc.exe | Normal | LocalSystem | 0 |
| Telephony TapiSrv | Running | Manual | Share Process | Process | c:\winnt\system32\svchost.exe -k tapisrv | Normal | LocalSystem | 0 | |
| Terminal Services | TermService | Running | Auto | Own | Process | c:\winnt\system32\termsrv.exe | Normal | LocalSystem | 0 |
| Telnet | TlntSvr | Stopped | Manual | Own Process | Process | c:\winnt\system32\tlntsvr.exe | Normal | LocalSystem | 0 |
| Distributed Link Tracking Server | TrkSvr | Stopped | Manual | Share | Process | c:\winnt\system32\services.exe | Normal | LocalSystem | 0 |
| Distributed Link Tracking Client | TrkWks | Running | Auto | Share | Process | c:\winnt\system32\services.exe | Normal | LocalSystem | 0 |
| Uninterruptible Power Supply | UPS | Stopped | Manual | Own | Process | c:\winnt\system32\ups.exe | Normal | LocalSystem | 0 |
| Utility Manager | UtilMan | Stopped | Manual | Own Process | Process | c:\winnt\system32\utilman.exe | Normal | LocalSystem | 0 |
| Windows Time | W32Time | Stopped | Manual | Share Process | Process | c:\winnt\system32\services.exe | Normal | LocalSystem | 0 |
| World Wide Web Publishing Service | W3SVC | Running | Auto | Share Process | Process | c:\winnt\system32\inetrv\inetinfo.exe | Normal | LocalSystem | 0 |
| Windows Management Instrumentation | WinMgmt | Running | Auto | Own Process | Process | c:\winnt\system32\wbem\winmgmt.exe | Ignore | LocalSystem | 0 |
| Windows Management Instrumentation Driver Extensions | Wmi | Running | Manual | Share Process | Process | c:\winnt\system32\services.exe | Normal | LocalSystem | 0 |
| Automatic Updates | wuauerv | Running | Auto | Share Process | Process | c:\winnt\system32\svchost.exe -k wugroup | Normal | LocalSystem | 0 |
| Wireless Configuration | WZCSVC | Stopped | Manual | Share | Process | c:\winnt\system32\svchost.exe -k netsvcs | Normal | LocalSystem | 0 |
| [Program Groups] | | | | | | | | | |
| Group Name | Name | User Name | | | | | | | |
| Accessories | Default User:Accessories | Default User | | | | | | | |
| Accessories\Accessibility | Default User:Accessories\Accessibility | Default User | | | | | | | |

Accessories\Entertainment Default User:Accessories\Entertainment Default User
 Accessories\System Tools Default User:Accessories\System Tools Default User
 Startup Default User:Startup Default User
 Accessories All Users:Accessories All Users
 Accessories\Communications All Users:Accessories\Communications All Users
 Accessories\Entertainment All Users:Accessories\Entertainment All Users
 Accessories\Microsoft Script Debugger All Users:Accessories\Microsoft Script Debugger All Users
 Accessories\System Tools All Users:Accessories\System Tools All Users
 Administrative Tools All Users:Administrative Tools All Users
 IBM DB2 All Users:IBM DB2 All Users
 IBM DB2\Command Line Tools All Users:IBM DB2\Command Line Tools All Users
 Microsoft Visual C++ 6.0 All Users:Microsoft Visual C++ 6.0 All Users
 Microsoft Visual C++ 6.0\Microsoft Visual C++ 6.0 Tools All Users:Microsoft Visual C++ 6.0\Microsoft Visual C++ 6.0 Tools All Users
 Startup All Users:Startup All Users
 Accessories CLIENT20\Administrator:Accessories
 CLIENT20\Administrator
 Accessories\Accessibility CLIENT20\Administrator:Accessories\Accessibility
 CLIENT20\Administrator
 Accessories\Entertainment CLIENT20\Administrator:Accessories\Entertainment
 CLIENT20\Administrator
 Accessories\System Tools CLIENT20\Administrator:Accessories\System Tools
 CLIENT20\Administrator
 Administrative Tools CLIENT20\Administrator:Administrative Tools
 CLIENT20\Administrator
 Startup CLIENT20\Administrator:Startup
 CLIENT20\Administrator

[Startup Programs]

| Program | Command | User Name | Location |
|----------|--------------|------------------------|----------|
| synctime | synctime.bat | CLIENT20\Administrator | Startup |

[OLE Registration]

| Object | Local Server |
|---|---|
| Sound (OLE2) | sndrec32.exe |
| Media Clip | mplay32.exe |
| Video Clip | mplay32.exe /avi |
| MIDI Sequence | mplay32.exe /mid |
| Sound | Not Available |
| Media Clip | Not Available |
| Image Document | "C:\Program Files\Windows NT\Accessories\ImageVue\KodakImg.exe" |
| WordPad Document | "%ProgramFiles%\Windows NT\Accessories\WORDPAD.EXE" |
| Windows Media Services DRM Storage object | Not Available |
| Bitmap Image | mspaint.exe |

[Internet Explorer 5]

[Following are sub-categories of this main category]

[Summary]

| Item | Value |
|---------|---------------|
| Version | 5.0.3700.1000 |
| Build | 53700.1000 |

Product ID 51876-270-4893362-05373
 Application Path C:\Program Files\Internet Explorer
 Language English (United States)
 Active Printer Not Available

Cipher Strength 168-bit
 Content Advisor Disabled
 IEAK Install No

[File Versions]

| File | Version | Size | Date | Path | Company |
|--------------|-----------------|---------------|-----------------------|------------------------------------|-----------------------|
| advapi32.dll | 5.0.2195.6710 | 378 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| advpack.dll | 5.0.3502.6601 | 87 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| browsecl.dll | 5.0.3700.6661 | 35 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| browseui.dll | 5.0.3700.6661 | 789 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| ckcnv.exe | 5.0.2189.1 | 9 KB | 12/7/1999 7:00:00 AM | C:\WINNT\system32 | Microsoft Corporation |
| comctl32.dll | 5.81.3502.6601 | 538 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| crypt32.dll | 5.131.2195.6661 | 468 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| ehsig.dll | <File Missing> | Not Available | Not Available | Not Available | Not Available |
| iemigrat.dll | <File Missing> | Not Available | Not Available | Not Available | Not Available |
| iesetup.dll | 5.0.3502.6601 | 57 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| iexplore.exe | 5.0.2920.0 | 59 KB | 12/7/1999 7:00:00 AM | C:\Program Files\Internet Explorer | Microsoft Corporation |
| imagehlp.dll | 5.0.2195.6613 | 126 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| imghelp.dll | <File Missing> | Not Available | Not Available | Not Available | Not Available |
| inseng.dll | 5.0.3502.6601 | 72 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| jobexec.dll | 5.0.0.1 | 47 KB | 12/7/1999 7:00:00 AM | C:\WINNT\system32 | Microsoft Corporation |
| jscrip.dll | 5.1.0.8513 | 476 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| jsproxy.dll | 5.0.2920.0 | 13 KB | 12/7/1999 7:00:00 AM | C:\WINNT\system32 | Microsoft Corporation |
| msahtml.dll | <File Missing> | Not Available | Not Available | Not Available | Not Available |
| mshtml.dll | 5.0.3700.6699 | 2299 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| msoss.dll | <File Missing> | Not Available | Not Available | Not Available | Not Available |
| msxml.dll | 8.0.6730.0 | 502 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| occache.dll | 5.0.3502.6601 | 86 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| ole32.dll | 5.0.2195.6692 | 973 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| oleaut32.dll | 2.40.4522.0 | 612 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| olepro32.dll | 5.0.4522.0 | 160 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| rsabase.dll | 5.0.2195.6619 | 129 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| rsaenh.dll | 5.0.2195.6611 | 132 KB | 6/19/2003 12:05:04 PM | C:\WINNT\system32 | Microsoft Corporation |
| rsapi32.dll | <File Missing> | Not Available | Not Available | Not Available | Not Available |
| | | Not Available | Not Available | Not Available | Not Available |

| | | | |
|-------------------|-----------------------|---------------|-----------------------|
| rsasig.dll | <File Missing> | Not Available | Not Available |
| Not Available | Not Available | | |
| schannel.dll | 5.1.2195.6705 | 144 KB | 6/19/2003 12:05:04 PM |
| C:\WINNT\system32 | Microsoft Corporation | | |
| shdoc401.dll | <File Missing> | Not Available | Not Available |
| Available | Not Available | Not Available | |
| shdocvw.dll | 5.0.3700.6668 | 1082 KB | 6/19/2003 12:05:04 PM |
| C:\WINNT\system32 | Microsoft Corporation | | |
| shell32.dll | 5.0.3700.6705 | 2328 KB | 6/19/2003 12:05:04 PM |
| C:\WINNT\system32 | Microsoft Corporation | | |
| shlwapi.dll | 5.0.3502.6601 | 283 KB | 6/19/2003 12:05:04 PM |
| C:\WINNT\system32 | Microsoft Corporation | | |
| url.dll | 5.0.3502.6601 | 82 KB | 6/19/2003 12:05:04 PM |
| C:\WINNT\system32 | Microsoft Corporation | | |
| urlmon.dll | 5.0.3700.6705 | 443 KB | 6/19/2003 12:05:04 PM |
| C:\WINNT\system32 | Microsoft Corporation | | |
| vbscript.dll | 5.1.0.7426 | 428 KB | 6/19/2003 12:05:04 PM |
| C:\WINNT\system32 | Microsoft Corporation | | |
| webcheck.dll | 5.0.3502.6601 | 252 KB | 6/19/2003 12:05:04 PM |
| C:\WINNT\system32 | Microsoft Corporation | | |
| win.com | 5.0.2134.1 | 24 KB | 12/7/1999 7:00:00 AM |
| C:\WINNT\system32 | Microsoft Corporation | | |
| wininet.dll | 5.0.3700.6713 | 456 KB | 6/19/2003 12:05:04 PM |
| C:\WINNT\system32 | Microsoft Corporation | | |
| winsock.dll | 3.10.0.103 | 3 KB | 12/7/1999 7:00:00 AM |
| C:\WINNT\system32 | Microsoft Corporation | | |
| wintrust.dll | 5.131.2195.6624 | 162 KB | 6/19/2003 12:05:04 PM |
| C:\WINNT\system32 | Microsoft Corporation | | |
| wsock.vxd | <File Missing> | Not Available | Not Available |
| Not Available | Not Available | | |
| wsock32.dll | 5.0.2195.6603 | 21 KB | 6/19/2003 12:05:04 PM |
| C:\WINNT\system32 | Microsoft Corporation | | |
| wsock32n.dll | <File Missing> | Not Available | Not Available |
| Available | Not Available | Not Available | |

[Connectivity]

| | |
|-----------------------|------------|
| Item | Value |
| Connection Preference | Never dial |
| EnableHttp1.1 | 1 |
| ProxyHttp1.1 | 0 |

LAN Settings

| | |
|---------------------|-------------|
| AutoConfigProxy | wininet.dll |
| AutoProxyDetectMode | Disabled |
| AutoConfigURL | |
| Proxy | Disabled |
| ProxyServer | |
| ProxyOverride | |

[Cache]

[Following are sub-categories of this main category]

[Summary]

| | |
|---------------------------------|---|
| Item | Value |
| Page Refresh Type | Automatic |
| Temporary Internet Files Folder | C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files |
| Total Disk Space | 34710 MB |
| Available Disk Space | 30664 MB |
| Maximum Cache Size | 1084 MB |
| Available Cache Size | 1084 MB |

[List of Objects]

| | | |
|--|--------|----------|
| Program File | Status | CodeBase |
| No cached object information available | | |

[Content]

[Following are sub-categories of this main category]

[Summary]

| | |
|-----------------|----------|
| Item | Value |
| Content Advisor | Disabled |

[Personal Certificates]

| | | | |
|---------------|---------------|-------------------------|---------------------|
| Issued To | Issued By | Validity | Signature Algorithm |
| Administrator | Administrator | 11/3/2004 to 10/10/2104 | sha1RSA |

[Other People Certificates]

| | | | |
|---|-----------|----------|---------------------|
| Issued To | Issued By | Validity | Signature Algorithm |
| No other people certificate information available | | | |

[Publishers]

| | |
|------------------------------------|--|
| Name | |
| No publisher information available | |

[Security]

| | |
|------------------|----------------|
| Zone | Security Level |
| Local intranet | Medium-low |
| Trusted sites | Low |
| Internet | Medium |
| Restricted sites | High |

Client Configuration Parameters

Client Configuration Parameters

COM+ Settings

tpccCom.tpcc_com.1:

Activation:

- Enable Object Pooling selected
- Minimum Pool Size: 50
- Maximum Pool Size: 50
- Creating Timeout: 60,000
- Enable Just in Time Activation

Concurrency:

- Concurrency Required

Microsoft Windows 2000 Client Registry Parameters

TPCC Application Registry Parameters

```
[HKEY_LOCAL_MACHINE\SOFTWARE\TPCC]
"dbType"="DB2"
"dIvyLogPath"="c:\inetpub\wwwroot\tpcc\dIvy"
"dIvyQueueLen"=dword:00004e20
>nullIDB"=dword:00000000
"dbName"="tpcc"
"errorLogFile"="c:\inetpub\wwwroot\tpcc\errorLog.txt"
"htmlTraceLogFile"="c:\inetpub\wwwroot\tpcc\htmlTrace.txt"
```

```

"numUsers"=dword:00007530
"dbUserName"="Administrator"
"dbPassword"="tpcc"
"dbInterfacePath"="C:\\inetpub\\wwwroot\\tpcc\\db2glue.dll"
"dlvyThreads"=dword:0000000a

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\InetInfo]

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\InetInfo\\Parameters]
"ListenBackLog"=dword:00000096
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,4c,00,\\
44,00,41,00,50,00,53,00,56,00,43,00,58,00,00,00,00,00
"PoolThreadLimit"=dword:0000000e
"ThreadTimeout"=dword:00015180

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\InetInfo\\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000842
"Last Help"=dword:00000843
"First Counter"=dword:00000802
"First Help"=dword:00000803
"Library Validation Code"=hex:bc,b6,7c,11,e0,c1,c4,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,a0,38,ed,84,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\IISADMIN]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,00,4e,00,54,00,5c,00,53,00,\\
79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00,65,00,74,00,73,\\
00,72,00,76,00,5c,00,69,00,6e,00,65,00,74,00,69,00,6e,00,66,00,6f,00,2e,00,\\
65,00,78,00,65,00,00,00
"DisplayName"="IIS Admin Service"
"DependOnService"=hex(7):52,00,50,00,43,00,53,00,53,00,00,00,50,00,72,00,6f,00,\\
74,00,65,00,63,00,74,00,65,00,64,00,53,00,74,00,6f,00,72,00,61,00,67,00,65,\\
00,00,00,00,00
"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"
"Description"="Allows administration of Web and FTP services through the Internet Information Services snap-in."
"FailureCommand"="\"C:\\WINNT\\System32\\iisreset.exe\" /fail=%1%"
"FailureActions"=hex:80,51,01,00,88,ca,0a,00,98,ca,0a,00,03,00,00,00,f4,ca,0a,\\
00,03,00,00,00,01,00,00,00,03,00,00,00,01,00,00,00,03,00,00,00,01,00,00,00

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\IISADMIN\\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\\
00,00,02,00,70,00,04,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\\
05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,05,\\
20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\\
00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\\
00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00,00,00,00,05,12,00,00,

```

```

00,01,01,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\IISADMIN\\Enum]
"0"="Root\\LEGACY_IISADMIN\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,00,4e,00,54,00,5c,00,53,00,\\
79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00,65,00,74,00,73,\\
00,72,00,76,00,5c,00,69,00,6e,00,65,00,74,00,69,00,6e,00,66,00,6f,00,2e,00,\\
65,00,78,00,65,00,00,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):49,00,49,00,53,00,41,00,44,00,4d,00,49,00,4e,00,00,00,\\
00,00
"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"
"Description"="Provides Web connectivity and administration through the Internet Information Services snap-in."

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\ASP]
"NOTE"="This is for backward compatibility only."

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\ASPLanguageEngines]

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\ASPLanguageEngines\\PerlScript]
"Write"="$Response->write();"
"WriteBlock"="$Response->writeblock();"

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\ASP\\Parameters]

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters]
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\\WINNT\\System32\\inetsrv"
"CertMapList"="C:\\WINNT\\System32\\inetsrv\\iisrmap.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\\WINNT\\System32\\LogFiles"
"AcceptExOutstanding"=dword:00000028

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\ADCLaunch]

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\ADCLaunch\\AdvancedDataFactory]

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\ADCLaunch\\RDSServer.DataFactory]

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\Script Map]

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\Virtual Roots]
"/"="c:\\inetpub\\wwwroot,207"
"/Scripts"="c:\\inetpub\\scripts,1"

```

```

"/IISHelp"="c:\\winnt\\help\\iishelp,,1"
"/IISAdmin"="C:\\WINNT\\System32\\inetstrv\\iisadmin,,1"
"/IISSamples"="c:\\inetpub\\iissamples,,1"
"/MSADC"="c:\\program files\\common files\\system\\msadc,,1"
"/_vti_bin"="C:\\Program Files\\Common Files\\Microsoft Shared\\Web Server
Extensions\\40\\isapi,,1"
"/Printers"="C:\\WINNT\\web\\printers,,201"
"/tpcc"="c:\\inetpub\\wwwroot\\tpcc,,207"
"/Rpc"="C:\\WINNT\\System32\\RpcProxy,,4"

```

```

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\P
erformance]
"Library"="w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:000008e6
"Last Help"=dword:000008e7
"First Counter"=dword:00000844
"First Help"=dword:00000845
"Library Validation
Code"=hex:ba,71,6e,c7,ac,c1,c4,01,10,3d,00,00,00,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,a0,38,ed,84,36,c3,01
"WbemAdapFileSize"=dword:00001d10
"WbemAdapStatus"=dword:00000000

```

```

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\S
ecurity]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,0
2,\\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\\
00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\\
05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\\
20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\\
00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\\
00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00,00,00,00,05,12,00,00,\\
00,01,01,00,00,00,00,00,05,12,00,00,00

```

```

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\E
num]
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

```

RTE Input Parameters

IBM BenchMaster benchmark profile. DO NOT CHANGE THE SPACING IN THIS FILE!

30 ** Number of slaves (all must be defined directly below)

| SEGMENT | MACHINE | LOG DIRECTORY | ODBC |
|-----------|-----------|---------------|-------------------------|
| WEBSERVER | DB SERVER | STARTWH | ENDWH |
| #USERS | | | |
| v51 | vrte150 | c:\\rtelogs | tpcc |
| 400 | 4000 | client1501 | db2serv1_tcp |
| v52 | vrte150 | c:\\rtelogs | tpcc |
| 401 | 800 | 4000 | client1502 db2serv1_tcp |
| v53 | vrte150 | c:\\rtelogs | tpcc |
| 801 | 1200 | 4000 | client1503 db2serv1_tcp |
| v54 | vrte150 | c:\\rtelogs | tpcc |
| 1201 | 1600 | 4000 | client1504 db2serv1_tcp |
| v55 | vrte150 | c:\\rtelogs | tpcc |
| 1601 | 2000 | 4000 | client1505 db2serv1_tcp |
| v61 | vrte160 | c:\\rtelogs | tpcc |
| 2001 | 2400 | 4000 | client1601 db2serv1_tcp |

| | | | | | |
|-------|---------|-------------|------|------------|--------------|
| v62 | vrte160 | c:\\rtelogs | tpcc | client1602 | db2serv1_tcp |
| 2401 | 2800 | 4000 | | | |
| v63 | vrte160 | c:\\rtelogs | tpcc | client1603 | db2serv1_tcp |
| 2801 | 3200 | 4000 | | | |
| v64 | vrte160 | c:\\rtelogs | tpcc | client1604 | db2serv1_tcp |
| 3201 | 3600 | 4000 | | | |
| v65 | vrte160 | c:\\rtelogs | tpcc | client1605 | db2serv1_tcp |
| 3601 | 4000 | 4000 | | | |
| v41 | rte40 | c:\\rtelogs | tpcc | client401 | db2serv1_tcp |
| 4001 | 4400 | 4000 | | | |
| v42 | rte40 | c:\\rtelogs | tpcc | client402 | db2serv1_tcp |
| 4401 | 4800 | 4000 | | | |
| v43 | rte40 | c:\\rtelogs | tpcc | client403 | db2serv1_tcp |
| 4801 | 5200 | 4000 | | | |
| v44 | rte40 | c:\\rtelogs | tpcc | client404 | db2serv1_tcp |
| 5201 | 5600 | 4000 | | | |
| v45 | rte40 | c:\\rtelogs | tpcc | client405 | db2serv1_tcp |
| 5601 | 6000 | 4000 | | | |
| v21 | rte20 | c:\\rtelogs | tpcc | client201 | db2serv1_tcp |
| 6001 | 6400 | 4000 | | | |
| v22 | rte20 | c:\\rtelogs | tpcc | client202 | db2serv1_tcp |
| 6401 | 6800 | 4000 | | | |
| v23 | rte20 | c:\\rtelogs | tpcc | client203 | db2serv1_tcp |
| 6801 | 7200 | 4000 | | | |
| v24 | rte20 | c:\\rtelogs | tpcc | client204 | db2serv1_tcp |
| 7201 | 7600 | 4000 | | | |
| v25 | rte20 | c:\\rtelogs | tpcc | client205 | db2serv1_tcp |
| 7601 | 8000 | 4000 | | | |
| v31 | rte30 | c:\\rtelogs | tpcc | client301 | db2serv1_tcp |
| 8001 | 8400 | 4000 | | | |
| v32 | rte30 | c:\\rtelogs | tpcc | client302 | db2serv1_tcp |
| 8401 | 8800 | 4000 | | | |
| v33 | rte30 | c:\\rtelogs | tpcc | client303 | db2serv1_tcp |
| 8801 | 9200 | 4000 | | | |
| v34 | rte30 | c:\\rtelogs | tpcc | client304 | db2serv1_tcp |
| 9201 | 9600 | 4000 | | | |
| v35 | rte30 | c:\\rtelogs | tpcc | client305 | db2serv1_tcp |
| 9601 | 10000 | 4000 | | | |
| v11 | rte10 | c:\\rtelogs | tpcc | client101 | db2serv1_tcp |
| 10001 | 10400 | 4000 | | | |
| v12 | rte10 | c:\\rtelogs | tpcc | client102 | db2serv1_tcp |
| 10401 | 10800 | 4000 | | | |
| v13 | rte10 | c:\\rtelogs | tpcc | client103 | db2serv1_tcp |
| 10801 | 11200 | 4000 | | | |
| v14 | rte10 | c:\\rtelogs | tpcc | client104 | db2serv1_tcp |
| 11201 | 11600 | 4000 | | | |
| v15 | rte10 | c:\\rtelogs | tpcc | client105 | db2serv1_tcp |
| 11601 | 12000 | 4000 | | | |

1000 ** Connect rate - rate users log in to the database (users per minute)

300 ** Run rate- rate users ramp in (users per minute)

0 *** Ramp-in type (0 = linear, 1 = 5 step descending rate)"

1 *** Web client (ignored for 2-tier; 0 = Microsoft's web client, 1 = IBM pSeries web client)"

12000 ** Total number of warehouses

0 *** Run type (0 = 3-tier, 1 = 2-tier)"

173 *** C_LOAD (0-255) - NURAND ""C"" value that WAS used for customer last name generation during database LOAD, usually 123 for SQL Server"

88 *** C_RUN (0-255) - NURAND ""C"" value to be used for customer last name generation when running. abs(C_LOAD - C_RUN) must be

65 to 119, inclusive, but not 96 or 112. TIP: Set C_RUN so abs(C_LOAD - C_RUN) is 85."

208 *** C_C_ID (0-1023) - NURAND ""C"" value to be used for customer ID generation when running"

208 *** C_OL_I_ID (0-8191) - NURAND ""C"" value to be used for orderline item ID generation when running"

administrator ** Database user name

tpcc ** Database password

| TOTAL | NEWORDPAYMENT | DELI | STCKLVLORDSTAT |
|-------|------------------|------|----------------|
| 0 | 44950 43020 4010 | 4010 | 4010 *** |

Transaction mix percentages (must add to 100,000)"

| | | | |
|---|------------------|------|-----------------|
| 0 | 12030 12030 5030 | 5030 | 10030 ** 3-tier |
|---|------------------|------|-----------------|

think times (milliseconds)

| | | | |
|---|-------|---|-------------|
| 0 | 0 0 0 | 0 | 0 ** 2-tier |
|---|-------|---|-------------|

think times (milliseconds)

| | | | |
|---|-----------------|------|--------------------|
| 0 | 18000 3000 2000 | 2000 | 2000 ** 3-tier key |
|---|-----------------|------|--------------------|

times (milliseconds)

| | | | |
|---|-------|---|-----------------|
| 0 | 0 0 0 | 0 | 0 ** 2-tier key |
|---|-------|---|-----------------|

times (milliseconds)

| | | | |
|---|----------------|-------|--------------|
| 0 | 5000 5000 5000 | 20000 | 5000 ** 90th |
|---|----------------|-------|--------------|

percentile values (milliseconds)

| | | | |
|---|-------------|-----|----------------|
| 0 | 100 100 100 | 100 | 100 ** Browser |
|---|-------------|-----|----------------|

painting menu delay (milliseconds)

| | | | |
|---|-------------|-----|----------------|
| 0 | 100 100 100 | 100 | 100 ** Browser |
|---|-------------|-----|----------------|

painting response time delay (milliseconds)

2000 ** 90th percentile value for menu transactions (milliseconds)

Appendix D: 60-Day Space

| 60-Day Space Computation | | | | | | |
|--|----------------------|-------------------|----------------------|-----------------------------------|----------------------|------------|
| All data sizes in MB unless otherwise stated | | | | | | |
| Warehouses | 12,600 | | | | | |
| Measured TpmC | 150,704.91 | | | | | |
| Table | Rows | Table | Index | 5% Space | Total Space | |
| Warehouse | 12,600 | 2 | 0 | 0 | 2 | |
| District | 126,000 | 15 | 0 | 1 | 15 | |
| Item | 100,000 | 11 | 0 | 1 | 11 | |
| Stock | 1,260,000,000 | 410,186 | 0 | 20,509 | 430,695 | |
| Customer | 378,000,000 | 295,344 | 18,242 | 15,679 | 329,265 | |
| New-Order | 113,400,000 | 8,764 | 0 | 0 | 8,764 | |
| Orders | 378,000,000 | 13,876 | 10,562 | 0 | 24,438 | |
| Order-Line | 5,670,000,000 | 372,400 | 0 | 0 | 372,400 | |
| History | 378,000,000 | 23,272 | 0 | 0 | 23,272 | |
| Free Space | 77,436 | | | <u>30 Minute log Computations</u> | | |
| Dynamic Space | 409,548 | | | Log Written (KB) | | 12,505,344 |
| Static Space | 779,315 | | | Total New-Order Txns | | 4,521,147 |
| Daily Growth | 78,376 | | | Log Written per New-Order (KB) | | 2.77 |
| Daily Spread | 0 | | | | | |
| Data Storage Requirement | | | | | | |
| 60 Days (MB) | 5,481,858 | | | | | |
| 60 Days (GB) | 5,353 | | | | | |
| Log Storage Requirement | | | | | | |
| 8 Hours (GB) | 190.82 | | | | | |
| Disk Sizing | | | | | | |
| | Formatted | SUT | | Priced | | |
| Disk Type | Capacity (GB) | # of Disks | Capacity (GB) | # of Disks | Capacity (GB) | |
| DB FastT RAID 0 (36GB disks) | 33.90 | 336 | 11,390 | 336 | 11,390 | |
| LOG FastT RAID 10 (36GB disks) | 16.95 | 14 | 237 | 14 | 237 | |
| OS Fast RAID 0 (36GB disks) | 33.90 | 1 | 34 | 1 | 34 | |
| Total Capacity | | | | | 11,662 | |

Appendix E: Third-Party Quotations

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
<http://www.microsoft.com/>

Microsoft

March 23, 2005

IBM Corporation
Chris King
3079 Cornwallis Road
Durham, NC 27709

Ms. King:

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 |
|-------------|---|------------|----------|---------|
| P72-00264 | Windows Server 2003, Enterprise x64 Edition <i>Server License Only - No CALs</i> <i>Discount Schedule: Open Program - No Level</i> <i>Unit Price reflects a 40% discount from the retail unit price of \$3,999.</i> | \$2,399 | 1 | \$2,399 |
| C11-00821 | Windows 2000 Server <i>Server License Only - No CALs</i> <i>Discount Schedule: Open Program - No Level</i> <i>Unit Price reflects a 8% discount from the retail unit price of \$799.</i> | \$738 | 6 | \$4,428 |
| 254-00170 | Visual C++ Standard Edition <i>Discount Schedule: No Discounts Applied</i> | \$109 | 1 | \$109 |
| | Microsoft Problem Resolution Services <i>Professional Support</i> <i>(1 incident)</i> | \$245 | 1 | \$245 |

Some products may not be currently orderable but will be available through Microsoft's normal distribution channels by May 6, 2005.

This quote is valid for the next 90 days.

If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or jamiere@microsoft.com.

Reference ID: PCchki0402124827.

Please include this Reference ID in any correspondence regarding this price quote.

Newegg Gift Certificates My Account Help 10 Item(s) | \$116.19 [| Login |](#)

[CTS](#)
[Computer Hardware](#)
[Software](#)
[Notebook / Desktop PC](#)
[Digital Cameras](#)
[Networking Products](#)
[Mac Products](#)
[Home Electronics](#)
[Accessories](#)
[Refurbished / Clearance](#)

[Shop By Category](#) | [Shop By Brand](#) | [Top 10 Sellers](#) | [New Products](#) | [Special Sales](#) | [Rebate Sales](#) |

SEARCH: [ADVANCED SEARCH](#)



Your Shopping Cart

| Qty | Product Description | | Unit Price |
|---|---|--|----------------|
| Network - Cables | | | |
| <input type="text" value="9"/> <input type="button" value="UPDATE"/> | Arrow Micro 14ft Cat 5E Blue Cable, Model "CC5E-B14B" Item# N82E16812105305 Status: In Stock | Save to Buy Later Remove Item | \$1.80 |
| Network - Switches | | | |
| <input type="text" value="1"/> <input type="button" value="UPDATE"/> | NETGEAR 8-Port 10/100/1000Mbps Copper Gigabit Switch, Model GS108 - Retail Item# N82E16833122111 Status: In Stock | Save to Buy Later Remove Item | \$99.99 |

****This item is warranted through the product manufacturer only.**

\$-10.00 Mail-In Rebate
Purchase this Item and Receive a \$-10.00 rebate by mail.
[Rebate form may be downloaded here.](#)

Product Price: \$116.19
Shipping & Handling: \$0.00

Enter Promotion Code:

Total (Before Tax): \$116.19

[PRINT VERSION](#)
[UPDATE CART](#)
[SAVE CART TO WISH LIST](#)
[CLEAR SHOPPING CART](#)
[CONTINUE](#)

HOW TO CALCULATE SHIPPING:
 SELECT SHIPPING METHOD AND CHOOSE STATE, THEN CLICK "CALCULATE".
HOW NEWEGG.COM DETERMINES SHIPPING CHARGES.
NOTE: ANY COMBO DISCOUNT WILL NOT BE SHOWN UNTIL YOU CHECK OUT.
 TAX INSTRUCTIONS FOR RESELLERS [CLICK HERE](#).

Important Shipping Instructions **All Orders Must Be Placed by 11:59 PM PT**

- All orders shipped within **CA, TN, PR and NJ** will be charged sales tax.
- FedEx express saver: 3-4 business days plus processing time.