

TPC Benchmark™ C
Full Disclosure Report
for
IBM® eServer™ xSeries® 460
using
IBM DB2® UDB 8.2
and
Microsoft® Windows® Server 2003
Enterprise x64 Edition

TPC-C Version 5.6

Submitted for Review
March 7, 2006



First Edition - March 2006

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 2006. 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® eServer™ xSeries® 460 configured as a client/server system. This report documents the full disclosure information required by the TPC Benchmark C Standard Specification, Revision 5.6, including the methodology used to achieve the reported results. All testing fully complied with this revision level.

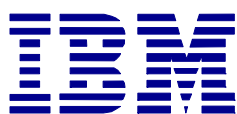
The software used on the xSeries 460 system includes Microsoft® Windows® Server 2003 Enterprise x64 Edition operating system and IBM DB2® UDB 8.2 Enterprise Edition database.

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 460	DB2 UDB 8.2 Microsoft Windows Server 2003 Enterprise x64 Edition	\$1,273,691 USD	273,520	\$4.66 USD	May 1, 2006

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™ Series® 460 c/s
and
DB2® UDB 8.2**

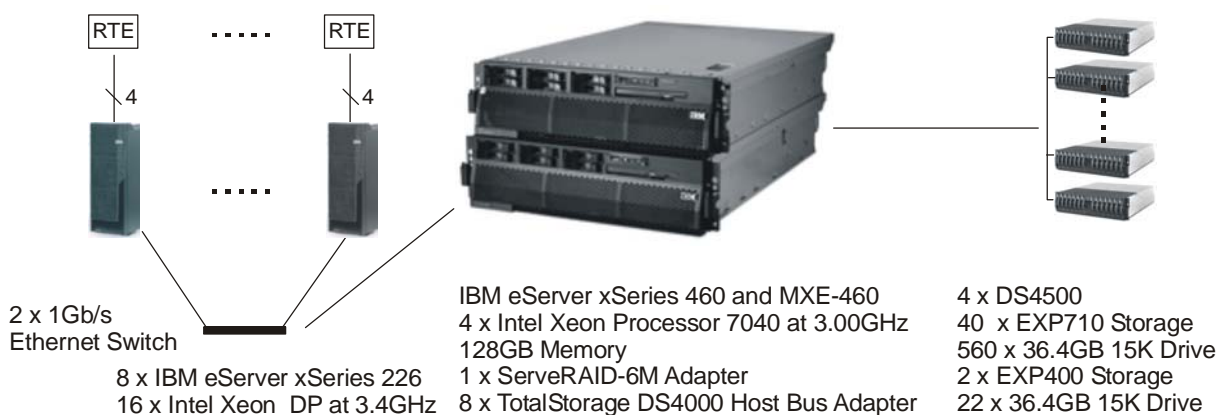
TPC-C Rev. 5.6

Report Date: Mar. 7, 2006

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
\$1,273,691 USD	273,520 tpmC	\$4.66 USD / tpmC	May 1, 2006

Database Server Processors/Cores/Threads	Database Manager	Operating System	Other Software	Number of Users
4/8/16 Intel® Xeon® Processor 7040 3.00GHz	DB2 UDB 8.2	Microsoft® Windows® Server 2003 Enterprise x64 Edition	Microsoft Visual C++ Standard Edition Microsoft COM+	217,600

8 RTEs emulating 217,600 users



System Component	Qty	Server:	Qty	Each of Eight Clients:
Processors/Cores/Threads	4/8/16	Intel Xeon Processor 7040 at 3.00GHz	2/2/4	Intel Xeon Processor at 3.4GHz
Cache		2x2MB L2 Cache		2MB L2 Cache
Memory	32	4GB ECC RDIMM	4	512MB
Disk Controllers	1	ServeRAID-6M Adapter	1	Ultra320 SCSI Interface
	8	TotalStorage DS4000 HBA	2	
	4	TotalStorage DS4500		
Disk Drives	560	36.4GB 2Gbps FC (15K)	1	36.4GB (15000 rpm)
	22	36.4GB Ultra320 (15K)		
Total Storage		19077GB		

IBM Corporation	IBM @server 460 c/s			TPC-C Revision 5.6			
	with DB2 UDB			Report Date: Mar. 7, 2006			
Description	Part Number	Third Party Brand	Pricing	Unit Price	Quantity	Extended Price	3-Yr. Maint. Price
Server Hardware							
xSeries 460 with 2 x Intel Xeon Processor 7040 3.00GHz/2x2MB L2	8872-6RU*	IBM	1a	25,999	1	25,999	
Intel Xeon Processor 7040 3.00GHz/2x2MB L2	25R8942	IBM	1	5,699	2	11,398	
MXE-460 (0 Processors, 0 Memory, 2 Memory Cards)	8874-1RU	IBM	1a	6,999	1	6,999	
Scalability Cable 2.3.M	13M7414	IBM	1	299	2	598	
8GB (2x4GB) PC2-3200 CL3 2RX4 ECC DDR2 SDRAM RDIMM	30R5145	IBM	1	5,299	16	84,784	
Active Memory 4-Slot Memory Expansion Card	13M7409	IBM	1	499	4	1,996	
ServeRAID-6M Ultra320 SCSI Adapter	32P0033	IBM	1	879	1	879	
E54 15" (13.8" Viewable) Color Monitor	633147N	IBM	1	139	1	139	
ThinkPlus Enhanced Performance USB Keyboard	73P2620	IBM	1	39	1	39	
ScrollPoint 800 DPI Optical Mouse - USB & PS/2	90P0742	IBM	1	25	1	25	
ServicePac for 3-Year 24x7x4 Support (x460 and MXE-460)	96P2688	IBM	1	3,390	2		6,780
ServicePac for 3-Year 24x7x4 Support (Monitor)	30L9183	IBM	1	90	1		90
					Subtotal	132,856	6,870
Server Storage							
IBM TotalStorage DS4000 Host Bus Adapter	24P0960	IBM	1	1,485	8	11,880	
IBM TotalStorage DS4500 Midrange Disk Subsystem	174290U	IBM	1a	49,900	4	199,600	
IBM DS4000 Mini Hub	19K1269	IBM	1	899	8	7,192	
IBM Short Wave SFP Module (4 Pack)	22R0483	IBM	1	550	42	23,100	
IBM 1m LC-LC Fibre Channel Cable	19K1247	IBM	1	79	72	5,688	
IBM 5m LC-LC Fibre Channel Cable	19K1248	IBM	1	129	16	2,064	
IBM TotalStorage DS4000 EXP710 Storage Exp. Unit	1740710	IBM	1a	6,000	40	240,000	
2Gbps FC 36.4GB 15K Hot-Swap HDD	06P5772	IBM	1	1,115	560	624,400	
IBM EXP400 Rack Storage Exp. Enclosure	17331RU	IBM	1a	3,099	2	6,198	
36.4GB 15K Ultra320 SCSI Hot-Swap HDD	90P1380	IBM	1	279	22	6,138	
2M SCSI cable	03K9310	IBM	1	75	2	150	
IBM UPS 750TLV	21301TX	IBM	1	299	1	299	
IBM S2 42U Standard Rack	930745X	IBM	1	1,489	5	7,445	
ServicePac for 3-Year 24x7x4 Support (EXP400)	41L2768	IBM	1	760	2		1,520
ServicePac for 3-Year 24x7x4 Support (EXP710)	41L2768	IBM	1	760	40		30,400
ServicePac for 3-Year 24x7x4 Support (DS4500)	96P2062	IBM	1	1,087	4		4,348
ServicePac for 3-Year 24x7x4 Support (Rack)	41L2760	IBM	1	300	5		1,500
					Subtotal	1,134,154	37,768
Server Software							
DB2 UDB ESE 8.2 for Windows Operating Systems on 64-Bit Extended Systems - SW License and Maintenance 12 Months		IBM	2	23,902	4	95,608	
SW Maintenance Renewal - 1 Year		IBM	2	1,138	8		9,104
Microsoft Windows Server 2003 Enterprise x64 Edition *	P72-00981	Microsoft	3	3,999	1	3,999	
Microsoft Problem Resolution Services		Microsoft	3	245	1		245
					Subtotal	99,607	9,349
Client Hardware							
x226 with 3.4GHz/2MB Xeon DP, 512MB (2x256MB) Memory	8648-6AU	IBM	1a	1,499	8	11,992	
3.4GHz/2MB Xeon DP Processor Upgrade	13N0676	IBM	1	879	8	7,032	
1GB (2x512MB) PC-3200 DDR2 ECC SDRAM RDIMM	73P3522	IBM	1	399	16	6,384	
36.4GB 15K Ultra320 SCSI Drive	90P1380	IBM	1	279	8	2,232	
PRO/1000 GT Dual-Port Server Adapter	73P5101	Intel	1	249	16	3,984	
E54 15" (13.8" Viewable) Color Monitor	633147N	IBM	1	139	8	1,112	
ServicePac for 3-Year 24x7x4 Support (x226)	96P2250	IBM	1	586	8		4,688
ServicePac for 3-Year 24x7x4 Support (Monitor)	30L9183	IBM	1	90	8		720
					Subtotal	32,736	5,408
Client Software							
Microsoft Windows 2000 Server with COM+*	C11-00821	Microsoft	3	738	8	5,904	
Microsoft Visual C++ Standard Edition	254-00170	Microsoft	3	109	1	109	
					Subtotal	6,013	
Network Components							
NetGear 8-Port Ethernet Switch (2 spares)	GS-108		4	76	4	304	
Ethernet Cable 8 Ft. (2 spares)	CC5E-B14B		4	4	12	48	
					Subtotal	352	
					Total	1,405,718	59,395
					Discount	185,417	6,005
Discounts amounts based on unit volume; prices vary if purchased separately (see Note 1).							
Pricing: 1- Compsat Technology; 2 - IBM; 3 - Microsoft; 4 - newegg.com				Three-Year Cost of Ownership USD:		\$1,273,691	
Note 1: All items with pricing 1 - 12% discount; all items with pricing 1a - 18% discount.				tpmC:		273,520	
Audited by Bradley J. Askins, InfoSizing, Inc.				\$ USD/tpmC:		\$4.66	
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: **273,520 tpmC**

Response Times (in seconds)	90th Percentile	Average	Maximum
New-Order	0.30	0.18	1.77
Payment	0.40	0.28	3.34
Delivery	0.30	0.30	1.50
Stock Level	0.42	0.33	2.48
Order Status	0.30	0.22	2.08
Delivery (Deferred)	0.28	0.19	2.05
Menu	0.29	0.20	1.56
Response Time Delay Added for Emulated Components: 0.1 Seconds			
Transaction Mix (in percent of total transactions)			Percent
New-Order			44.95
Payment			43.02
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 / 12.04	18.02 / 120.31
Payment	3.00 / 0.00	3.00 / 12.04	3.02 / 120.31
Delivery	2.00 / 0.00	2.00 / 5.04	2.02 / 50.31
Stock Level	2.00 / 0.00	2.00 / 5.04	2.02 / 50.31
Order Status	2.00 / 0.00	2.00 / 10.04	2.02 / 100.31
Test Duration			
Ramp-up time			25 minutes 51 seconds
Measurement interval			120 minutes
Number of checkpoints			NA
Checkpoint interval			NA
Number of transactions (all types) completed in measurement interval			73,016,459

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	35
Partitions/Replications Mapping	35
60-Day Space Requirement	35
Clause 5: Performance Metrics and Response Time Related Items	36
Measured tpmC	36
Response Times	36
Keying/Think Times	36
Response Time Frequency Distribution Curves	37
Performance Curve for Response Time vs. Throughput	39
New Order Think Time Distribution	40
Steady State Methodology	41
Work Performed during Steady State	41
Measurement Interval	42
Transaction Mix	42
Percentage of Total Mix	42
Number of Checkpoints	43
Clause 6: SUT, Driver and Communication Definition Related Items	44
Description of RTE	44
Emulated Components	44
Benchmarked and Targeted System Configuration Diagrams	44
Network Configuration	44
Network Bandwidth	44
Operator Intervention	44

Clause 7: Pricing Related Items	45
Hardware and Software Components	45
Availability Date	45
Measured tpmC	45
Country-Specific Pricing	45
Usage Pricing	45
System Pricing	46
Clause 9: Audit Related Items	47
Auditor	47
Availability of the Full Disclosure Report	47
<i>Attestation letter</i>	48
Appendix A: Client Server Code	50
A.1 Client/Terminal Handler Code	50
<i>makefile.config</i>	50
<i>include/tpccapp.h</i>	50
<i>include/tpccdbg.h</i>	51
<i>Src.Common/Makefile</i>	51
<i>Src.Common/tpccctx.sqc</i>	52
<i>Src.Common/tpccdbg.c</i>	53
<i>Src.Cli/Makefile</i>	58
<i>Src.Cli/tpcccli.sqc</i>	59
<i>nullDB/nullDB.cpp</i>	65
<i>nullDB/stdafx.h</i>	68
<i>nullDB/stdafx.cpp</i>	68
<i>tpccIsapi/htmlPhraser.h</i>	68
<i>tpccIsapi/resource.h</i>	69
<i>tpccIsapi/StdAfx.h</i>	69
<i>tpccIsapi/tpcc.h</i>	69
<i>tpccIsapi/tpccIsapi.def</i>	79
<i>tpccIsapi/tpccIsapi.hpp</i>	79
<i>tpccIsapi/htmlPhraser.cpp</i>	80
<i>tpccIsapi/StdAfx.cpp</i>	82
<i>tpccIsapi/tpccIsapi.cpp</i>	82
A.2 Client Transaction Code	107
<i>Makefile.config</i>	107
<i>tpccenv.bat</i>	108
<i>include/db2tpcc.h</i>	108
<i>include/lval.h</i>	110
<i>include/tpccapp.h</i>	110
<i>include/tpccdbg.h</i>	111
<i>Src.Common/Makefile</i>	112
<i>Src.Common/tpccctx.sqc</i>	113
<i>Src.Common/tpccdbg.c</i>	114
<i>Src.Common/tpccmisc.c</i>	119
<i>Src.Srv/Makefile</i>	119
<i>Src.Srv/cat-func.ddl</i>	121
<i>Src.Srv/cat-proc.ddl</i>	128
<i>Src.Srv/tpcc_all_sql.sqc</i>	128
<i>Src.Srv/rpctpcc.def</i>	158
<i>utils/EXPLAIN.ddl</i>	158
<i>utils/UNEXPLAIN.ddl</i>	162
<i>tpccCom/comreg.h</i>	162
<i>tpccCom/dlldatax.h</i>	163
<i>tpccCom/Resource.h</i>	163
<i>tpccCom/stdafx.h</i>	163

<i>tpccCom/tpccCom.h</i>	164
<i>tpccCom/tpcc_com.h</i>	165
<i>tpccCom/tpccCom.def</i>	166
<i>tpccCom/tpccCom.idl</i>	166
<i>tpccCom/tpcc_com.rgs</i>	167
<i>tpccCom/comreg.cpp</i>	167
<i>tpccCom/stdafx.cpp</i>	167
<i>tpccCom/tpccCom.cpp</i>	167
<i>tpccCom/tpcc_com.cpp</i>	168
<i>TpccCom/dlldata.c</i>	172
<i>tpccCom/dlldatax.c</i>	172
<i>tpccCom/tpccCom_i.c</i>	172
<i>tpccCom/tpccCom_p.c</i>	172
<i>TpccDB2Glue/stdafx.h</i>	183
<i>tpccDB2Glue/tpccDB2glue.h</i>	183
<i>tpccDB2Glue/stdafx.cpp</i>	184
<i>tpccDB2Glue/tpccDB2glue.cpp</i>	184
<i>NullDB.cpp</i>	188
<i>NullDB.h</i>	191
<i>Stdafx.cpp</i>	192
<i>Stdafx.h</i>	192
<i>Stdafx.cpp</i>	192
<i>StdAfx.h</i>	192
<i>TpccComClient.cpp</i>	192
<i>HtmlPhraser.cpp</i>	193
<i>HtmlPhraser.h</i>	196
<i>Resource.h</i>	197
<i>StdAfx.cpp</i>	197
<i>StdAfx.h</i>	197
<i>Tpcc.h</i>	197
<i>TpccIsapi.cpp</i>	208
<i>TpccIsapi.def</i>	236
<i>TpccIsapi.hpp</i>	236
<i>TpccIsapi.rc</i>	237
Appendix B: Database Design Scripts	239
<i>create_tablespace.ddl</i>	239
<i>alter_tablespace.ddl</i>	248
<i>alter_bufferpool.ddl</i>	250
<i>create_bufferpool.ddl</i>	251
<i>create_database.ddl</i>	252
<i>alttbsp_pf_0.ddl</i>	252
<i>alttbsp_pf_4096.ddl</i>	253
<i>crconst_customer_all.ddl</i>	255
<i>crconst_district_all.ddl</i>	257
<i>crconst_history_all.ddl</i>	257
<i>crconst_new_order_all.ddl</i>	258
<i>crconst_order_line_all.ddl</i>	259
<i>crconst_orders_all.ddl</i>	259
<i>crconst_stock_all.ddl</i>	260
<i>crconst_warehouse_all.ddl</i>	262
<i>cridx_cust_idxb_all.ddl</i>	262
<i>cridx_ordr_idxb_all.ddl</i>	264
<i>crtb_customer_all.ddl</i>	264
<i>crtb_district_all.ddl</i>	275
<i>crtb_history_all.ddl</i>	276

<i>crtb_item.ddl</i>	277
<i>crtb_new_order_all.ddl</i>	277
<i>crtb_orders_all.ddl</i>	279
<i>crtb_order_line_all.ddl</i>	281
<i>crtb_stock_all.ddl</i>	282
<i>crtb_warehouse_all.ddl</i>	291
<i>crvw_customer.ddl</i>	292
<i>crvw_district.ddl</i>	293
<i>crvw_history.ddl</i>	293
<i>crvw_new_order.ddl</i>	293
<i>crvw_order_line.ddl</i>	293
<i>crvw_orders.ddl</i>	294
<i>crvw_stock.ddl</i>	294
<i>crvw_warehouse.ddl</i>	294
<i>gen_customer_all.bat</i>	294
<i>gen_district_all.bat</i>	295
<i>gen_history_all.bat</i>	295
<i>gen_item_1.bat</i>	295
<i>gen_new_order_all.bat</i>	295
<i>gen_orders_all.bat</i>	296
<i>gen_stock_all.bat</i>	296
<i>gen_warehouse_all.bat</i>	296
<i>load_customer_all.ddl</i>	296
<i>load_district_all.ddl</i>	299
<i>load_history_all.ddl</i>	299
<i>load_order_line_all.ddl</i>	300
<i>load_new_order_all.ddl</i>	300
<i>load_item_1.ddl</i>	300
<i>load_orders_all.ddl</i>	301
<i>load_stock_all.ddl</i>	301
<i>load_warehouse_all.ddl</i>	303
<i>rnst_customer_all.ddl</i>	304
<i>rnst_district_all.ddl</i>	305
<i>rnst_history_all.ddl</i>	305
<i>rnst_item.ddl</i>	305
<i>rnst_new_order_all.ddl</i>	305
<i>rnst_order_line_all.ddl</i>	306
<i>rnst_orders_all.ddl</i>	306
<i>rnst_stock_all.ddl</i>	306
<i>rnst_warehouse_all.ddl</i>	308
<i>dbgen\gendata.c</i>	308
<i>dbgen\makefile</i>	316
<i>dbgen\tpccrnd.c</i>	316
<i>dbgen\include\db2tpcc.h</i>	319
<i>dbgen\include\lval.h</i>	321
<i>dbgen\include\platform.h</i>	321
<i>dbgen\include\tpccrnd.h</i>	323
<i>dbgen\makefile.config</i>	323
<i>dbgen\Src.Common\makefile</i>	324
<i>dbgen\Src.Common\tpccmisc.c</i>	325
<i>dbgen\tpccenv.bat</i>	325
Appendix C: Tunable Parameters	327
IBM DB2 UDB	327
<i>Database Manager Configuration</i>	327
<i>Database Configuration</i>	328

<i>DB2set Parameters</i>	330
<i>Aff8.cfg</i>	330
Microsoft Windows Server 2003 Enterprise x64 Edition	330
<i>Server Configuration Parameters</i>	330
<i>System Information Report</i>	330
ServeRAID-6M Disk Controller Configuration Parameters	379
DS4500 Disk Subsystem Configuration	385
<i>Rack 1</i>	385
Client Configuration	413
<i>Microsoft Windows 2000 Client System Information Report</i>	413
<i>Client Configuration Parameters</i>	437
<i>Client Registry Parameters</i>	437
RTE Parameters	439
Appendix D: 60-Day Space	441
Appendix E: Third-Party Quotations	442

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 5.6.

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

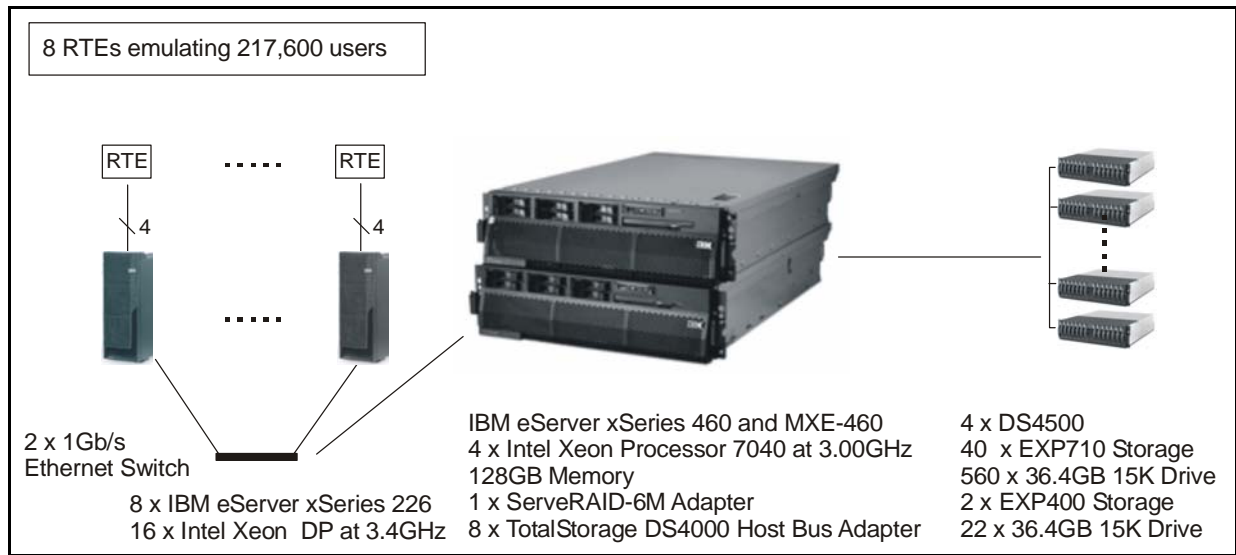
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

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 5.6, 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 eight IBM eServer xSeries 226 systems 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 is the DBMS executing on the server

Measured and Priced Configurations



The measured and priced configurations were different in that the measured configuration used four x226 clients each with two 3.2GHz/2MB processors and four x226 with two 3.4GHz/2MB processors. For the priced configuration, eight x226 clients with 3.4GHz processors were priced.

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 Table 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.

All tables except the Item table were horizontally partitioned. The tables for Stock and Customer were partitioned into 40 tables of 544 warehouses each while other tables were partitioned into 8 tables of 2,720 warehouses each. For each partitioned table, a view was created over all table partitions to provide full transparency of data manipulation.

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 460 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	85.00%
Remote warehouse payment transactions	15.00%
Non-Primary Key Access	
Payment transactions using C_LAST	60.01%
Order-Status transactions using C_LAST	60.00%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.95%
Payment	43.02%
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 batch 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 batch 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 DS4500 disk subsystem contains two RAID controller blades, which provide RAID functionality to the attached disks. Each RAID controller blade contains a Read/Write cache. Write caching was enabled. The attached disks contained a portion of each of the tables in the tpcc database. During steady state one RAID controller blade was deinserted causing DB2 to report errors accessing that device.

The following steps were successfully performed to pass the Durability test of failure of a DS4500 controller with database tables:

1. The contents of the database were backed up to several database dump devices during the initial database load.
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 68,000 users submitting transactions.
4. A DS4500 controller blade containing a portion of each of the tables in the tpcc database was pulled out causing DB2 to report errors accessing that device.
5. The run was aborted and DB2 was stopped.
6. The failed DS4500 controller blade was reinserted and was recovered.
7. The database was recovered and restored from the backup dump devices. Afterwards, the transaction log was rolled forward 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 217,600 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-1E 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 21,760 warehouses and the audited run used 21,760 warehouses. No warehouses were inactive.

Table 4-1. Initial Cardinality of Tables

Table Name	Rows
Warehouse	21,760
District	217,600
Item	100,000
Stock	2,176,000,000
Customer	652,800,000
New-Order	195,840,000
Orders	652,800,000
Order-Line	6,528,244,745
History	652,800,000
Inactive Warehouses	0

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-1E disk array with 20 physical disks of 36.4GB each attached via an internal ServeRAID-6M adapter.

For the database tables, there is a total of 560 disk drives. Each physical disk has a capacity of 36.4GB. A total of 8 Fibre Channel Host Bus Adapters connect these 560 disks.

Each disk volume, as seen by the Windows operating system, is configured as a RAID-0 disk array with 14 physical disks. The total number of disk volumes is 40. Each disk volume is further partitioned into 12 partitions. Each partition corresponds to a DB2 container.

For the tables Stock and Customer, each partition within a partitioned table is made of one DB2 container; thus, there are 40 table partitions with a total of 40 DB2 containers so that the corresponding view spans all 8 adapters connected to the 560 disks.

The ITEM table, which is not partitioned, is made up of 40 DB2 containers and also spans all 8 adapters.

For all other tables, each partition within a partitioned table is made of 5 DB2 containers; thus, there are 8 table partitions with a total of 40 DB2 containers so that the corresponding view spans all 8 adapters connected to the 560 disks.

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

Table 4-2. Data Distribution for the Benchmarked Configuration

Disk #	Drives	Partition	Size	Use
0	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\001 C:\Containers\CST\001 C:\Containers\CSTI\001 C:\Containers\NEWA\001 C:\Containers\ORDI\001 C:\Containers\ORD\001 C:\Containers\ITM\001 C:\Containers\WAR\001 C:\Containers\DIS\001 C:\Containers\OLN\001 C:\Containers\HST\001 C:\Containers\NEWB\001 C:\backup\bk_001	467.61GB	Database Files
1	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\002 C:\Containers\CST\002 C:\Containers\CSTI\002 C:\Containers\NEWA\002 C:\Containers\ORDI\002 C:\Containers\ORD\002 C:\Containers\ITM\002 C:\Containers\WAR\002 C:\Containers\DIS\002 C:\Containers\OLN\002 C:\Containers\HST\002 C:\Containers\NEWB\002 C:\backup\bk_002	467.61GB	Database Files
2	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\003 C:\Containers\CST\003 C:\Containers\CSTI\003 C:\Containers\NEWA\003 C:\Containers\ORDI\003 C:\Containers\ORD\003 C:\Containers\ITM\003 C:\Containers\WAR\003 C:\Containers\DIS\003 C:\Containers\OLN\003 C:\Containers\HST\003 C:\Containers\NEWB\003 C:\backup\bk_003	467.61GB	Database Files
3	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\004 C:\Containers\CST\004 C:\Containers\CSTI\004 C:\Containers\NEWA\004 C:\Containers\ORDI\004 C:\Containers\ORD\004 C:\Containers\ITM\004 C:\Containers\WAR\004 C:\Containers\DIS\004 C:\Containers\OLN\004 C:\Containers\HST\004 C:\Containers\NEWB\004 C:\backup\bk_004	467.61GB	Database Files

4	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\005 C:\Containers\CST\005 C:\Containers\CSTI\005 C:\Containers\NEWA\005 C:\Containers\ORDI\005 C:\Containers\ORD\005 C:\Containers\ITM\005 C:\Containers\WAR\005 C:\Containers\DIS\005 C:\Containers\OLN\005 C:\Containers\HST\005 C:\Containers\NEWB\005 C:\backup\bk_005	467.61GB	Database Files
5	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\006 C:\Containers\CST\006 C:\Containers\CSTI\006 C:\Containers\NEWA\006 C:\Containers\ORDI\006 C:\Containers\ORD\006 C:\Containers\ITM\006 C:\Containers\WAR\006 C:\Containers\DIS\006 C:\Containers\OLN\006 C:\Containers\HST\006 C:\Containers\NEWB\006 C:\backup\bk_006	467.61GB	Database Files

Disk #	Drives	Partition	Size	Use
6	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\007 C:\Containers\CST\007 C:\Containers\CSTI\007 C:\Containers\NEWA\007 C:\Containers\ORD\007 C:\Containers\ORD\007 C:\Containers\ITM\007 C:\Containers\WAR\007 C:\Containers\DIS\007 C:\Containers\OLN\007 C:\Containers\HST\007 C:\Containers\NEWB\007 C:\backup\bk_007	467.61GB	Database Files
7	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\008 C:\Containers\CST\008 C:\Containers\CSTI\008 C:\Containers\NEWA\008 C:\Containers\ORD\008 C:\Containers\ORD\008 C:\Containers\ITM\008 C:\Containers\WAR\008 C:\Containers\DIS\008 C:\Containers\OLN\008 C:\Containers\HST\008 C:\Containers\NEWB\008 C:\backup\bk_008	467.61GB	Database Files
8	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\009 C:\Containers\CST\009 C:\Containers\CSTI\009 C:\Containers\NEWA\009 C:\Containers\ORD\009 C:\Containers\ORD\009 C:\Containers\ITM\009 C:\Containers\WAR\009 C:\Containers\DIS\009 C:\Containers\OLN\009 C:\Containers\HST\009 C:\Containers\NEWB\009 C:\backup\bk_009	467.61GB	Database Files
9	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\010 C:\Containers\CST\010 C:\Containers\CSTK\010 C:\Containers\NEWA\010 C:\Containers\ORD\010 C:\Containers\ORD\010 C:\Containers\ITM\010 C:\Containers\WAR\010 C:\Containers\DIS\010 C:\Containers\OLN\010 C:\Containers\HST\010 C:\Containers\NEWB\010 C:\backup\bk_010	467.61GB	Database Files

10	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\011 C:\Containers\CST\011 C:\Containers\CSTI\011 C:\Containers\NEWA\011 C:\Containers\ORDI\011 C:\Containers\ORD\011 C:\Containers\ITM\011 C:\Containers\WAR\011 C:\Containers\DIS\011 C:\Containers\OLN\011 C:\Containers\HST\011 C:\Containers\NEWB\011 C:\backup\bk_011	467.61GB	Database Files
11	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\012 C:\Containers\CSTK\012 C:\Containers\CSTI\012 C:\Containers\NEWA\012 C:\Containers\ORDI\012 C:\Containers\ORD\012 C:\Containers\ITM\012 C:\Containers\WAR\012 C:\Containers\DIS\012 C:\Containers\OLN\012 C:\Containers\HST\012 C:\Containers\NEWB\012 C:\backup\bk_012	467.61GB	Database Files

Disk #	Drives	Partition	Size	Use
12	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\013 C:\Containers\CST\013 C:\Containers\CSTI\013 C:\Containers\NEWA\013 C:\Containers\ORDI\013 C:\Containers\ORD\013 C:\Containers\ITM\013 C:\Containers\WAR\013 C:\Containers\DIS\013 C:\Containers\OLN\013 C:\Containers\HST\013 C:\Containers\NEWB\013 C:\backup\bk_013	467.61GB	Database Files
13	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\014 C:\Containers\CST\014 C:\Containers\CSTI\014 C:\Containers\NEWA\014 C:\Containers\ORDI\014 C:\Containers\ORD\014 C:\Containers\ITM\014 C:\Containers\WAR\014 C:\Containers\DIS\014 C:\Containers\OLN\014 C:\Containers\HST\014 C:\Containers\NEWB\014 C:\backup\bk_014	467.61GB	Database Files
14	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\015 C:\Containers\CST\015 C:\Containers\CSTI\015 C:\Containers\NEWA\015 C:\Containers\ORDI\015 C:\Containers\ORD\015 C:\Containers\ITM\015 C:\Containers\WAR\015 C:\Containers\DIS\015 C:\Containers\OLN\015 C:\Containers\HST\015 C:\Containers\NEWB\015 C:\backup\bk_015	467.61GB	Database Files
15	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\016 C:\Containers\CST\016 C:\Containers\CSTI\016 C:\Containers\NEWA\016 C:\Containers\ORDI\016 C:\Containers\ORD\016 C:\Containers\ITM\016 C:\Containers\WAR\016 C:\Containers\DIS\016 C:\Containers\OLN\016 C:\Containers\HST\016 C:\Containers\NEWB\016 C:\backup\bk_016	467.61GB	Database Files

16	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\017 C:\Containers\CST\017 C:\Containers\CSTI\017 C:\Containers\NEWA\017 C:\Containers\ORDI\017 C:\Containers\ORD\017 C:\Containers\ITM\017 C:\Containers\WAR\017 C:\Containers\DIS\017 C:\Containers\OLN\017 C:\Containers\HST\017 C:\Containers\NEWB\017 C:\backup\bk_017	467.61GB	Database Files
17	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\018 C:\Containers\CST\018 C:\Containers\CSTI\018 C:\Containers\NEWA\018 C:\Containers\ORDI\018 C:\Containers\ORD\018 C:\Containers\ITM\018 C:\Containers\WAR\018 C:\Containers\DIS\018 C:\Containers\OLN\018 C:\Containers\HST\018 C:\Containers\NEWB\018 C:\backup\bk_018	467.61GB	Database Files

Disk #	Drives	Partition	Size	Use
18	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\019 C:\Containers\CST\019 C:\Containers\CSTI\019 C:\Containers\NEWA\019 C:\Containers\ORDI\019 C:\Containers\ORD\019 C:\Containers\ITM\019 C:\Containers\WAR\019 C:\Containers\DIS\019 C:\Containers\OLN\019 C:\Containers\HST\019 C:\Containers\NEWB\019 C:\backup\bk_019	467.61GB	Database Files
19	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\020 C:\Containers\CST\020 C:\Containers\CSTI\020 C:\Containers\NEWA\020 C:\Containers\ORDI\020 C:\Containers\ORD\020 C:\Containers\ITM\020 C:\Containers\WAR\020 C:\Containers\DIS\020 C:\Containers\OLN\020 C:\Containers\HST\020 C:\Containers\NEWB\020 C:\backup\bk_020	467.61GB	Database Files
20	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\021 C:\Containers\CST\021 C:\Containers\CSTI\021 C:\Containers\NEWA\021 C:\Containers\ORDI\021 C:\Containers\ORD\021 C:\Containers\ITM\021 C:\Containers\WAR\021 C:\Containers\DIS\021 C:\Containers\OLN\021 C:\Containers\HST\021 C:\Containers\NEWB\021 C:\backup\bk_021	467.61GB	Database Files
21	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\022 C:\Containers\CST\022 C:\Containers\CSTI\022 C:\Containers\NEWA\022 C:\Containers\ORDI\022 C:\Containers\ORD\022 C:\Containers\ITM\022 C:\Containers\WAR\022 C:\Containers\DIS\022 C:\Containers\OLN\022 C:\Containers\HST\022 C:\Containers\NEWB\022 C:\backup\bk_022	467.61GB	Database Files

22	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\023 C:\Containers\CST\023 C:\Containers\CSTI\023 C:\Containers\NEWA\023 C:\Containers\ORDI\023 C:\Containers\ORD\023 C:\Containers\ITM\023 C:\Containers\WAR\023 C:\Containers\DIS\023 C:\Containers\OLN\023 C:\Containers\HST\023 C:\Containers\NEWB\023 C:\backup\bk_023	467.61GB	Database Files
23	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\024 C:\Containers\CST\024 C:\Containers\CSTI\024 C:\Containers\NEWA\024 C:\Containers\ORDI\024 C:\Containers\ORD\024 C:\Containers\ITM\024 C:\Containers\WAR\024 C:\Containers\DIS\024 C:\Containers\OLN\024 C:\Containers\HST\024 C:\Containers\NEWB\024 C:\backup\bk_024	467.61GB	Database Files
24	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\025 C:\Containers\CST\025 C:\Containers\CSTI\025 C:\Containers\NEWA\025 C:\Containers\ORDI\025 C:\Containers\ORD\025 C:\Containers\ITM\025 C:\Containers\WAR\025 C:\Containers\DIS\025 C:\Containers\OLN\025 C:\Containers\HST\025 C:\Containers\NEWB\025 C:\backup\bk_025	467.61GB	Database Files
25	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\026 C:\Containers\CST\026 C:\Containers\CSTI\026 C:\Containers\NEWA\026 C:\Containers\ORDI\026 C:\Containers\ORD\026 C:\Containers\ITM\026 C:\Containers\WAR\026 C:\Containers\DIS\026 C:\Containers\OLN\026 C:\Containers\HST\026 C:\Containers\NEWB\026 C:\backup\bk_026	467.61GB	Database Files

26	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\027 C:\Containers\CST\027 C:\Containers\CSTI\027 C:\Containers\NEWA\027 C:\Containers\ORDI\027 C:\Containers\ORD\027 C:\Containers\ITM\027 C:\Containers\WAR\027 C:\Containers\DIS\027 C:\Containers\OLN\027 C:\Containers\HST\027 C:\Containers\NEWB\027 C:\backup\bk_027	467.61GB	Database Files
27	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\028 C:\Containers\CST\028 C:\Containers\CSTI\028 C:\Containers\NEWA\028 C:\Containers\ORDI\028 C:\Containers\ORD\028 C:\Containers\ITM\028 C:\Containers\WAR\028 C:\Containers\DIS\028 C:\Containers\OLN\028 C:\Containers\HST\028 C:\Containers\NEWB\028 C:\backup\bk_028	467.61GB	Database Files

Disk #	Drives	Partition	Size	Use
28	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\029 C:\Containers\CST\029 C:\Containers\CSTI\029 C:\Containers\NEWA\029 C:\Containers\ORDI\029 C:\Containers\ORD\029 C:\Containers\ITM\029 C:\Containers\WAR\029 C:\Containers\DIS\029 C:\Containers\OLN\029 C:\Containers\HST\029 C:\Containers\NEWB\029 C:\backup\bk_029	467.61GB	Database Files
29	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\030 C:\Containers\CST\030 C:\Containers\CSTI\030 C:\Containers\NEWA\030 C:\Containers\ORDI\030 C:\Containers\ORD\030 C:\Containers\ITM\030 C:\Containers\WAR\030 C:\Containers\DIS\030 C:\Containers\OLN\030 C:\Containers\HST\030 C:\Containers\NEWB\030 C:\backup\bk_030	467.61GB	Database Files
30	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\031 C:\Containers\CST\031 C:\Containers\CSTI\031 C:\Containers\NEWA\031 C:\Containers\ORDI\031 C:\Containers\ORD\031 C:\Containers\ITM\031 C:\Containers\WAR\031 C:\Containers\DIS\031 C:\Containers\OLN\031 C:\Containers\HST\031 C:\Containers\NEWB\031 C:\backup\bk_031	467.61GB	Database Files
31	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\032 C:\Containers\CST\032 C:\Containers\CSTI\032 C:\Containers\NEWA\032 C:\Containers\ORDI\032 C:\Containers\ORD\032 C:\Containers\ITM\032 C:\Containers\WAR\032 C:\Containers\DIS\032 C:\Containers\OLN\032 C:\Containers\HST\032 C:\Containers\NEWB\032 C:\backup\bk_032	467.61GB	Database Files

32	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\033 C:\Containers\CST\033 C:\Containers\CSTI\033 C:\Containers\NEWA\033 C:\Containers\ORDI\033 C:\Containers\ORD\033 C:\Containers\ITM\033 C:\Containers\WAR\033 C:\Containers\DIS\033 C:\Containers\OLN\033 C:\Containers\HST\033 C:\Containers\NEWB\033 C:\backup\bk_033	467.61GB	Database Files
33	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\034 C:\Containers\CST\034 C:\Containers\CSTI\034 C:\Containers\NEWA\034 C:\Containers\ORDI\034 C:\Containers\ORD\034 C:\Containers\ITM\034 C:\Containers\WAR\034 C:\Containers\DIS\034 C:\Containers\OLN\034 C:\Containers\HST\034 C:\Containers\NEWB\034 C:\backup\bk_034	467.61GB	Database Files

Disk #	Drives	Partition	Size	Use
34	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\035 C:\Containers\CST\035 C:\Containers\CSTI\035 C:\Containers\NEWA\035 C:\Containers\ORDI\035 C:\Containers\ORD\035 C:\Containers\ITM\035 C:\Containers\WAR\035 C:\Containers\DIS\035 C:\Containers\OLN\035 C:\Containers\HST\035 C:\Containers\NEWB\035 C:\backup\bk_035	467.61GB	Database Files
35	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\036 C:\Containers\CST\036 C:\Containers\CSTI\036 C:\Containers\NEWA\036 C:\Containers\ORDI\036 C:\Containers\ORD\036 C:\Containers\ITM\036 C:\Containers\WAR\036 C:\Containers\DIS\036 C:\Containers\OLN\036 C:\Containers\HST\036 C:\Containers\NEWB\036 C:\backup\bk_036	467.61GB	Database Files
36	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\037 C:\Containers\CST\037 C:\Containers\CSTI\037 C:\Containers\NEWA\037 C:\Containers\ORDI\037 C:\Containers\ORD\037 C:\Containers\ITM\037 C:\Containers\WAR\037 C:\Containers\DIS\037 C:\Containers\OLN\037 C:\Containers\HST\037 C:\Containers\NEWB\037 C:\backup\bk_037	467.61GB	Database Files
37	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\038 C:\Containers\CST\038 C:\Containers\CSTI\038 C:\Containers\NEWA\038 C:\Containers\ORDI\038 C:\Containers\ORD\038 C:\Containers\ITM\038 C:\Containers\WAR\038 C:\Containers\DIS\038 C:\Containers\OLN\038 C:\Containers\HST\038 C:\Containers\NEWB\038 C:\backup\bk_038	467.61GB	Database Files

38	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\039 C:\Containers\CST\039 C:\Containers\CSTI\039 C:\Containers\NEWA\039 C:\Containers\ORDI\039 C:\Containers\ORD\039 C:\Containers\ITM\039 C:\Containers\WAR\039 C:\Containers\DIS\039 C:\Containers\OLN\039 C:\Containers\HST\039 C:\Containers\NEWB\039 C:\backup\bk_039	467.61GB	Database Files
39	14 - 36.4GB EXP710 Enclosure	C:\Containers\STK\040 C:\Containers\CST\040 C:\Containers\CSTI\040 C:\Containers\NEWA\040 C:\Containers\ORDI\040 C:\Containers\ORD\040 C:\Containers\ITM\040 C:\Containers\WAR\040 C:\Containers\DIS\040 C:\Containers\OLN\040 C:\Containers\HST\040 C:\Containers\NEWB\040 C:\backup\bk_040	467.61GB	Database Files
40	2 - 36.4GB EXP400 Enclosure	C:	33.9GB	OS Drive
41	20 - 36.4GB EXP400 Enclosure	L:	339.0GB	Database Log Drive

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.

Except the Item table, all other 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: 273,520 tpmC

Price per tpmC: \$4.66 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	0.18	1.77	0.30
Payment	0.28	3.34	0.40
Delivery	0.30	1.50	0.30
Stock Level	0.33	2.48	0.42
Order Status	0.22	2.08	0.30
Delivery (Deferred)	0.19	2.05	0.28
Menu	0.20	1.56	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 / 12.04	18.00 / 0.00	18.02 / 120.31
Payment	3.00 / 12.04	3.00 / 0.00	3.02 / 120.31
Delivery	2.00 / 5.04	2.00 / 0.00	2.02 / 50.31
Stock Level	2.00 / 5.04	2.00 / 0.00	2.02 / 50.31
Order Status	2.00 / 10.04	2.00 / 0.00	2.02 / 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

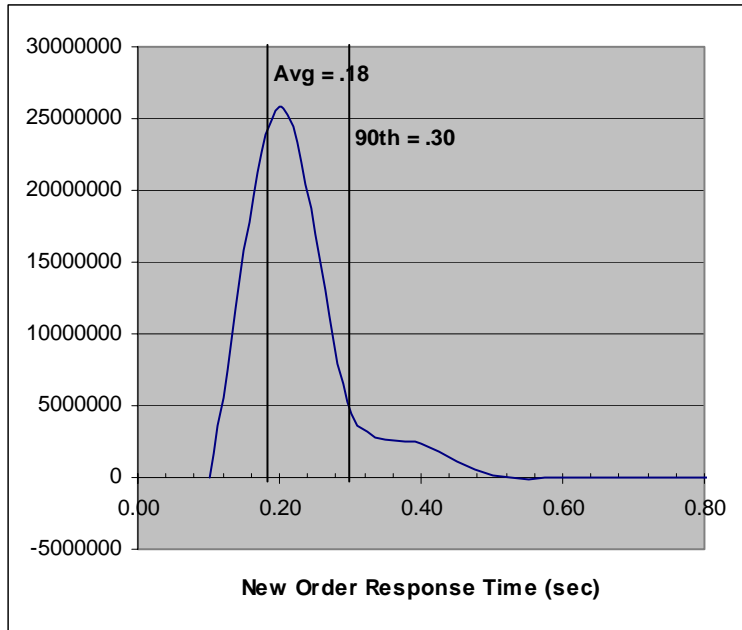


Figure 5-2. Payment Transaction - Response Time Frequency Distribution

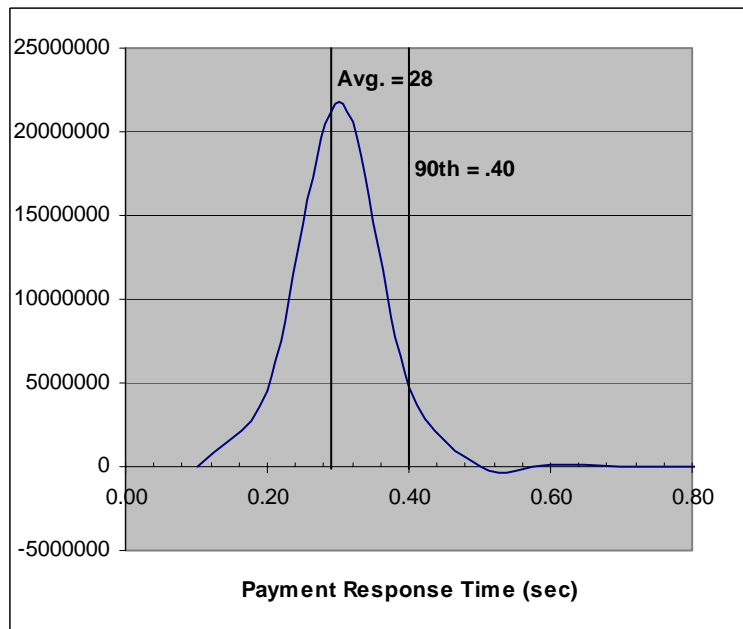


Figure 5-3. Order-Status Transaction - Response Time Frequency Distribution

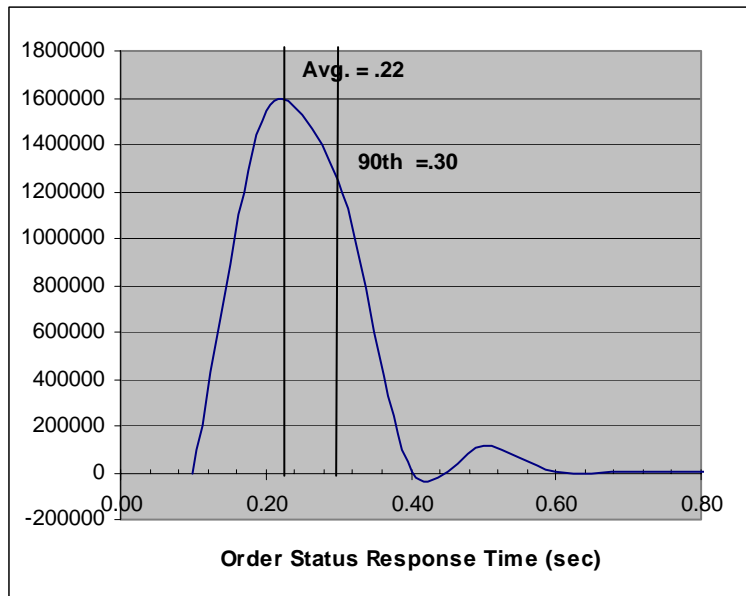


Figure 5-4. Delivery Transaction - Response Time Frequency Distribution

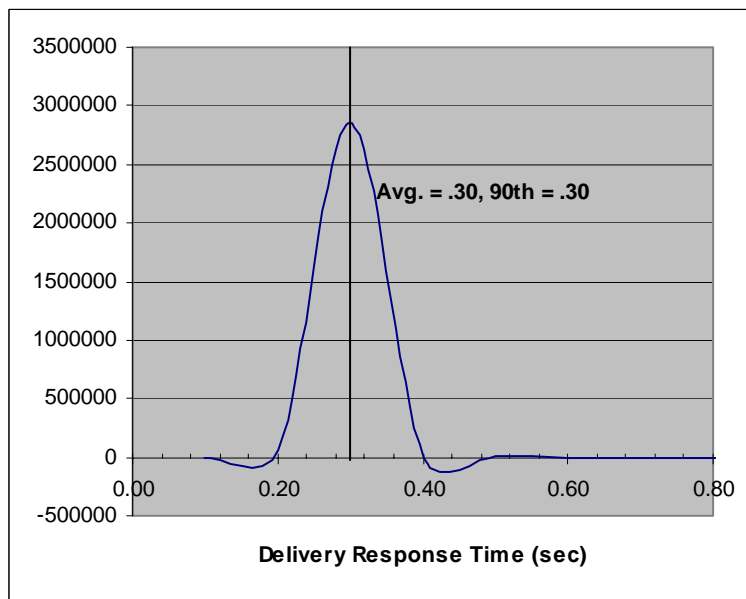
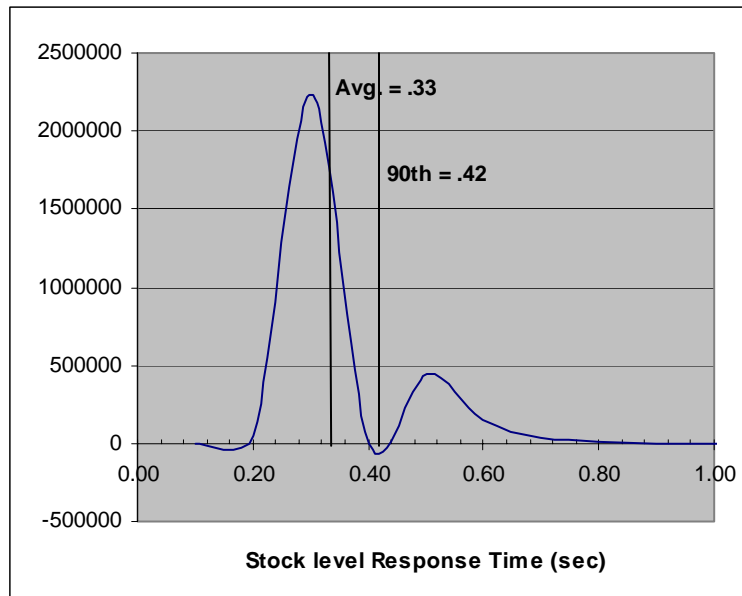


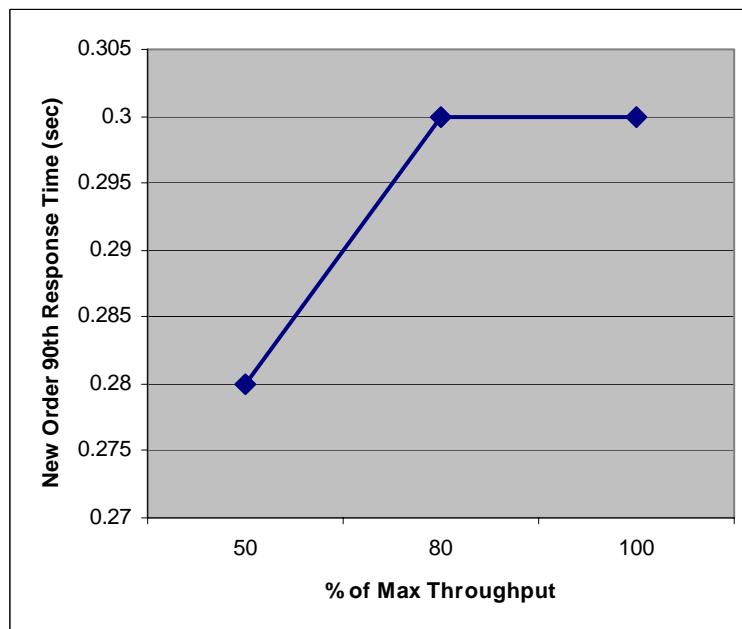
Figure 5-5. Stock-Level Transaction - Response Time Frequency Distribution



Performance Curve for Response Time vs. Throughput

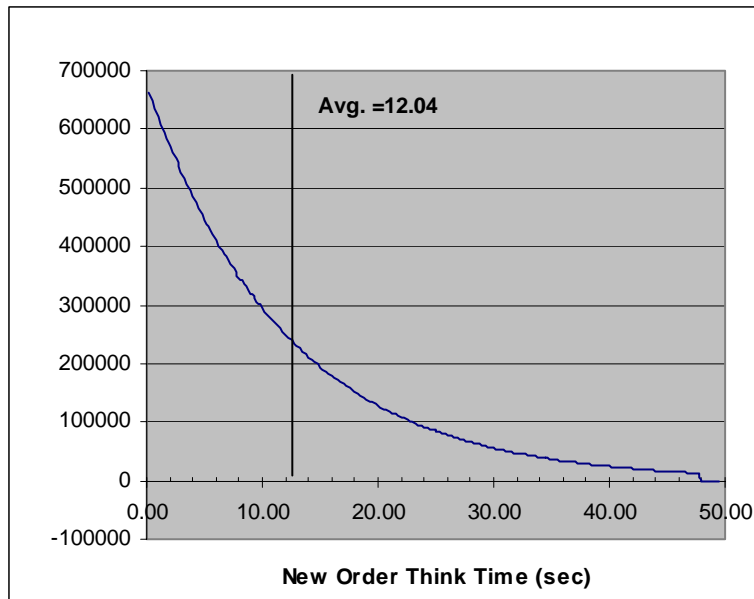
The performance curve for response time vs. throughput (see Clause 5.6.2) must be reported for the New-Order transaction.

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



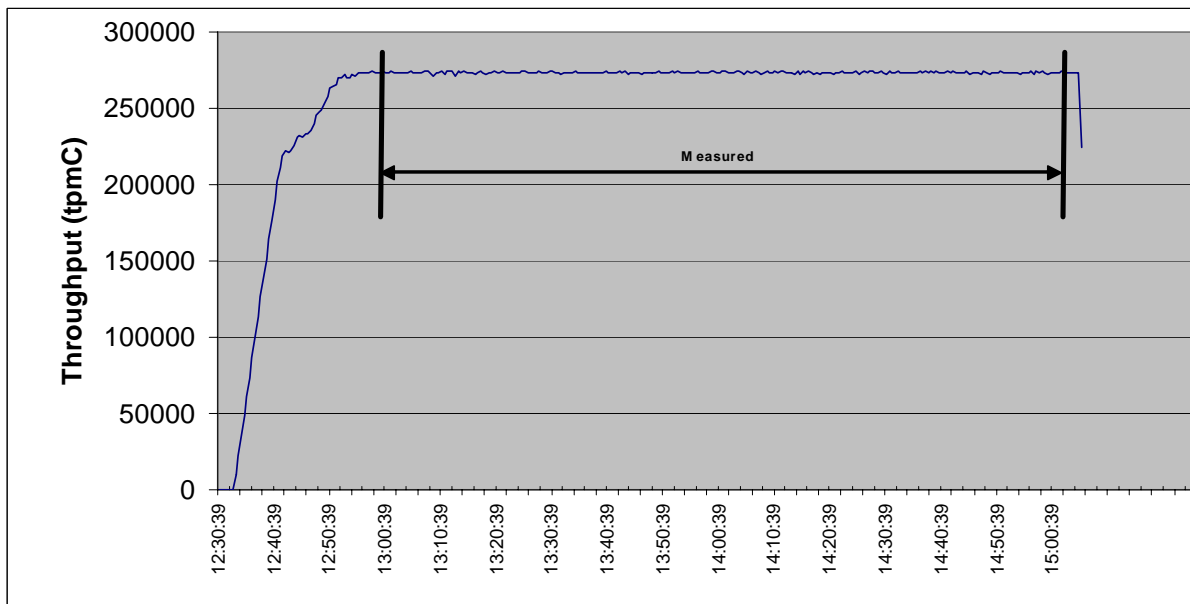
New Order Think Time Distribution

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 C. 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	85.00%
Remote warehouse payment transactions	15.00%
Non-Primary Key Access	
Payment transactions using C_LAST	60.01%
Order-Status transactions using C_LAST	60.00%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.95%
Payment	43.02%
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 page 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 available May 1, 2006.

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: 273,520 tpmC
- Price per tpmC: \$4.66 USD per tpmC
- Three-year cost of ownership: \$1,273,691 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
- 8 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: Celia Schreiber
 Manager, xSeries Performance
 IBM Systems and Technology
 Group
 3039 Cornwallis Road
 Research Triangle Park, NC 27709

February 24, 2006

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

Platform: **IBM @server xSeries 460 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 460				
4 x Xeon 7040 (3.0 GHz)	128 GB Main (2x2 MB L2 Cache)	582 x 36.4 GB	0.30 Seconds	273,520
Clients: Eight (8) IBM @server xSeries 226 (Specification for each)				
2 x Xeon DP (3.4 GHz)	2.5 GB Main (2 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:

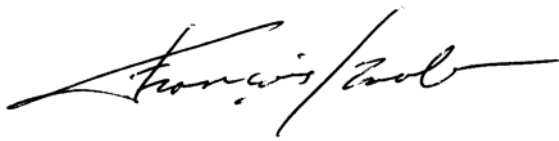
- The database records were the proper size
- The database was properly scaled and populated

- 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,



François Raab, President



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=$(LDFLAGS_SHLIB) /DEF:rptcpc.def
LDFLAGS_LIB=/LIBPATH:$(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=&
```

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
 */

#ifndef __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))
#endif

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

#ifndef __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 - 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) $(INCLUDES) \
-D$(DB2VERSION) \
-D$(TPCC_SPTYPE)

UTIL_OBJ =     tpcmisc$(OBJEXT) tpcdbg$(OBJEXT)
UTIL_OBJ_DB2 = tpcctx$(OBJEXT)

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

all:          dbgen connect $(UTIL_OBJ_DB2) disconnect

dbgen:       $(UTIL_OBJ)

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

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

connect:
- db2 connect to $(TPCC_DBNAME)

disconnect:
- db2 connect reset
- db2 terminate

rebind:      connect
            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

Src.Common/tpcctx.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 - 2005
** 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 <string.h>
#include <sqlutil.h>
#include "db2tpcc.h"
#include "tpcdbg.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; }
}

```

```

/* Copy 9 characters - 8 for dbname, 1 for NULL */
strncpy(dbname,in_dbname,9);
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;

sqlSetTypeCtx(SQL_CTX_MULTI_MANUAL);
sqlBeginCtx(&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;

sqlAttachToCtx(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;

sqlDetachFromCtx(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; }

sqlEndCtx(&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 - 2005
** 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 ORDDSTAT_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);

if ((err_fp = fopen(err_fn, "a+")) == NULL)
{
    return;
}

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 = %ld (%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;

```



```

{
    return;
}

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

fprintf(debug_fp,"n=====
=====\\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\\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,"tdeadlocks = %d (%X)\n",
ordstat_ptr->deadlocks, ordstat_ptr->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,"\\t\\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,"\\t\\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,"\\t\\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,"\\t\\ts_OL_AMOUNT[%d] = %d\n",
j, ordstat_ptr->item[j].s_OL_AMOUNT);
    fprintf(debug_fp,"\\t\\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\\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=====
=====\\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\\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 (%IX)\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)
OBSJ =          $(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT) \
                $(TPCC_ROOT)/Src.Common/tpccdbg$(OBJEXT) \
                $(TPCC_ROOT)/Src.Common/tpccctx$(OBJEXT) \
                tpcccli$(OBJEXT)
LIBS =          tpcccli$(LIBEXT)
#
#####
#####
# User Targets
#
#####
#####
all:            connect $(OBSJ) plan $(LIBS) disconnect
                $(AR) $(ARFLAGS) $(ARFLAGS_OUT)tpcccli$(LIBEXT)
$(OBSJ) $(ARFLAGS_LIB)
                @echo "-----"
                @echo "Please copy lval.h, db2tpcc.h, and tpcccli$(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) tpcccli.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 TPCCCLI -g -# 0 -o TPCCCLI.exfmt.plan
                - db2expln -d $(TPCC_DBNAME) -c $(TPCC_SCHEMA) -p
TPCCCLI -s 0 -g -o TPCCCLI.expln.plan
rebind:        connect
                db2 bind tpcccli.bnd $(BND_OPTS) QUERYOPT 7
```

```
#
#####
# Build Rules
#
#####
.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc
tpcccli.c:
                @echo "Prepping $*.sqc"
                -db2 prep $*.sqc $(PRP_OPTS) ISOLATION RR
                @echo "Binding $*.bnd"
                db2 bind $*.bnd $(BND_OPTS) QUERYOPT 7
#
#####
#####
# Dependencies
#
#####
#####
# Client Library:
tpcccli$(LIBEXT): $(OBSJ)
# Source
tpcc_all_sql$(OBJEXT):                tpcc_all_sql.c
# Headers
tpcc_all_sql.c:                        $(TPCC_ROOT)/include/db2tpcc.h
$(TPCC_ROOT)/include/lval.h
Src.Cli/tpcccli.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 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****
*****/
/*
 * tpcccli.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"
// -----
// 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 ;
```

```

// Primary comparison key: I_ID
// Secondary comparison key: W_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 ;

    /* Create Timestamp */
    in_neword->s_O_ENTRY_D_time = time(NULL) ;

    // Determine if order is "all-local" or not
    // NOTE: This loop will exit on the iteration *after* finding the last
    // item; this effectively takes care of the 0-based/1-based conversion
    // and we don't have to add one when assigning to s_O_OL_CNT below.
    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 ;
        }
    }

    in_neword->s_O_OL_CNT = itemIndex ;

    // Sort the item list. Since invalid item IDs = 100001, we will remain
    // compliant with the spec (Section 2.4.2.3 Comment 1.

    qsort( in_neword->in_item, in_neword->s_O_OL_CNT
          , sizeof ( in_neword->in_item[ 0 ] )
          , itemComparison
          ) ;

```

```

pHostvarInput = (struct vc_new_in *) in_neword ;
pHostvarInput->len = sizeof(struct in_neword_struct) -
SPGENERAL_ADJUST ;

pHostvarOutput = (struct vc_new_out *) neword;
pHostvarOutput->len = sizeof(struct out_neword_struct) -
SPGENERAL_ADJUST ;

#ifdef DEBUGIT
    new_debug(neword, in_neword, "Client before SP call");
#endif /* DEBUGIT */

#ifdef SWAP_ENDIAN
    for (itemIndex=0; itemIndex<in_neword->s_O_OL_CNT; itemIndex++)
    {
        SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_I_ID);
        SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID);
        SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_QUANTITY);
    }
    SWAP_BYTE(in_neword->s_O_ENTRY_D_time);
    SWAP_BYTE(in_neword->s_C_ID);
    SWAP_BYTE(in_neword->s_W_ID);
    SWAP_BYTE(in_neword->s_D_ID);
    SWAP_BYTE(in_neword->s_O_OL_CNT);
    SWAP_BYTE(in_neword->s_all_local);
    SWAP_BYTE(in_neword->duplicate_items);
#endif //SWAP_ENDIAN

    EXEC SQL CALL news ( :*pHostvarInput, :*pHostvarOutput );

#ifdef SWAP_ENDIAN
    SWAP_BYTE(in_neword->s_O_ENTRY_D_time);
    SWAP_BYTE(in_neword->s_C_ID);
    SWAP_BYTE(in_neword->s_W_ID);
    SWAP_BYTE(in_neword->s_D_ID);
    SWAP_BYTE(in_neword->s_O_OL_CNT);
    SWAP_BYTE(in_neword->s_all_local);
    SWAP_BYTE(in_neword->duplicate_items);
    for (itemIndex=0; itemIndex<in_neword->s_O_OL_CNT; itemIndex++)
    {
        SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_I_ID);
        SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID);
        SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_QUANTITY);
    }

    SWAP_BYTE(neword->s_O_ENTRY_D_time);
    SWAP_BYTE(neword->s_W_TAX);
    SWAP_BYTE(neword->s_D_TAX);
    SWAP_BYTE(neword->s_C_DISCOUNT);
    SWAP_BYTE(neword->s_total_amount);
    SWAP_BYTE(neword->s_O_ID);
    SWAP_BYTE(neword->s_O_OL_CNT);
    SWAP_BYTE(neword->s_transtatus);
    SWAP_BYTE(neword->deadlocks);
    for (itemIndex=0; itemIndex<in_neword->s_O_OL_CNT; itemIndex++)
    {
        SWAP_BYTE(neword->item[ itemIndex ].s_I_PRICE);
        SWAP_BYTE(neword->item[ itemIndex ].s_OL_AMOUNT);
        SWAP_BYTE(neword->item[ itemIndex ].s_S_QUANTITY);
    }
#endif //SWAP_ENDIAN

    if ( sqlca.sqlcode == 0 )
    {
        double wtax = neword->s_W_TAX / 10000.0 ;
        double dtax = neword->s_D_TAX / 10000.0 ;
        double cdisc = neword->s_C_DISCOUNT / 10000.0 ;

```

```

double factor = (1.0 - cdisc) * (1.0 + wtax + dtax);

// Compute order total

neword->s_total_amount = 0;

for ( itemIndex = 0;
      itemIndex < in_neword->s_O_OL_CNT; // from input , not output
      itemIndex++
    )
{
    if ( neword->item[ itemIndex ].s_I_PRICE > 0 ) // A zero price signifies a
    bad item
    {
        neword->item[ itemIndex ].s_OL_AMOUNT = neword->item[
itemIndex ].s_I_PRICE *
        in_neword->in_item[ itemIndex
].s_OL_QUANTITY; // reference input value

        neword->s_total_amount += neword->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.
neword->s_total_amount *= factor;
}
else
{
    sqlerror( NEWORD_SQL, "NEW", __FILE__, __LINE__, &sqlca );
    neword->s_transtatus = FATAL_SQLERROR;
    clientRc = FATAL_SQLERROR;
}

/* Update Output Structure with Timestamp */
neword->s_O_ENTRY_D_time = in_neword->s_O_ENTRY_D_time;

#ifdef DEBUGIT
    new_debug(neword, in_neword, "Client after SP call");
#endif /* DEBUGIT */

if (neword->s_transtatus <= FATAL_SQLERROR)
{
    new_debug(neword, in_neword, "NEW failed");
    clientRc = FATAL_SQLERROR;
}

if (neword->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 )
{
    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 ;

    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 )
{
    // Propogate 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 ;
}

#ifdef 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;

#ifdef DEBUGIT
del_debug(delivery, in_delivery, "Client before SP call");
#endif /* DEBUGIT */

}

#ifdef 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 ) ;

#ifdef 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

#ifdef 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 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 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.cpp

```

// nullDB.cpp : Defines the entry point for the DLL application.
//
#include "stdafx.h"
#include "nullDB.h"
#include "..\tpcc\api\tpcc.h"
BOOL WINAPI 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';
dataSet = 1;
}
else
{
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(pymt->out_paym.s_D_STREET_1,"11900 Hobby Horse");
strcpy(pymt->out_paym.s_D_STREET_2,"Apt. 525");
strcpy(pymt->out_paym.s_D_CITY,"Valley");
strcpy(pymt->out_paym.s_D_STATE,"TX");
strcpy(pymt->out_paym.s_D_ZIP,"78559");
strcpy(pymt->out_paym.s_C_FIRST,"Jim");
strcpy(pymt->out_paym.s_C_MIDDLE,"F");
strcpy(pymt->out_paym.s_C_LAST,"Truck");
strcpy(pymt->out_paym.s_C_STREET_1,"100 N Solis");
strcpy(pymt->out_paym.s_C_STREET_2,"Flat 343");
strcpy(pymt->out_paym.s_C_CITY,"Cambridge");
strcpy(pymt->out_paym.s_C_STATE,"NY");
strcpy(pymt->out_paym.s_C_ZIP,"785585432");
strcpy(pymt->out_paym.s_C_PHONE,"1234567890123456");
pymt->out_paym.s_C_SINCE_time = 0;
strcpy(pymt->out_paym.s_C_CREDIT,"BC");

strcpy(pymt->out_paym.s_C_DATA,"XXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X");
    dataSet = 1;
}
else
{
    pymt->out_paym.s_C_CREDIT_LIM = 4000000;
    pymt->out_paym.s_C_DISCOUNT = 52400;
    pymt->out_paym.s_C_BALANCE = 14080;
    pymt->out_paym.s_C_ID = 3180;
    pymt->out_paym.s_H_DATE_time = 1234567890;
    strcpy(pymt->out_paym.s_W_STREET_1,"1201 Park Ave.");
    strcpy(pymt->out_paym.s_W_STREET_2,"Suite 432");
    strcpy(pymt->out_paym.s_W_CITY,"Denver");
    strcpy(pymt->out_paym.s_W_STATE,"CO");
    strcpy(pymt->out_paym.s_W_ZIP,"787562356");
    strcpy(pymt->out_paym.s_D_STREET_1,"3404 Garth Rd");
    strcpy(pymt->out_paym.s_D_STREET_2,"Suite 320");
    strcpy(pymt->out_paym.s_D_CITY,"Austin");
    strcpy(pymt->out_paym.s_D_STATE,"TX");
    strcpy(pymt->out_paym.s_D_ZIP,"785598767");
    strcpy(pymt->out_paym.s_C_FIRST,"John");
    strcpy(pymt->out_paym.s_C_MIDDLE,"P");
    strcpy(pymt->out_paym.s_C_LAST,"Williams");
    strcpy(pymt->out_paym.s_C_STREET_1,"North Rab Road");
    strcpy(pymt->out_paym.s_C_STREET_2,"Apt 343");
    strcpy(pymt->out_paym.s_C_CITY,"La Fiera");
    strcpy(pymt->out_paym.s_C_STATE,"TX");
    strcpy(pymt->out_paym.s_C_ZIP,"785585432");
    strcpy(pymt->out_paym.s_C_PHONE,"1234567890123456");
    pymt->out_paym.s_C_SINCE_time = 0;
    strcpy(pymt->out_paym.s_C_CREDIT,"GC");
    strcpy(pymt->out_paym.s_C_DATA,"Great Ebay");
    dataSet = 0;
}
return OK;
}
extern "C" NULLDB_API int do_ords(struct ords_wrapper *ords,void *ctx)
{
    ords->out_ords.s_transtatus = 0;
    if (dataSet == 0)
    {
        ords->out_ords.s_C_BALANCE = 100000;
        ords->out_ords.s_C_ID = 3;
        ords->out_ords.s_O_ID = 1696;
        ords->out_ords.s_O_CARRIER_ID = 9;
        ords->out_ords.s_ol_cnt = 6;
        ords->out_ords.s_O_ENTRY_D_time = 1234567890;
        strcpy(ords->out_ords.s_C_FIRST,"Homer");
        strcpy(ords->out_ords.s_C_MIDDLE,"J");
        strcpy(ords->out_ords.s_C_LAST,"Simpson");
        ords->out_ords.item[0].s_OL_AMOUNT = 30000;
        ords->out_ords.item[0].s_OL_I_ID = 23492;
        ords->out_ords.item[0].s_OL_SUPPLY_W_ID = 9;
        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 = 12300;
        ords->out_ords.item[1].s_OL_I_ID = 18860;
        ords->out_ords.item[1].s_OL_SUPPLY_W_ID = 9;
        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 = 15000;
        ords->out_ords.item[2].s_OL_I_ID = 90488;
        ords->out_ords.item[2].s_OL_SUPPLY_W_ID = 9;
        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 = 25000;
        ords->out_ords.item[3].s_OL_I_ID = 22741;
        ords->out_ords.item[3].s_OL_SUPPLY_W_ID = 9;
        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 = 20000;
        ords->out_ords.item[4].s_OL_I_ID = 92952;
        ords->out_ords.item[4].s_OL_SUPPLY_W_ID = 9;
        ords->out_ords.item[4].s_OL_QUANTITY = 5;
        ords->out_ords.item[4].s_OL_DELIVERY_D_time = 1234567890;
        ords->out_ords.item[5].s_OL_AMOUNT = 2345;
        ords->out_ords.item[5].s_OL_I_ID = 29956;
        ords->out_ords.item[5].s_OL_SUPPLY_W_ID = 9;
        ords->out_ords.item[5].s_OL_QUANTITY = 5;
        ords->out_ords.item[5].s_OL_DELIVERY_D_time = 1234567890;
        dataSet = 1;
    }
}
else
{
    ords->out_ords.s_C_BALANCE = 123000;
    ords->out_ords.s_C_ID = 856;
    ords->out_ords.s_O_ID = 418;
    ords->out_ords.s_O_CARRIER_ID = 10;
    ords->out_ords.s_ol_cnt = 5;
    strcpy(ords->out_ords.s_C_FIRST,"Erick");
    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_dlv(struct dlv_wrapper *dlv,void *ctx)
{
    dlv->out_dlv.s_transtatus = 0;
    if (dataSet == 0)
    {
        dataSet = 1;
        for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
            dlv->out_dlv.s_O_ID[districtIndex]= 2055;
    }
}

```

```

else
{
for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
    dlvy->out_dlvy.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

```

tpcclsapi/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 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];
};

```

```

////////////////////////////////////

```

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
#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

```

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
#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 <atlapi.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
#ifdef _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 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
#define TXN_NEW_ORDER_RESULTS            7
#define TXN_PAYMENT_RESULTS             8
#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 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
#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;
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
////////////////////////////////////
#define SECONDS_IN_DAY          86400
#define SECONDS_IN_HOUR        3600
#define SECONDS_IN_MINUTE      60
#define GMT_OFFSET              5
#define DAYS_IN_YEAR           365
#define YEARS_IN_LEAP          4
#define START_YEAR              1970
#define MONTHS_IN_YEAR         12
////////////////////////////////////
// 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
// 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
// page
// Parameters :
// **htmlPage
// Returns   :
// amount of characters
// the function appended

```

```

// 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)++ = ' ';   *(*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 remaining
                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 NEGITIVE_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
                        indicator
                        if(negativeFlag)
                        {
                            negativeFlag
                            = false;
                            buffer[--index] = NEGATIVE_SYMBOL;
                        }
                        else
                        {
                            // add the
                            money indicator
                            if(moneyFlag)
                            {
                                moneyFlag = false;
                                buffer[--index] = MONEY_SYMBOL;
                            }
                            else
                            {
                                // add the
                                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 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))
        {
            fileName--;
            return fileName;
        }
    }
#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 __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
// 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\nW_ID: %d CARRIER_ID: %d
%\s\nend-time: %d.%03d\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

```

tpcc/sapi/htmlPhraser.cpp

```

////////////////////////////////////
// htmlPhraser.cpp
////////////////////////////////////
// Class implementation of htmlPhraser.
// This class will take a query string and break it into a series
// of consituant parts
////////////////////////////////////
#include "htmlPhraser.h"
////////////////////////////////////
// htmlPhraser::htmlPhraser
////////////////////////////////////
// Title      : Constructor
// Parameters  : char * query string
// Return Value : None
// Comments    :
////////////////////////////////////
htmlPhraser::htmlPhraser(char *queryString)
{
    // initilize query values
    iCustomerIdFlag = iCarrierNumFlag = iStockThresholdFlag = false;
    // this initilizes 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 repersents 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;
}
///////////////////////////////////////////////////////////////////

```

tpccIsapi/StdAfx.cpp

```

// stdafx.cpp : source file that includes just the standard includes
// tpccIsapi.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/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          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 DLVYQUEUEUEDATA *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(&errorLog);
// 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

```

```

**
** Parameters      :
**
isapi txn handle
** Returns        :
**
** Comments       :
**
*****
*/
int initTxnHandle(TXN_HANDLE **txnHandle)
{
    DEBUGMSG("Inside init txn handle, getting isapiLock." << endl);
    EnterCriticalSection(&isapiLock);

    HRESULT hres = NULL;
    try
    {
        DEBUGMSG("Got isapiLock, 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

```

```

**
** 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 queue
** 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      :

```

```

** 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,(BYTE *)&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,(BYTE 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>"
" <FORM
ACTION=|"
APP_NAME
"|"
METHOD=|"GET"|>"

```

```

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">"

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>"
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">"
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"
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
** 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_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
                return OK;
            }
        }
    }
}
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"
//
"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,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 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"
//
"123456789012345678901234567890123456789012345678901234567890123
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,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> <BR>");
appendText(&html,"\\r\\n <BR> ");
html+=displayStatus(html,rc);
if(rc == OK)
    html+=sprintf(html," Total:
$%.2f",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 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 :
** 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,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>\"
                                \"
                                \"
                                \"
                                \"Amount Paid: \"
                                \"<INPUT NAME=\"
                                CMD_AMT_PAID
                                \"\|\" SIZE=10>\"
                                \" \"
                                \"New
                                \"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"
ACTION="|"
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
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 = (paym_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"
ACTION="\<BODY><FORM
APP_NAME
"
METHOD="\<GET>\>\r\n"
"<CENTER><H3>Please Fill In Payment Form.</H3></CENTER> <BR>\r\n"
"<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"
ACTION="\<BODY><FORM
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>"
Item-Num  Qty    Amount    Delivery<BR></PRE>";
"Supply-W
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"
"<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 doSetComple
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,"&#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,"<BR> <BR>"
                "Order-Number ");
    appendText(&html,itoa(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,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 qty
appendText(&html,itoa(ords->out_ords.item[itemCount].s_OL_QUANTITY,bu
ffer,10),6,1);
        //get item dollar 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"
                "<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>");
        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
ACTION=\\"
        APP_NAME
        "\\"
METHOD=\\"GET\ ">\r\n"
        "<CENTER><H3>Delivery.</H3></CENTER>\r\n"
        "Submit Transaction
        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
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_dlvvy.s_W_ID = txnHandle->w_id;
    //set the carrier id from command blk
    if( dlvy.in_dlvvy.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_dlvvy.s_W_ID,dlvy.in_dlvvy.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(dlv.in_dlvvy.s_W_ID,buffer,10));
    appendText(&html,"<BR><BR>Carrier Number: ");

    //get carrier_id from wrapper

    appendText(&html,itoa(dlv.in_dlvvy.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

```

```

        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"
ACTION=|"|"
APP_NAME
|"|"
METHOD=|"GET"|>\r\n"
"<CENTER><H3>Please Fill In Stock Form.</H3></CENTER><BR>\r\n"
"Submit Transaction
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
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, itoa(stok->in_stok.s_W_ID, buffer, 10), 6+1, 1);
    appendText(&html, "District: ");
    appendText(&html, itoa(stok->in_stok.s_D_ID, buffer, 10));

    appendText(&html, "<BR> <BR>"
                "Stock Level
Threshold: ");
    appendText(&html, itoa(stok->in_stok.s_threshold, buffer, 10));
    appendText(&html, "<BR> <BR>"
                "Low Stock: ");
    appendText(&html, itoa(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,ittoa(txnHandle->w_id,buffer,10));
    appendSpaces(&html,2);
    appendText(&html,"District: ");
    appendText(&html,commandBlock->get_D_ID());
    appendText(&html," <BR> <BR>"
"Stock Level
" <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
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_DLVY
                "\">\r\n"
                "<INPUT
TYPE=\"submit\" NAME=\""
                CMD_TXN_ID
                "\" VALUE=\""
                CMD_EXIT
                "\">\r\n<BR>");
}
/*
*****
** 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>
<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      :
**               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
        {
            DEBUGMSG("dlvyThread initializing wait
handles" << endl);
            //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,"%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 terminal id:"<<terminalID<<endl);
    DEBUGMSG("User sent in a terminal id:"<<terminalID<<endl, 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/Winx64 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 /MD -GS- -DWIN64 -J -Zp8
-DREG_KIT_METHOD
CFLAGS_OUT=/Fo
CFLAGS_DEBUG=

# Linker Configuration (MSVC)
LD_EXEC=link.exe
LD_STORP=link.exe
LD_FLAGS_EXEC=
LD_FLAGS_SHLIB=/DLL
LD_FLAGS_STORP=$(LD_FLAGS_SHLIB) /DEF:rptpc.def
LD_FLAGS_LIB=/LIBPATH:$(TPCC_SQLLIB)\lib
/LIBPATH:"C:\MsSDKx64\lib\amd64" db2api.lib winmm.lib
LD_FLAGS_OUT=/OUT:

# Library Configuration
AR=lib.exe
AR_FLAGS=
AR_FLAGS_LIB=
AR_FLAGS_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=CK050901

@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=%HOME%\tpc-c.ibm
set TPCC_SQLLIB=%HOME%\sqllib
set TPCC_RUNDATA=%HOME%\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 - 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 "Ival.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/lval.h

#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 21760
#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 - 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
 */

#ifndef __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
 */

#ifndef __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 - 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 =      tpccmisc$(OBJEXT) tpccdbg$(OBJEXT)
UTIL_OBJ_DB2 =  tpccctx$(OBJEXT)

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

all:            dbgen connect $(UTIL_OBJ_DB2) disconnect

dbgen:          $(UTIL_OBJ)

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

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

connect:        - db2 connect to $(TPCC_DBNAME)

disconnect:     - db2 connect reset
                - db2 terminate

rebind:         connect
                db2 bind tpccctx.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 - 2005
** 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 <string.h>
#include <sqlutil.h>
#include "db2tpcc.h"
#include "tpccdbg.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; }

    /* Copy 9 characters - 8 for dbname, 1 for NULL */
    strncpy(dbname,in_dbname,9);
    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;

    sqleSetTypeCtx(SQL_CTX_MULTI_MANUAL);
    sqleBeginCtx(&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;

    sqleAttachToCtx(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;

    sqlcGetCurrentCtx(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 - 2005
** 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);

if ((err_fp = fopen(err_fn, "a+")) == NULL)
{
    return;
}

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", psqca->sqlerrp[j]);
fprintf(err_fp, "\n");

fprintf(err_fp, "sqlerrd: ");
for(j = 0; j < 6; j++)
    fprintf(err_fp, "%d", psqca->sqlerrd[j]);
fprintf(err_fp, "\n");

if (psqca->sqlwarn[0] != ' ')
{
    fprintf(err_fp, "sqlwarn: ");
    for(j = 0; j < 8; j++)
        fprintf(err_fp, "%c ", psqca->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\\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, "\\t\\ts_O_ID[%d] = %d\\n",
            j, delivery_ptr->s_O_ID[j]);
}
fprintf(debug_fp, "\\t\\n\\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, "\\t\\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, "\\t\\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, "\\t\\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);
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",

```



```

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=====
=====\\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");

    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");
    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 - 2005
** 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 <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);
}

```

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=300000

plan:
                - db2exfmt -d $(TPCC_DBNAME) -e $(TPCC_SCHEMA) -s
                $(TPCC_SCHEMA) -w -l -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

# 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 - 2005
-- 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)
  )
;

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_LAST )
= ( SELECT C_BALANCE, C_FIRST, C_MIDDLE, C_LAST

    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

    , OL_DIST_INFO = CASE D_ID WHEN
SMALLINT( 1 ) THEN S_DIST_01
    THEN S_DIST_02
    WHEN SMALLINT( 2 )

```

```

    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:\sqlib\function\|rpctpc!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:\sqlib\function\|rpctpc!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:\sqlib\function\|rpctpc!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 - 2005
** 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

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_neword->in_item[ 12 ].s_OL_QUANTITY
#define ol_quantity13 in_neword->in_item[ 13 ].s_OL_QUANTITY
#define ol_quantity14 in_neword->in_item[ 14 ].s_OL_QUANTITY

#define supply_w_id0 in_neword->in_item[ 0 ].s_OL_SUPPLY_W_ID
#define supply_w_id1 in_neword->in_item[ 1 ].s_OL_SUPPLY_W_ID
#define supply_w_id2 in_neword->in_item[ 2 ].s_OL_SUPPLY_W_ID
#define supply_w_id3 in_neword->in_item[ 3 ].s_OL_SUPPLY_W_ID
#define supply_w_id4 in_neword->in_item[ 4 ].s_OL_SUPPLY_W_ID
#define supply_w_id5 in_neword->in_item[ 5 ].s_OL_SUPPLY_W_ID
#define supply_w_id6 in_neword->in_item[ 6 ].s_OL_SUPPLY_W_ID
#define supply_w_id7 in_neword->in_item[ 7 ].s_OL_SUPPLY_W_ID
#define supply_w_id8 in_neword->in_item[ 8 ].s_OL_SUPPLY_W_ID
#define supply_w_id9 in_neword->in_item[ 9 ].s_OL_SUPPLY_W_ID
#define supply_w_id10 in_neword->in_item[ 10 ].s_OL_SUPPLY_W_ID
#define supply_w_id11 in_neword->in_item[ 11 ].s_OL_SUPPLY_W_ID
#define supply_w_id12 in_neword->in_item[ 12 ].s_OL_SUPPLY_W_ID
#define supply_w_id13 in_neword->in_item[ 13 ].s_OL_SUPPLY_W_ID
#define supply_w_id14 in_neword->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 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
      ) 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
  ,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 )
      ) 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_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

, :supply_w_id0 )
, ( SMALLINT( 1 ) , :id0 , :ol_quantity0
, :supply_w_id1 )
, ( SMALLINT( 2 ) , :id1 , :ol_quantity1
, :supply_w_id2 )
, ( SMALLINT( 3 ) , :id2 , :ol_quantity2
, :supply_w_id3 )
, ( SMALLINT( 4 ) , :id3 , :ol_quantity3
, :supply_w_id4 )
, ( SMALLINT( 5 ) , :id4 , :ol_quantity4

```

```

, ( 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
,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 )

) 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_8 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 )

) 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_9 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 )

) 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
, :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 )
) 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_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
, 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 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_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
        ,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 )
        ) 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
      , 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 )
              ) 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
    ,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 )
        ) 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
, 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 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_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 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_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 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_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_newword = (struct in_newword_struct *) pin ;
newword = (struct out_newword_struct *) pout ;

#ifdef DEBUGIT
new_debug( newword, in_newword, "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_newword->s_O_OL_CNT ; inputItemArrayIndex++ )
{
i_priceArray[ inputItemArrayIndex ] = 0 ;
}

newword->deadlocks = -1 ;

retry_tran:

newword->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 ferror;
}

#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 ( newword->s_O_OL_CNT == in_newword->s_O_OL_CNT )
  {
    newword->s_transtatus = TRAN_OK ;

    EXEC SQL COMMIT;

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

    EXEC SQL ROLLBACK WORK ;

    if ( sqlca.sqlcode != 0 )
    {
      newword->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( newword, in_newword, "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:

newword->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*/
                                8,8,8,8,1,8,8,8,8, /*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*/
                                8,8,8,8,8,8,8,8,8, /*250-254*/
                                8,8,8,8,8,8,8,8,8, /*255*/
}

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_id
#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;

###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 ;

#ifdef DEBUGIT
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

    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 )
    {
        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/rpctpc.def

```
LIBRARY rpctpc
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,
EXTENTSIZE 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,
TBCREATOR 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 - 2005
-- 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);
    }
    STDMETHODCALLTYPE(UnregisterAll())
    {
        _AtlComModule.UnregisterServer(TRUE);
        return S_OK;
    }
    STDMETHODCALLTYPE(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);
                pBSTR = OLE2BSTR(pszCLSID);
                if( pBSTR
== NULL )
                {
                    CoTaskMemFree(pszCLSID);
                    return E_OUTOFMEMORY;
                }
                HRESULT
hResult = SafeArrayPutElement(*ppCLSIDs, &i, pBSTR);
                CoTaskMemFree(pszCLSID);
                if( FAILED(hResult) )
                {
                    return hResult;
                }
                T2BSTR_EX(pszDescription);
                if( pBSTR
== NULL )
                {
                    return E_OUTOFMEMORY;
                }
                HRESULT
hResult =
                SafeArrayPutElement(*ppDescriptions, &i, pBSTR);
                if( FAILED(hResult) )
                {
                    return hResult;
                }
                i++;
            }
        }
    }
}
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.

```

```

#ifndef 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
#ifndef _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
#ifndef _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
#ifndef _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 //open comLog
#define _ATL_NO_AUTOMATIC_NAMESPACE //open comLog
#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;
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<IOBJECTCONTROL> 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/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 "..\tpccIsapi\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(Itppc_com)
    COM_INTERFACE_ENTRY(IObjectControl)
END_COM_MAP()
// IObjectControl
public:
    STDMETHOD(Activate)();
    STDMETHOD_(BOOL, CanBePooled)();
    STDMETHOD_(void, Deactivate)();
    CComPtr<IObjectContext> m_spObjectContext;

// Itppc_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 "oidl.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)* pbstrCLSIDs, [out] SAFEARRAY(BSTR)*
pbstrDescriptions);
    [id(5)] HRESULT RegisterComponent([in] BSTR bstrCLSID);
    [id(6)] HRESULT UnregisterComponent([in] BSTR bstrCLSID);
};
[
    object,
    uuid(5B4FA473-2E68-4D79-A626-F38B30B8196E),
    helpstring("Itppc_com Interface"),
    pointer_default(unique)
]
interface Itppc_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 "compreg.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 "compreg.h"
#include "dldatax.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)
*****
*/
STDMETHODIMP 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
** Returns       :
**               UCHAR**
char buffer that holds txn wrapper struct
** Comments      :
**               int - return code
*****
*/
STDMETHODIMP 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     :
**
*****
*/
STDMETHODIMP 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 "dlldata.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( )
#ifdef !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>
#undef INITGUID
#else
#include <guiddef.h>
#endif
#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
    DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)
#else // !_MIDL_USE_GUIDDEF_
#ifndef __IID_DEFINED__
#define __IID_DEFINED__
typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char c[8];
} IID;
#endif // __IID_DEFINED__
#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED
#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
    const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}
#endif // !_MIDL_USE_GUIDDEF_
MIDL_DEFINE_GUID(IID,
IID_IComponentRegistrar,0xa817e7a2,0x43fa,0x11d0,0x9e,0x44,0x00,0xaa,0x00,0xb6,0x77,0x0a);

MIDL_DEFINE_GUID(IID,
IID_Itpcc_com,0x5B4FA473,0x2E68,0x4D79,0xA6,0x26,0xF3,0x8B,0x30,0xB8,0x19,0x6E);

MIDL_DEFINE_GUID(IID,
LIBID_tpccComLib,0x91F1B8B0,0x89E9,0x457B,0xA2,0x28,0x3E,0x2D,0x6C,0xE3,0xE7,0x52);

MIDL_DEFINE_GUID(CLSID,
CLSID_CompReg,0x90EEDAFF,0xF8D3,0x4711,0x99,0xA9,0x8A,0xC3,0xC0,0xFE,0x5D,0xB9);

MIDL_DEFINE_GUID(CLSID,
CLSID_tpcc_com,0x5F752BF2,0xF739,0x43D4,0x84,0x92,0x44,0xC1,0x95,0x81,0xC0,0xA1);
#undef MIDL_DEFINE_GUID
#ifdef __cplusplus
}
#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 */
#if _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
*/

#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif

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

#include "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,0x10,0x48,0x6
0}},{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 =
{
    0,
    {
        /* 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 GetComponent */
/* 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, */
/* 0x3, /* 3 */
/* 112 */ 0x8, /* 8 */
/* 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 */
/* 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 */
/* Procedure doNewOrder */
/* 252 */ 0x33, /* FC_AUTO_HANDLE */
/* 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, */
/* 268 */ 0x8, /* 3 */
/* 8 */
/* 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 */
/* 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,
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 */
/* 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 */
/* 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 */
/* Procedure doOrderStatus */
/* 336 */ 0x33, /* FC_AUTO_HANDLE */
/* 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 */
/* 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 */
/* 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 */
/* Procedure doDBInfo */
/* 378 */ 0x33, /* FC_AUTO_HANDLE */
/* 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, */
/* 394 */ 0x8, /* 1 */
/* 8 */
/* 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 */
/* 408 */ 0x0, /* 0 */

```

```

    }
};
static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
    0,
    {
        /* 2 */
        NdrFcShort( 0x0 ), /* 0 */

        /* 4 */ NdrFcShort( 0xe ), /* Offset= 14 (18) */
        /* 6 */

        /* 8 */ NdrFcShort( 0x2 ), /* 2 */
        /* 10 */ 0x9, /* Corr desc: FC_ULONG */
        /* 12 */ NdrFcShort( 0xffc ), /* -4 */
        /* 14 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
        /* 16 */ 0x6, /* FC_SHORT */
        /* 18 */ 0x5b, /* FC_END */

        /* 20 */ NdrFcShort( 0x8 ), /* 8 */
        /* 22 */ NdrFcShort( 0xff0 ), /* Offset= -16 (6) */
        /* 24 */ 0x8, /* FC_LONG */
        /* 26 */ 0x5c, /* FC_PAD */
        /* 28 */ 0xb4, /* FC_USER_MARSHAL */
        /* 30 */ NdrFcShort( 0x0 ), /* 0 */
        /* 32 */ NdrFcShort( 0x4 ), /* 4 */
        /* 34 */ NdrFcShort( 0x0 ), /* 0 */
        /* 36 */ NdrFcShort( 0xffde ), /* Offset= -34 (2) */
        /* 38 */

        /* 40 */ NdrFcShort( 0x3f6 ), /* Offset= 1014 (1054) */
        /* 42 */

        /* 44 */ NdrFcShort( 0x2 ), /* Offset= 2 (46) */
        /* 46 */

        /* 48 */ NdrFcShort( 0x3dc ), /* Offset= 988 (1036) */
        /* 50 */

        /* 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 */

        /* 120 */ NdrFcShort( 0x4 ), /* 4 */
        /* 122 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
        /* 124 */ NdrFcShort( 0x0 ), /* 0 */
        /* 126 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
        /* 128 */

        /* 130 */

        /* 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 */

        /* 148 */ 0x5c, /* FC_PAD */
        /* 150 */

        /* 152 */ NdrFcShort( 0x8 ), /* 8 */
        /* 154 */

        /* 156 */

        /* 158 */ NdrFcShort( 0x4 ), /* 4 */
        /* 160 */ NdrFcShort( 0x4 ), /* 4 */
        /* 162 */ 0x11, 0x0, /* FC_RP */
        /* 164 */ NdrFcShort( 0xffd2 ), /* Offset= -46 (118) */
        /* 166 */

        /* 168 */ 0x8, /* FC_LONG */
        /* 170 */

        /* 172 */ NdrFcLong( 0x0 ), /* 0 */
        /* 176 */ NdrFcShort( 0x0 ), /* 0 */
        /* 178 */ NdrFcShort( 0x0 ), /* 0 */
        /* 180 */ 0xc0, /* 192 */
        /* 182 */ 0x0, /* 0 */
        /* 184 */ 0x0, /* 0 */
        /* 186 */ 0x0, /* 0 */
        /* 188 */

        /* 190 */ 0x1b, /* FC_CARRAY */
        /* 192 */ 0x3, /* 3 */

        /* 194 */ 0x0, /* 0 */
        /* 196 */ 0x4b, /* FC_PP */
        /* 198 */ 0x5c, /* FC_PAD */

        /* 200 */ 0x48, /* 48 */
        /* 202 */ 0x49, /* 49 */

        /* 204 */ FC_VARIABLE_REPEAT /* FC_VARIABLE_REPEAT */
        /* 206 */ FC_FIXED_OFFSET /* FC_FIXED_OFFSET */
        /* 208 */ 0x49, /* 49 */
        /* 210 */ 0x5b, /* FC_END */
        /* 212 */ 0x8, /* FC_LONG */
        /* 214 */ 0x5b, /* FC_PAD */
        /* 216 */ 0x16, /* FC_PSTRUCT */
        /* 218 */ 0x3, /* 3 */
        /* 220 */ NdrFcShort( 0x8 ), /* 8 */
        /* 222 */ 0x4b, /* FC_PP */
        /* 224 */ 0x5c, /* FC_PAD */
        /* 226 */ 0x46, /* FC_NO_REPEAT */
        /* 228 */ 0x5c, /* FC_PAD */
        /* 230 */ NdrFcShort( 0x4 ), /* 4 */
        /* 232 */ NdrFcShort( 0x4 ), /* 4 */
        /* 234 */ 0x11, 0x0, /* FC_RP */
        /* 236 */ NdrFcShort( 0xffd2 ), /* Offset= -46 (118) */
        /* 238 */ 0x5b, /* FC_END */
        /* 240 */ 0x8, /* FC_LONG */
        /* 242 */ 0x5b, /* FC_END */
        /* 244 */ 0x2f, /* FC_IP */
        /* 246 */ 0x5a, /* 5a */

        /* 248 */ FC_CONSTANT_IID /* FC_CONSTANT_IID */
        /* 250 */ NdrFcLong( 0x0 ), /* 0 */
        /* 252 */ NdrFcShort( 0x0 ), /* 0 */
        /* 254 */ NdrFcShort( 0x0 ), /* 0 */
        /* 256 */ 0xc0, /* 192 */
        /* 258 */ 0x0, /* 0 */
        /* 260 */ 0x0, /* 0 */
        /* 262 */ 0x0, /* 0 */
        /* 264 */ 0x0, /* 0 */
        /* 266 */ 0x46, /* 70 */
        /* 268 */
        /* 270 */ 0x46, /* 70 */
    }
};

```


FC_BOGUS_ARRAY */	0x21,	/*	0x36,	/* FC_POINTER */
/* 190 */ NdrFcShort(0x0),	/* 0 */	/* 3 */	/* 276 */ 0x5c,	/* FC_PAD */
/* 192 */ 0x19,	/* Corr desc: field pointer, FC_ULONG */		0x5b,	/* FC_END */
/* 194 */ NdrFcShort(0x0),	/* 0 */		0x11, 0x0, /* FC_RP */	
/* 196 */ NdrFcShort(0x1),	/* Corr flags: early, */		/* 280 */ NdrFcShort(0xffdc), /* Offset= -36 (244) */	
/* 198 */ NdrFcLong(0xffffffff),	/* -1 */		/* 282 */	
/* 202 */ NdrFcShort(0x0),	/* Corr flags: */		0x2b,	/*
/* 204 */ 0x4c,	/* FC_EMBEDDED_COMPLEX */		FC_NON_ENCAPSULATED_UNION */	
/* 206 */ NdrFcShort(0xffdc), /* Offset= -36 (170) */	/* 0 */		0x9,	/* FC_ULONG */
/* 208 */ 0x5c,	/* FC_PAD */		/* 284 */ 0x7,	/* Corr desc: FC_USHORT */
/* 210 */	0x5b,	/* FC_END */	0x0,	/* */
FC_BOGUS_STRUCT */	0x1a,	/*	/* 286 */ NdrFcShort(0xffff8), /* -8 */	
/* 212 */ NdrFcShort(0x8),	/* 3 */		/* 288 */ NdrFcShort(0x1), /* Corr flags: early, */	
/* 214 */ NdrFcShort(0x0),	/* 8 */		/* 290 */ NdrFcShort(0x2), /* Offset= 2 (292) */	
/* 216 */ NdrFcShort(0x6),	/* Offset= 6 (222) */		/* 292 */ NdrFcShort(0x10), /* 16 */	
/* 218 */ 0x8,	/* FC_LONG */		/* 294 */ NdrFcShort(0x2f), /* 47 */	
/* 220 */ 0x5c,	0x36,	/* FC_POINTER */	/* 296 */ NdrFcLong(0x14), /* 20 */	
/* 222 */	0x5b,	/* FC_END */	/* 300 */ NdrFcShort(0x800b), /* Simple arm type: FC_HYPER */	
/* 224 */ NdrFcShort(0xffdc), /* Offset= -36 (188) */	0x11, 0x0, /* FC_RP */		/* 302 */ NdrFcLong(0x3), /* 3 */	
/* 226 */	/* FC_PAD */		/* 306 */ NdrFcShort(0x8008), /* Simple arm type: FC_LONG */	
0x2f,	/* FC_IP */		/* 308 */ NdrFcLong(0x11), /* 17 */	
0x5a,	/*		/* 312 */ NdrFcShort(0x8001), /* Simple arm type: FC_BYTE */	
FC_CONSTANT_IID */	/* 132096 */		/* 314 */ NdrFcLong(0x2), /* 2 */	
/* 228 */ NdrFcLong(0x20400),	/* 0 */		/* 318 */ NdrFcShort(0x8006), /* Simple arm type: FC_SHORT */	
/* 232 */ NdrFcShort(0x0),	/* 0 */		/* 320 */ NdrFcLong(0x4), /* 4 */	
/* 234 */ NdrFcShort(0x0),	/* 0 */		/* 324 */ NdrFcShort(0x800a), /* Simple arm type: FC_FLOAT */	
/* 236 */ 0xc0,	/* 192 */		/* 326 */ NdrFcLong(0x5), /* 5 */	
/* 238 */ 0x0,	/* 0 */		/* 330 */ NdrFcShort(0x800c), /* Simple arm type: FC_DOUBLE	
/* 240 */ 0x0,	/* 0 */		/*	
/* 242 */ 0x0,	/* 0 */		/* 332 */ NdrFcLong(0xb), /* 11 */	
/* 244 */	0x46,	/* 70 */	/* 336 */ NdrFcShort(0x8006), /* Simple arm type: FC_SHORT */	
FC_BOGUS_ARRAY */	0x21,	/*	/* 338 */ NdrFcLong(0xa), /* 10 */	
/* 246 */ NdrFcShort(0x0),	/* 3 */		/* 342 */ NdrFcShort(0x8008), /* Simple arm type: FC_LONG */	
/* 248 */ 0x19,	/* Corr desc: field pointer, FC_ULONG */		/* 344 */ NdrFcLong(0x6), /* 6 */	
/* 250 */ NdrFcShort(0x0),	/* 0 */		/* 348 */ NdrFcShort(0xe8), /* Offset= 232 (580) */	
/* 252 */ NdrFcShort(0x1),	/* Corr flags: early, */		/* 350 */ NdrFcLong(0x7), /* 7 */	
/* 254 */ NdrFcLong(0xffffffff),	/* -1 */		/* 354 */ NdrFcShort(0x800c), /* Simple arm type: FC_DOUBLE	
/* 258 */ NdrFcShort(0x0),	/* Corr flags: */		/*	
/* 260 */ 0x4c,	/* FC_EMBEDDED_COMPLEX */		/* 356 */ NdrFcLong(0x8), /* 8 */	
/* 262 */ NdrFcShort(0xffdc), /* Offset= -36 (226) */	/* 0 */		/* 360 */ NdrFcShort(0xe2), /* Offset= 226 (586) */	
/* 264 */ 0x5c,	/* FC_PAD */		/* 362 */ NdrFcLong(0xd), /* 13 */	
/* 266 */	0x5b,	/* FC_END */	/* 366 */ NdrFcShort(0xff3c), /* Offset= -196 (170) */	
FC_BOGUS_STRUCT */	0x1a,	/*	/* 368 */ NdrFcLong(0x9), /* 9 */	
/* 268 */ NdrFcShort(0x8),	/* 3 */		/* 372 */ NdrFcShort(0xff6e), /* Offset= -146 (226) */	
/* 270 */ NdrFcShort(0x0),	/* 8 */		/* 374 */ NdrFcLong(0x2000), /* 8192 */	
/* 272 */ NdrFcShort(0x6),	/* Offset= 6 (278) */		/* 378 */ NdrFcShort(0xd4), /* Offset= 212 (590) */	
/* 274 */ 0x8,	/* FC_LONG */		/* 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 */	


```

                                0x13, 0x10,          /* FC_OP
[pointer_deref] */
/* 682 */ NdrFcShort( 0xffa0 ), /* Offset= -96 (586) */
/* 684 */
                                0x13, 0x10,          /* FC_OP
[pointer_deref] */
/* 686 */ NdrFcShort( 0xffdc ), /* 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( 0xfffd ), /* Offset= -12 (712) */
/* 726 */
                                0x13, 0x8, /* FC_OP [simple_pointer] */
/* 728 */ 0x2,          /* FC_CHAR */
                                0x5c,          /* FC_PAD */
/* 730 */
FC_BOGUS_STRUCT */
                                0x1a,          /*
/* 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 */

```

FC_BOGUS_STRUCT */	0x1a,	/*	/* 904 */ NdrFcShort(0x4),	/* 4 */
/* 832 */ NdrFcShort(0x8),	0x3,	/* 3 */	/* 906 */ NdrFcShort(0x4),	/* 4 */
/* 834 */ NdrFcShort(0x0),	/* 8 */		/* 908 */ 0x13, 0x0, /* FC_OP */	
/* 836 */ NdrFcShort(0x6),	/* 0 */		/* 910 */ NdrFcShort(0xffe6), /* Offset= -26 (884) */	
/* 838 */ 0x8,	/* Offset= 6 (842) */		/* 912 */	
/* 840 */ 0x5c,	/* FC_LONG */		0x5b,	/* FC_END */
/* 842 */	0x36,	/* FC_POINTER */	0x8,	/* FC_LONG */
/* 844 */ NdrFcShort(0xffd2), /* Offset= -46 (798) */	/* FC_PAD */		0x5b,	/* FC_END */
/* 846 */	0x5b,	/* FC_END */	/* 916 */	
/* 848 */ NdrFcShort(0x8),	0x11, 0x0, /* FC_RP */		0x1b,	/* FC_CARRAY */
/* 850 */ 0x1,	/* 8 */		0x1,	/* 1 */
/* 852 */	/* FC_BYTE */		/* 918 */ NdrFcShort(0x2),	/* 2 */
/* 854 */ NdrFcShort(0x10),	0x1d,	/* FC_SMFARRAY */	/* 920 */ 0x19,	/* Corr desc: field pointer, FC_ULONG */
/* 856 */ 0x8,	0x0,	/* 0 */	/* 922 */ NdrFcShort(0x0),	/* 0 */
/* 858 */ 0x6,	/* FC_END */		/* 924 */ NdrFcShort(0x1),	/* Corr flags: early, */
FC_EMBEDDED_COMPLEX */	0x5b,	/* FC_END */	/* 926 */ 0x6,	/* FC_SHORT */
/* 860 */ 0x0,	0x15,	/* FC_STRUCT */	/* 928 */	0x5b,
/* 864 */	0x3,	/* 3 */	/* 930 */ NdrFcShort(0x8),	/* FC_PSTRUCT */
/* 866 */ NdrFcShort(0x18),	/* 16 */		/* 932 */	0x3,
/* 868 */ NdrFcShort(0x0),	/* FC_LONG */		/* 934 */	0x4b,
/* 870 */ NdrFcShort(0xa),	0x6,	/* FC_SHORT */	/* 936 */ NdrFcShort(0x4),	/* FC_PP */
/* 872 */ 0x8,	0x4c,	/*	/* 938 */ NdrFcShort(0x4),	/* FC_PAD */
/* 874 */ 0x4c,	/* FC_END */		/* 940 */ 0x13, 0x0, /* FC_OP */	
/* 876 */ NdrFcShort(0xffe8), /* Offset= -24 (852) */	0x1a,	/*	/* 942 */ NdrFcShort(0xffe6), /* Offset= -26 (916) */	
/* 878 */ 0x5c,	0x3,	/* 3 */	/* 944 */	
/* 880 */	/* FC_END */		0x5b,	/* FC_END */
/* 882 */ NdrFcShort(0xfd4a), /* Offset= -694 (188) */	0x1a,	/*	0x8,	/* FC_LONG */
/* 884 */	0x3,	/* 3 */	/* 948 */	0x5b,
/* 886 */ NdrFcShort(0x1),	0x0,	/* FC_CARRAY */	/* 950 */ NdrFcShort(0x4),	/* FC_END */
/* 888 */ 0x19,	/* 0 */		/* 952 */ 0x19,	/* FC_LONG */
/* 890 */ NdrFcShort(0x0),	/* Offset= 10 (880) */		/* 954 */ NdrFcShort(0x0),	/* 0 */
/* 892 */ NdrFcShort(0x1),	/* FC_LONG */		/* 956 */ NdrFcShort(0x1),	/* Corr flags: early, */
/* 894 */ 0x1,	/* FC_POINTER */		/* 958 */ 0x8,	/* FC_LONG */
/* 896 */	/* FC_EMBEDDED_COMPLEX */		/* 960 */	0x5b,
/* 898 */ NdrFcShort(0x8),	0x0,	/* 0 */	/* 962 */ NdrFcShort(0x8),	/* FC_PSTRUCT */
/* 900 */	/* FC_PAD */		/* 964 */	0x3,
/* 902 */	0x5b,	/* FC_END */	/* 966 */	0x4b,
/* 904 */	0x11, 0x0, /* FC_RP */		/* 968 */ NdrFcShort(0x4),	/* FC_PP */
/* 906 */	/* Offset= -24 (852) */		/* 970 */ NdrFcShort(0x4),	/* FC_PAD */
/* 908 */	/* FC_POINTER */		/* 972 */ 0x13, 0x0, /* FC_OP */	
/* 910 */	/* FC_PAD */		/* 974 */ NdrFcShort(0xffe6), /* Offset= -26 (948) */	
/* 912 */	0x5b,	/* FC_END */	/* 976 */	
/* 914 */	0x1b,	/* FC_CARRAY */	0x5b,	/* FC_END */
/* 916 */	0x0,	/* 0 */	0x8,	/* FC_LONG */
/* 918 */	/* Corr desc: field pointer, FC_ULONG */		/* 978 */ 0x8,	/* FC_LONG */
/* 920 */	/* 0 */		/* 980 */	0x5b,
/* 922 */	/* Corr flags: early, */		0x5b,	/* FC_END */
/* 924 */	/* FC_BYTE */		0x8,	/* FC_LONG */
/* 926 */	/* FC_END */		/* 982 */	0x5b,
/* 928 */	0x16,	/* FC_PSTRUCT */	/* 984 */	0x8,
/* 930 */	0x3,	/* 3 */	/* 986 */	0x5b,
/* 932 */	/* FC_PP */		/* 988 */	0x8,
/* 934 */	/* FC_PAD */		/* 990 */	0x5b,
/* 936 */	/* FC_NO_REPEAT */		/* 992 */	0x8,
/* 938 */	/* FC_PAD */		/* 994 */	0x5b,
/* 940 */	0x46,	/* FC_NO_REPEAT */	/* 996 */	0x8,
/* 942 */	0x5c,	/* FC_PAD */	/* 998 */	0x5b,
/* 944 */	0x4b,	/* FC_PP */	/* 1000 */	0x8,
/* 946 */	0x5c,	/* FC_PAD */	/* 1002 */	0x5b,
/* 948 */	0x46,	/* FC_NO_REPEAT */	/* 1004 */	0x8,
/* 950 */	0x5c,	/* FC_PAD */	/* 1006 */	0x5b,
/* 952 */	0x4b,	/* FC_PP */	/* 1008 */	0x8,
/* 954 */	0x5c,	/* FC_PAD */	/* 1010 */	0x5b,
/* 956 */	0x46,	/* FC_NO_REPEAT */	/* 1012 */	0x8,
/* 958 */	0x5c,	/* FC_PAD */	/* 1014 */	0x5b,
/* 960 */	0x4b,	/* FC_PP */	/* 1016 */	0x8,
/* 962 */	0x5c,	/* FC_PAD */	/* 1018 */	0x5b,
/* 964 */	0x46,	/* FC_NO_REPEAT */	/* 1020 */	0x8,
/* 966 */	0x5c,	/* FC_PAD */	/* 1022 */	0x5b,
/* 968 */	0x4b,	/* FC_PP */	/* 1024 */	0x8,
/* 970 */	0x5c,	/* FC_PAD */	/* 1026 */	0x5b,
/* 972 */	0x46,	/* FC_NO_REPEAT */	/* 1028 */	0x8,
/* 974 */	0x5c,	/* FC_PAD */	/* 1030 */	0x5b,
/* 976 */	0x4b,	/* FC_PP */	/* 1032 */	0x8,
/* 978 */	0x5c,	/* FC_PAD */	/* 1034 */	0x5b,
/* 980 */	0x46,	/* FC_NO_REPEAT */	/* 1036 */	0x8,
/* 982 */	0x5c,	/* FC_PAD */	/* 1038 */	0x5b,
/* 984 */	0x4b,	/* FC_PP */	/* 1040 */	0x8,
/* 986 */	0x5c,	/* FC_PAD */	/* 1042 */	0x5b,
/* 988 */	0x46,	/* FC_NO_REPEAT */	/* 1044 */	0x8,
/* 990 */	0x5c,	/* FC_PAD */	/* 1046 */	0x5b,
/* 992 */	0x4b,	/* FC_PP */	/* 1048 */	0x8,
/* 994 */	0x5c,	/* FC_PAD */	/* 1050 */	0x5b,
/* 996 */	0x46,	/* FC_NO_REPEAT */	/* 1052 */	0x8,
/* 998 */	0x5c,	/* FC_PAD */	/* 1054 */	0x5b,
/* 1000 */	0x4b,	/* FC_PP */	/* 1056 */	0x8,

```

0x1b,          /* FC_CARRAY */
/* 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 */
/* 994 */ NdrFcShort( 0x8 ), /* 8 */
/* 996 */ 0x16, /* FC_PSTRUCT */
/* 998 */ 0x3, /* 3 */
/* 999 */ 0x4b, /* FC_PP */
/* 1000 */ 0x5c, /* FC_PAD */
/* 1001 */ 0x46, /* FC_NO_REPEAT */
/* 1002 */ 0x5c, /* FC_PAD */
/* 1003 */ 0x4, /* 4 */
/* 1004 */ NdrFcShort( 0x4 ), /* 4 */
/* 1005 */ 0x13, 0x0, /* FC_OP */
/* 1006 */ NdrFcShort( 0xffe6 ), /* Offset= -26 (980) */
/* 1008 */ 0x5b, /* FC_END */
/* 1010 */ 0x8, /* FC_LONG */
/* 1012 */ 0x5b, /* FC_END */
/* 1014 */ NdrFcShort( 0x8 ), /* 8 */
/* 1016 */ 0x8, /* FC_LONG */
/* 1018 */ 0x5c, /* FC_PAD */
/* 1020 */ 0x5b, /* FC_END */
/* 1022 */ NdrFcShort( 0x8 ), /* 8 */
/* 1024 */ 0x7, /* Corr desc: FC_USHORT */
/* 1026 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 1028 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 1030 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 1032 */ NdrFcShort( 0xffec ), /* Offset= -20 (1012) */
/* 1034 */ 0x5c, /* FC_PAD */
/* 1036 */ 0x5b, /* FC_END */
FC_BOGUS_STRUCT /*
/* 1038 */ NdrFcShort( 0x28 ), /* 40 */
/* 1040 */ NdrFcShort( 0xffec ), /* Offset= -20 (1020) */
/* 1042 */ NdrFcShort( 0x0 ), /* Offset= 0 (1042) */
/* 1044 */ 0x6, /* FC_SHORT */
/* 1046 */ 0x8, /* FC_LONG */
/* 1048 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 1050 */ NdrFcShort( 0xfc18 ), /* Offset= -1000 (50) */
/* 1052 */ 0x5c, /* FC_PAD */
/* 1054 */ 0xb4, /* FC_USER_MARSHAL */
/* 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 */
/* 1068 */ 0x5c, /* FC_PAD */
/* 1070 */ 0x11, 0x14, /* FC_RP
[allocated_on_stack] [pointer_deref] */
/* 1072 */ NdrFcShort( 0x2 ), /* Offset= 2 (1072) */
/* 1074 */ NdrFcShort( 0x2 ), /* Offset= 2 (1076) */
/* 1076 */ 0x1b, /* FC_CARRAY */
/* 1078 */ 0x0, /* 0 */
/* 1080 */ 0x28, /* Corr desc: parameter, FC_LONG */
/* 1082 */ NdrFcShort( 0x1 ), /* 1 */
/* 1084 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 1086 */ 0x2, /* FC_CHAR */
/* 1088 */ 0x5b, /* FC_END */
/* 1090 */ 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;
};

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,
    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 * pIndex )
{
    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 */
};
#if _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
#ifdef 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 startTime;
    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(paym_wrapper *pymt,void
*ctx);
extern "C" TPCCDB2GLUE_API int do_ords(ords_wrapper *ords,void *ctx);
extern "C" TPCCDB2GLUE_API int do_dlv(dlv_wrapper *dlvy,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 APIENTRY 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,(BYTE *)
&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);
        #endif
        fclose(respTimes);
        break;
    }
    return TRUE;
}
/*
*****
** Name          :          attachContext
** Description   :
**              :          Function calls db2 api
**              :          to attach thread to
**              :          a specific context per
** Parameters   :
**              :          void*
** 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)
    {
        rc: "<<rc<<endl);
        ERRORMSG("error detaching context from db,
        return ERR_DETACHING_CONTEXT;
    }
    return OK;
}
/*
*****
** Name          :          connect_db
** Description    :
**               Function calls db2 api
to connect to db
** Parameters    :
**               char*
dbName
**               void**
uninitialized context
** Returns       :
**               int - return code
** Comments      :
**               To connect to db, first
connection must be
**               established. Next,
context for that connect
**               be saved off. Finally,
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
to disconnect from db
** 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
to execute nord txn
** 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
#ifdef TIMING
    //call pymt txn
    payment_sql(&pymt->in_paym,&pymt->out_paym);
#endif
#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_ords
** Description    :
**               Function calls db2 api
to execute ords txn
** Parameters    :
**               ords_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_ords(ords_wrapper *ords, void *ctx)
{
    DEBUGMSG("Entered do_ords, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(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_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_dlvly
** Description    :
**               Function calls db2 api
to execute ords txn
** Parameters    :
**               dlvly_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_dlvly(dlvly_wrapper *dlvy, void *ctx)
{
    DEBUGMSG("Entered do_dlvly, 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 dlvly txn
    delivery_sql(&dlvy->in_dlvly, &dlvy->out_dlvly);
#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_dlvly.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" TPCADB2GLUE_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 "..\tpccIsapi\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;

```

<pre> 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; Supra Turbo"); strcpy(nord->out_nord.item[0].s_I_NAME, "98 Toyota 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'; Timer"); strcpy(nord->out_nord.item[1].s_I_NAME, "HKS Turbo 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'; Exhaust"); strcpy(nord->out_nord.item[2].s_I_NAME, "TRD GEN2 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'; DUAL-SOLENOID"); strcpy(nord->out_nord.item[3].s_I_NAME, "BLITZ 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 { 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; Supra NA"); strcpy(nord->out_nord.item[0].s_I_NAME, "97 Toyota 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'; </pre>	<pre> 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'; Exhaust Header"); strcpy(nord->out_nord.item[2].s_I_NAME, "XERD 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'; Conditioner"); strcpy(nord->out_nord.item[3].s_I_NAME, "LEXOL 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'; 1"); strcpy(nord->out_nord.item[4].s_I_NAME, "TRD Sticker 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'; 2"); strcpy(nord->out_nord.item[5].s_I_NAME, "TRD Sticker 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'; 3"); strcpy(nord->out_nord.item[6].s_I_NAME, "TRD Sticker 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'; 4"); strcpy(nord->out_nord.item[7].s_I_NAME, "TRD Sticker 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; </pre>
---	---

```

nord->out_nord.item[7].s_S_QUANTITY = 35;
nord->out_nord.item[7].s_brand_generic = 'G';

OEM Bra");
    strcpy(nord->out_nord.item[8].s_I_NAME,"98 Toyota

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 pymt_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;

Rd");
        strcpy(pymt->out_paym.s_W_STREET_1,"11501 Burnet

        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(pymt->out_paym.s_D_STREET_1,"11900 Hobby

Horse");
        strcpy(pymt->out_paym.s_D_STREET_2,"Apt. 525");
        strcpy(pymt->out_paym.s_D_CITY,"Valley");
        strcpy(pymt->out_paym.s_D_STATE,"TX");
        strcpy(pymt->out_paym.s_D_ZIP,"78559");
        strcpy(pymt->out_paym.s_C_FIRST,"Jim");
        strcpy(pymt->out_paym.s_C_MIDDLE,"F");
        strcpy(pymt->out_paym.s_C_LAST,"Truck");
        strcpy(pymt->out_paym.s_C_STREET_1,"100 N Solis");
        strcpy(pymt->out_paym.s_C_STREET_2,"Flat 343");
        strcpy(pymt->out_paym.s_C_CITY,"Cambridge");
        strcpy(pymt->out_paym.s_C_STATE,"NY");
        strcpy(pymt->out_paym.s_C_ZIP,"785585432");

strcpy(pymt->out_paym.s_C_PHONE,"1234567890123456");
    pymt->out_paym.s_C_SINCE_time = 0;
    strcpy(pymt->out_paym.s_C_CREDIT,"BC");

strcpy(pymt->out_paym.s_C_DATA,"XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X");
        dataSet = 1;
    }
    else
    {
        pymt->out_paym.s_C_CREDIT_LIM = 4000000;

```

```

pymt->out_paym.s_C_DISCOUNT = 52400;
pymt->out_paym.s_C_BALANCE = 14080;
pymt->out_paym.s_C_ID = 3180;
pymt->out_paym.s_H_DATE_time = 1234567890;

    strcpy(pymt->out_paym.s_W_STREET_1,"1201 Park
Ave.");

    strcpy(pymt->out_paym.s_W_STREET_2,"Suite 432");
    strcpy(pymt->out_paym.s_W_CITY,"Denver");
    strcpy(pymt->out_paym.s_W_STATE,"CO");
    strcpy(pymt->out_paym.s_W_ZIP,"787562356");
    strcpy(pymt->out_paym.s_D_STREET_1,"3404 Garth
Rd");

    strcpy(pymt->out_paym.s_D_STREET_2,"Suite 320");
    strcpy(pymt->out_paym.s_D_CITY,"Austin");
    strcpy(pymt->out_paym.s_D_STATE,"TX");
    strcpy(pymt->out_paym.s_D_ZIP,"785598767");
    strcpy(pymt->out_paym.s_C_FIRST,"John");
    strcpy(pymt->out_paym.s_C_MIDDLE,"P");
    strcpy(pymt->out_paym.s_C_LAST,"Williams");
    strcpy(pymt->out_paym.s_C_STREET_1,"North Rab
Road");

    strcpy(pymt->out_paym.s_C_STREET_2,"Apt 343");
    strcpy(pymt->out_paym.s_C_CITY,"La Fiera");
    strcpy(pymt->out_paym.s_C_STATE,"TX");
    strcpy(pymt->out_paym.s_C_ZIP,"785585432");

strcpy(pymt->out_paym.s_C_PHONE,"1234567890123456");
    pymt->out_paym.s_C_SINCE_time = 0;
    strcpy(pymt->out_paym.s_C_CREDIT,"GC");
    strcpy(pymt->out_paym.s_C_DATA,"Great Ebaye");

    dataSet = 0;
}
return OK;
}

extern "C" NULLDB_API int do_ordr(struct ordr_wrapper *ordr,void *ctx)
{
    ordr->out_ordr.s_transtatus = 0;

    if (dataSet == 0)
    {
        ordr->out_ordr.s_C_BALANCE = 100000;
        ordr->out_ordr.s_C_ID = 3;
        ordr->out_ordr.s_O_ID = 1696;
        ordr->out_ordr.s_O_CARRIER_ID = 9;
        ordr->out_ordr.s_ol_cnt = 6;
        ordr->out_ordr.s_O_ENTRY_D_time = 1234567890;

        strcpy(ordr->out_ordr.s_C_FIRST,"Homer");
        strcpy(ordr->out_ordr.s_C_MIDDLE,"J");
        strcpy(ordr->out_ordr.s_C_LAST,"Simpson");

        ordr->out_ordr.item[0].s_OL_AMOUNT = 30000;
        ordr->out_ordr.item[0].s_OL_I_ID = 23492;
        ordr->out_ordr.item[0].s_OL_SUPPLY_W_ID = 9;
        ordr->out_ordr.item[0].s_OL_QUANTITY = 5;
        ordr->out_ordr.item[0].s_OL_DELIVERY_D_time =
1234567890;

        ordr->out_ordr.item[1].s_OL_AMOUNT = 12300;
        ordr->out_ordr.item[1].s_OL_I_ID = 18860;
        ordr->out_ordr.item[1].s_OL_SUPPLY_W_ID = 9;
        ordr->out_ordr.item[1].s_OL_QUANTITY = 5;
        ordr->out_ordr.item[1].s_OL_DELIVERY_D_time =
1234567890;

```

```

ords->out_ords.item[2].s_OL_AMOUNT = 15000;
ords->out_ords.item[2].s_OL_I_ID = 90488;
ords->out_ords.item[2].s_OL_SUPPLY_W_ID = 9;
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 = 25000;
ords->out_ords.item[3].s_OL_I_ID = 22741;
ords->out_ords.item[3].s_OL_SUPPLY_W_ID = 9;
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 = 20000;
ords->out_ords.item[4].s_OL_I_ID = 92952;
ords->out_ords.item[4].s_OL_SUPPLY_W_ID = 9;
ords->out_ords.item[4].s_OL_QUANTITY = 5;
ords->out_ords.item[4].s_OL_DELIVERY_D_time =
1234567890;

ords->out_ords.item[5].s_OL_AMOUNT = 2345;
ords->out_ords.item[5].s_OL_I_ID = 29956;
ords->out_ords.item[5].s_OL_SUPPLY_W_ID = 9;
ords->out_ords.item[5].s_OL_QUANTITY = 5;
ords->out_ords.item[5].s_OL_DELIVERY_D_time =
1234567890;

dataSet = 1;
}
else
{
ords->out_ords.s_C_BALANCE = 123000;
ords->out_ords.s_C_ID = 856;
ords->out_ords.s_O_ID = 418;
ords->out_ords.s_O_CARRIER_ID = 10;
ords->out_ords.s_ol_cnt = 5;
strcpy(ords->out_ords.s_C_FIRST,"Erick");
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_dlv(struct dlv_wrapper *dlvy,void *ctx)
{
dlvy->out_dlv.s_transtatus = 0;

if (dataSet == 0)
{
dataSet = 1;

for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
dlvy->out_dlv.s_O_ID[districtIndex]= 2055;
}
else
{
for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
dlvy->out_dlv.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 paym_wrapper *pymt,void *ctx);
extern "C" NULLDB_API int do_ords(struct ords_wrapper *ords,void *ctx);
extern "C" NULLDB_API int do_dlvv(struct dlvy_wrapper *dlvy,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
// tpccIsapi.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 consituant 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 repersents 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 =
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;

                }
            }

        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;

}

        case '4':
            {
                // what follows?
                (*queryString)++;

                switch(**queryString)
                {
                    case '0':
                        queryChar = '@';
                        break;

                    case ' ':
                        queryChar = ' ';
                        break;

                }
            }

        case '5':
            {
                // what follows?
                (*queryString)++;

                switch(**queryString)
                {
                    case 'B':
                        queryChar = '[';
                        break;

                    case 'D':
                        queryChar = ']';
                        break;

                    case 'E':
                        queryChar = '^';
                        break;

                    case ' ':
                        queryChar = ' ';
                        break;

                }
            }

        case '7':
            {
                // what follows?
                (*queryString)++;

                switch(**queryString)

```

<pre> { 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 </pre>	<pre> #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];} </pre>
--	--

```

        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 tpccIsapi.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
// tpccIsapi.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 <atlsrvcs.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 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_DLTV "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 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 DAYS_IN_YEAR	365	
#define DEFAULT_DECIMAL32_LEN	5		#define YEARS_IN_LEAP	4	
#define DEFAULT_DECIMAL16_LEN	5		#define START_YEAR		1970
#define DEFAULT_DATETIME_LEN	19		#define MONTHS_IN_YEAR	12	
#define DEFAULT_DATE_LEN	11		//		
#define DEFAULT_TIME_LEN	8		// Error codes		
#define DEFAULT_STRING_LEN	25		//		
#define DEFAULT_ZIP_LEN	17		#define ERR_INVALID_TXN_TYPE		-1
#define DEFAULT_PHONE_LEN	18				
//			#define ERR_MISSING_W_ID		-2
// String Field Lengths			#define ERR_NON_NUMERIC_W_ID		-3
//			#define ERR_MISSING_D_ID		-4
			#define ERR_NON_NUMERIC_D_ID		-5
#define NAME_LEN	24		#define ERR_MISSING_C_ID		-6
#define LAST_NAME_LEN	16		#define ERR_NON_NUMERIC_C_ID		-7
#define FIRST_NAME_LEN	16				
#define INITIALS_LEN	2		#define ERR_MISSING_SUPP_W		-8
#define CREDIT_LEN	2		#define ERR_NON_NUMERIC_SUPP_W		-9
			#define ERR_MISSING_ITEM_NUM		-10
#define STREET_LEN	20		#define ERR_NON_NUMERIC_ITEM_NUM		-11
#define CITY_LEN	20		#define ERR_MISSING_ITEM_OTY		-12
#define STATE_LEN	2		#define ERR_NON_NUMERIC_ITEM_QTY		-13
#define ZIP_LEN	9				
			#define ERR_MISSING_CLAST_NAME		-14
#define PHONE_LEN	16		#define ERR_NON_NUMERIC_CUST_W_ID		-15
#define DATA_LEN	200		#define ERR_NON_NUMERIC_CUST_D_ID		-16
			#define ERR_MISSING_AMOUNT_PAID		-17
#define ITEM_LIST	15		#define ERR_NON_NUMERIC_AMOUNT_PAID		-18
#define ORDER_LIST	10				
//			#define ERR_INVALID_D_ID		"ERROR:
// Type definitions			Invalid District ID. Try Again."		
//			#define ERR_INVALID_W_ID		"ERROR:
			Invalid Warehouse ID. Try Again."		
typedef __int8	INT8b;		#define ERR_INVALID_C_ID		"ERROR: Invalid
typedef __int16	INT16b;		Customer ID. Try Again."		
typedef __int32	INT32b;		#define ERR_INVALID_SUPPLY_W_ID		"ERROR: Invalid Item
typedef __int64	INT64b;		Supply Warehouse. Try Again."		
			#define ERR_INVALID_ITEM_NUM		"ERROR: Invalid Item
typedef unsigned __int8	UINT8b;		Number. Try Again."		
typedef unsigned __int16	UINT16b;		#define ERR_INVALID_ITEM_OTY		"ERROR: Invalid Item
typedef unsigned __int32	UINT32b;		Qty. Try Again."		
typedef unsigned __int64	UINT64b;		#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."		
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		-51
#define SECONDS_IN_HOUR	3600				
#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 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
// 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
// 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_dlv;
    struct out_delivery_struct out_dlv;
};

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
//           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
// page
// Parameters :
//           **htmlPage
//           amount of characters
// Returns   :
//           the function appended
//           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
// page
// 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)++ = ' ';
        }
    }
    **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
reset month
            timestamp->year += 1;
            timestamp->month = 1;

            // are we now on a leap
year
            leap =
!(timestamp->year % YEARS_IN_LEAP);
        }
        else
        {
            // set day of month to remaioning
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 >
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
                        indicator
                        if(negativeFlag)

```

```

        {
            negativeFlag
= false;
buffer[--index] = NEGATIVE_SYMBOL;
        }
        else
        {
            // add the
money indicator
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
                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 buffer[--index] =
NUMERIC_FILL;
                    }
                }
                // need to trace place for decimal point and
                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__)
errorStream \
\
errorStream.flush();

LeaveCriticalSection(&errorMutex);

#ifdef _DEBUG

#define DEBUGMSG(TEXT)

\
EnterCriticalSection(&debugMutex);
\

\
debugStream << debugFileName(__FILE__)
debugStream \
\
debugStream.flush();

LeaveCriticalSection(&debugMutex);

```

```

errorStream
\
<< "|" <<
\
<< _getpid()
\
<< TEXT;

\
errorStream \
\
errorStream.flush();

\
debugStream << debugFileName(__FILE__)
debugStream \
\
debugStream.flush();

```

```

#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 DLVYQUEUEUEDATA *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
** Parameters     :
**               value.
** 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\\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 = (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,(BYTE
    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
** 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>"

    "<FORM
    ACTION=\"\"
    APP_NAME
    \"\"
    METHOD=\"GET\">"
    "<H2>Please
    Login.</H2>"
    "<INPUT
    TYPE=\"hidden\" NAME=\"\"
    CMD_TXN_ID
    \"\" VALUE=\"\"
    CMD_MENU
    \"\">"
    "<H3>Warehouse
    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
point
** Parameters   :
**              htmlPhraser*    command
**              block
**              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"
    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>"
    "<FORM
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=""
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
<INPUT TYPE=""submit"" NAME=""
CMD_TXN_ID
"" VALUE=""
CMD_NORD
"">");

    //append the hidden
    html+=appendHiddenFields(html,txnHandle);

    //int buffer for warehouse

```

```

char buffer[15];
appendText(&html," <PRE>"
//
4 5 6 7 8 9\r\n"
//
"12345678901234567890123456789012345678901234567890123456789012345678901234567890\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"
//
"12345678901234567890123456789012345678901234567890123456789012345678901234567890\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"
                "
//
"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,ittoa(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,ittoa(nord->out_nord.s_O_ID,buffer,10),10,1);

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

if(rc != INVALID_STATUS)

appendText(&html,ittoa(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"
//
"123456789012345678901234567890123456789012345678901234567890123
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,ittoa(nord->in_nord.in_item[itemCount].s_OL_SUPPLY_W
_ID,buffer,10),8,1);

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

        appendText(&html,ittoa(nord->out_nord.item[itemCount].s_S_QUANTITY,bu
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>");

```

```

        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
    **               char *           html result
    **               char *           error
    **               htmlPhraser*    command block
    **               TXN_HANDLE*     txn handle
    ** 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=\\"
METHOD=\\"GET\>\r\n"
" <CENTER><H3>Please Fill In New Order Form.</H3></CENTER>\r\n"
<INPUT TYPE=\\"submit\\" NAME=\\"
"Submit Transaction
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>
6 7 8 9\r\n"
1 2 3 4 5
9\r\n"
*/

```

```

"1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890\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"
/* 1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890\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    :
**               htmlPhraser*    command
**               TXN_HANDLE*     txn handle
** 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=\\"
METHOD=\\"GET\>\r\n"
" <CENTER><H3>Please Fill In Payment Form.</H3></CENTER> <BR>\r\n"
"Submit Transaction
"Submit Transaction
CMD_TXN_ID
\>\");

    //append the hidden warehouse and district fields
    html+=appendHiddenFields(html,txnHandle);

    //int buffer for warehouse
    char buffer[15];
    /*appendText(&html, "<PRE>
6 7 8 9\r\n"
1 2 3 4 5
9\r\n"
*/

```

```

        "\ 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
        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
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,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= \"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 = (paym_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    :
** 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></HTML>");
    " <BODY><FORM
ACTION=\\\"
APP_NAME
\\\"
METHOD=\\\"GET\\\"></HTML>";
" <CENTER><H3>Please Fill In Payment Form.</H3></CENTER> <BR></HTML>";
"Submit Transaction
<INPUT TYPE=\\\"submit\\\" NAME=\\\"
CMD_TXN_ID

```

```

"\" VALUE=\\\"
CMD_PYMT
\\\">");
html+=appendHiddenFields(html,txnHandle);
appendText(&html,"<BR><PRE></HTML>";
"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></HTML><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>";
" ";
" ";
Credit: <BR>";
" ";
" ";
%Disc: <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    :
** Parameters    :
**               HTML orderStatus page entry
point
** Returns       :
**               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"
                                "<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
excepcion.</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,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,"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,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,"<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></HTML>");

```

```

ACTION=""

APP_NAME
""

METHOD="GET"></HTML>

"<CENTER><H3>Please Fill In Order Status Form.</H3></CENTER>
<BR></HTML>

"Submit Transaction

"CMD_TXN_ID
"VALUE=""
"CMD_ORDS
">
" <BR> ";

html+=appendHiddenFields(html,txnHandle);

appendText(&html,"<PRE></HTML>");

"Warehouse: ";

char buffer[15];
appendText(&html,itoa(txnHandle->w_id,buffer,10));

appendText(&html," District: <INPUT NAME=""
CMD_D_ID
"SIZE=1></HTML><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 Oty 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"
ACTION=""
METHOD="GET"
"<CENTER><H3>Delivery.</H3></CENTER>\r\n"
<INPUT TYPE="submit" NAME=""
"Submit Transaction
CMD_TXN_ID
" VALUE=""
CMD_DLVE
">";
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>");
html+=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"");
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,ittoa(dlvy.in_dlvy.s_W_ID,buffer,10));
appendText(&html,"<BR> <BR>Carrier Number: ");
//get carrier_id from wrapper
appendText(&html,ittoa(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="\
METHOD="\GET">\r\n"
"<CENTER><H3>Delivery.</H3></CENTER>\r\n"
<INPUT TYPE="\submit" NAME="\
CMD_TXN_ID
"\ VALUE="\
CMD_DLVY
"\>");
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"<BR> <PRE>"
"Warehouse: ");
    char buffer[15];
    appendText(&html,ittoa(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"
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),6+1,1);
    appendText(&html,"District: ");
    appendText(&html,ittoa(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
** 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,txnHan
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,itoa(stok->in_stok.s_W_ID,buffer,10),6+1,1);

        appendText(&html,"District: ");
        appendText(&html,itoa(stok->in_stok.s_D_ID,buffer,10));

        appendText(&html," <BR> <BR>"
                "Stock Level
Threshold: ");
        appendText(&html,itoa(stok->in_stok.s_threshold,buffer,10));

        appendText(&html," <BR> <BR>"
                "Low Stock: ");
        appendText(&html,itoa(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"
                "<BODY><FORM
ACTION= \"\"
                APP_NAME
                \"\"
METHOD= \"GET\">\r\n"
                "Submit Transaction
                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= \"\"
                \"\" SIZE=1> <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
**              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"
                "<INPUT
                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>
        <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    :
**              All dlvy threads created by
initDlvy 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
        {
            DEBUGMSG("dlvyThread initializing wait
handles" << endl);

            //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,"\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
exception in dlvy thread. Thread exiting"<<endl);
            fprintf(dlvyLog,"ERROR: Unhandled
exception 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 terminal 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\nW_ID: %d CARRIER_ID: %d
%s\nend-time: %d.%03d\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

```

TpccIsapi.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        "tpccIsapi"
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_001 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
) EXTENTSIZE 32 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace WAR_002 PAGESIZE 4096 managed by database using (

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
) EXTENTSIZE 32 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace WAR_003 PAGESIZE 4096 managed by database using (

DEVICE 'C:\Containers\WAR\011' 256,
DEVICE 'C:\Containers\WAR\012' 256,
DEVICE 'C:\Containers\WAR\013' 256,
DEVICE 'C:\Containers\WAR\014' 256,
DEVICE 'C:\Containers\WAR\015' 256
) EXTENTSIZE 32 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace WAR_004 PAGESIZE 4096 managed by database using (

DEVICE 'C:\Containers\WAR\016' 256,
DEVICE 'C:\Containers\WAR\017' 256,
DEVICE 'C:\Containers\WAR\018' 256,
DEVICE 'C:\Containers\WAR\019' 256,
DEVICE 'C:\Containers\WAR\020' 256
) EXTENTSIZE 32 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace WAR_005 PAGESIZE 4096 managed by database using (

DEVICE 'C:\Containers\WAR\021' 256,
DEVICE 'C:\Containers\WAR\022' 256,
DEVICE 'C:\Containers\WAR\023' 256,
DEVICE 'C:\Containers\WAR\024' 256,
DEVICE 'C:\Containers\WAR\025' 256
) EXTENTSIZE 32 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace WAR_006 PAGESIZE 4096 managed by database using (

DEVICE 'C:\Containers\WAR\026' 256,
DEVICE 'C:\Containers\WAR\027' 256,
DEVICE 'C:\Containers\WAR\028' 256,
DEVICE 'C:\Containers\WAR\029' 256,
DEVICE 'C:\Containers\WAR\030' 256
) EXTENTSIZE 32 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace WAR_007 PAGESIZE 4096 managed by database using (

DEVICE 'C:\Containers\WAR\031' 256,

DEVICE 'C:\Containers\WAR\032' 256,
DEVICE 'C:\Containers\WAR\033' 256,
DEVICE 'C:\Containers\WAR\034' 256,
DEVICE 'C:\Containers\WAR\035' 256
) EXTENTSIZE 32 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace WAR_008 PAGESIZE 4096 managed by database using (

DEVICE 'C:\Containers\WAR\036' 256,
DEVICE 'C:\Containers\WAR\037' 256,
DEVICE 'C:\Containers\WAR\038' 256,
DEVICE 'C:\Containers\WAR\039' 256,
DEVICE 'C:\Containers\WAR\040' 256
) EXTENTSIZE 32 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

-- DIS

create regular tablespace DIS_001 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
) EXTENTSIZE 64 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace DIS_002 PAGESIZE 4096 managed by database using (

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
) EXTENTSIZE 64 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace DIS_003 PAGESIZE 4096 managed by database using (

DEVICE 'C:\Containers\DIS\011' 512,
DEVICE 'C:\Containers\DIS\012' 512,
DEVICE 'C:\Containers\DIS\013' 512,
DEVICE 'C:\Containers\DIS\014' 512,
DEVICE 'C:\Containers\DIS\015' 512
) EXTENTSIZE 64 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace DIS_004 PAGESIZE 4096 managed by database using (

DEVICE 'C:\Containers\DIS\016' 512,
DEVICE 'C:\Containers\DIS\017' 512,
DEVICE 'C:\Containers\DIS\018' 512,
DEVICE 'C:\Containers\DIS\019' 512,
DEVICE 'C:\Containers\DIS\020' 512
) EXTENTSIZE 64 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace DIS_005 PAGESIZE 4096 managed by database using (

DEVICE 'C:\Containers\DIS\021' 512,
DEVICE 'C:\Containers\DIS\022' 512,
DEVICE 'C:\Containers\DIS\023' 512,
DEVICE 'C:\Containers\DIS\024' 512,
DEVICE 'C:\Containers\DIS\025' 512
) EXTENTSIZE 64 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace DIS_006 PAGESIZE 4096 managed by database using (

DEVICE 'C:\Containers\DIS\026' 512,
DEVICE 'C:\Containers\DIS\027' 512,
DEVICE 'C:\Containers\DIS\028' 512,
DEVICE 'C:\Containers\DIS\029' 512,
DEVICE 'C:\Containers\DIS\030' 512

```

) EXTENTSIZE 64 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;
create regular tablespace DIS_007 PAGESIZE 4096 managed by database using
(
DEVICE 'C:\Containers\DIS\031' 512,
DEVICE 'C:\Containers\DIS\032' 512,
DEVICE 'C:\Containers\DIS\033' 512,
DEVICE 'C:\Containers\DIS\034' 512,
DEVICE 'C:\Containers\DIS\035' 512
) EXTENTSIZE 64 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace DIS_008 PAGESIZE 4096 managed by database using
(
DEVICE 'C:\Containers\DIS\036' 512,
DEVICE 'C:\Containers\DIS\037' 512,
DEVICE 'C:\Containers\DIS\038' 512,
DEVICE 'C:\Containers\DIS\039' 512,
DEVICE 'C:\Containers\DIS\040' 512
) EXTENTSIZE 64 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

-- ITM
create regular tablespace ITM PAGESIZE 8192 managed by database using (
DEVICE 'C:\Containers\ITM\001' 128,
DEVICE 'C:\Containers\ITM\002' 128,
DEVICE 'C:\Containers\ITM\003' 128,
DEVICE 'C:\Containers\ITM\004' 128,
DEVICE 'C:\Containers\ITM\005' 128,
DEVICE 'C:\Containers\ITM\006' 128,
DEVICE 'C:\Containers\ITM\007' 128,
DEVICE 'C:\Containers\ITM\008' 128,
DEVICE 'C:\Containers\ITM\009' 128,
DEVICE 'C:\Containers\ITM\010' 128,
DEVICE 'C:\Containers\ITM\011' 128,
DEVICE 'C:\Containers\ITM\012' 128,
DEVICE 'C:\Containers\ITM\013' 128,
DEVICE 'C:\Containers\ITM\014' 128,
DEVICE 'C:\Containers\ITM\015' 128,
DEVICE 'C:\Containers\ITM\016' 128,
DEVICE 'C:\Containers\ITM\017' 128,
DEVICE 'C:\Containers\ITM\018' 128,
DEVICE 'C:\Containers\ITM\019' 128,
DEVICE 'C:\Containers\ITM\020' 128,
DEVICE 'C:\Containers\ITM\021' 128,
DEVICE 'C:\Containers\ITM\022' 128,
DEVICE 'C:\Containers\ITM\023' 128,
DEVICE 'C:\Containers\ITM\024' 128,
DEVICE 'C:\Containers\ITM\025' 128,
DEVICE 'C:\Containers\ITM\026' 128,
DEVICE 'C:\Containers\ITM\027' 128,
DEVICE 'C:\Containers\ITM\028' 128,
DEVICE 'C:\Containers\ITM\029' 128,
DEVICE 'C:\Containers\ITM\030' 128,
DEVICE 'C:\Containers\ITM\031' 128,
DEVICE 'C:\Containers\ITM\032' 128,
DEVICE 'C:\Containers\ITM\033' 128,
DEVICE 'C:\Containers\ITM\034' 128,
DEVICE 'C:\Containers\ITM\035' 128,
DEVICE 'C:\Containers\ITM\036' 128,
DEVICE 'C:\Containers\ITM\037' 128,
DEVICE 'C:\Containers\ITM\038' 128,
DEVICE 'C:\Containers\ITM\039' 128,
DEVICE 'C:\Containers\ITM\040' 128
) EXTENTSIZE 16 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

-- STK
create regular tablespace STK_001 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\001' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace STK_002 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\002' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace STK_003 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\003' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace STK_004 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\004' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace STK_005 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\005' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace STK_006 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\006' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace STK_007 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\007' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace STK_008 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\008' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace STK_009 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\009' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace STK_010 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\010' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace STK_011 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\011' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace STK_012 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\012' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace STK_013 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\013' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace STK_014 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\STK\014' 4761600
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

```



```

create regular tablespace ORD_002 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORD\006' 99840,
DEVICE 'C:\Containers\ORD\007' 99840,
DEVICE 'C:\Containers\ORD\008' 99840,
DEVICE 'C:\Containers\ORD\009' 99840,
DEVICE 'C:\Containers\ORD\010' 99840
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace ORD_003 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORD\011' 99840,
DEVICE 'C:\Containers\ORD\012' 99840,
DEVICE 'C:\Containers\ORD\013' 99840,
DEVICE 'C:\Containers\ORD\014' 99840,
DEVICE 'C:\Containers\ORD\015' 99840
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace ORD_004 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORD\016' 99840,
DEVICE 'C:\Containers\ORD\017' 99840,
DEVICE 'C:\Containers\ORD\018' 99840,
DEVICE 'C:\Containers\ORD\019' 99840,
DEVICE 'C:\Containers\ORD\020' 99840
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace ORD_005 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORD\021' 99840,
DEVICE 'C:\Containers\ORD\022' 99840,
DEVICE 'C:\Containers\ORD\023' 99840,
DEVICE 'C:\Containers\ORD\024' 99840,
DEVICE 'C:\Containers\ORD\025' 99840
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace ORD_006 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORD\026' 99840,
DEVICE 'C:\Containers\ORD\027' 99840,
DEVICE 'C:\Containers\ORD\028' 99840,
DEVICE 'C:\Containers\ORD\029' 99840,
DEVICE 'C:\Containers\ORD\030' 99840
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace ORD_007 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORD\031' 99840,
DEVICE 'C:\Containers\ORD\032' 99840,
DEVICE 'C:\Containers\ORD\033' 99840,
DEVICE 'C:\Containers\ORD\034' 99840,
DEVICE 'C:\Containers\ORD\035' 99840
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace ORD_008 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORD\036' 99840,
DEVICE 'C:\Containers\ORD\037' 99840,
DEVICE 'C:\Containers\ORD\038' 99840,
DEVICE 'C:\Containers\ORD\039' 99840,
DEVICE 'C:\Containers\ORD\040' 99840
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

-- ORDI
create regular tablespace ORDI_001 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORDI\001' 97280,
DEVICE 'C:\Containers\ORDI\002' 97280,
DEVICE 'C:\Containers\ORDI\003' 97280,
DEVICE 'C:\Containers\ORDI\004' 97280,
DEVICE 'C:\Containers\ORDI\005' 97280
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace ORDI_002 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORDI\006' 97280,
DEVICE 'C:\Containers\ORDI\007' 97280,
DEVICE 'C:\Containers\ORDI\008' 97280,
DEVICE 'C:\Containers\ORDI\009' 97280,
DEVICE 'C:\Containers\ORDI\010' 97280
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace ORDI_003 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORDI\011' 97280,
DEVICE 'C:\Containers\ORDI\012' 97280,
DEVICE 'C:\Containers\ORDI\013' 97280,
DEVICE 'C:\Containers\ORDI\014' 97280,
DEVICE 'C:\Containers\ORDI\015' 97280
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace ORDI_004 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORDI\016' 97280,
DEVICE 'C:\Containers\ORDI\017' 97280,
DEVICE 'C:\Containers\ORDI\018' 97280,
DEVICE 'C:\Containers\ORDI\019' 97280,
DEVICE 'C:\Containers\ORDI\020' 97280
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace ORDI_005 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORDI\021' 97280,
DEVICE 'C:\Containers\ORDI\022' 97280,
DEVICE 'C:\Containers\ORDI\023' 97280,
DEVICE 'C:\Containers\ORDI\024' 97280,
DEVICE 'C:\Containers\ORDI\025' 97280
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace ORDI_006 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORDI\026' 97280,
DEVICE 'C:\Containers\ORDI\027' 97280,
DEVICE 'C:\Containers\ORDI\028' 97280,
DEVICE 'C:\Containers\ORDI\029' 97280,
DEVICE 'C:\Containers\ORDI\030' 97280
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace ORDI_007 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORDI\031' 97280,
DEVICE 'C:\Containers\ORDI\032' 97280,
DEVICE 'C:\Containers\ORDI\033' 97280,
DEVICE 'C:\Containers\ORDI\034' 97280,
DEVICE 'C:\Containers\ORDI\035' 97280
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

create regular tablespace ORDI_008 PAGESIZE 8192 managed by database
using (
DEVICE 'C:\Containers\ORDI\036' 97280,
DEVICE 'C:\Containers\ORDI\037' 97280,
DEVICE 'C:\Containers\ORDI\038' 97280,
DEVICE 'C:\Containers\ORDI\039' 97280,
DEVICE 'C:\Containers\ORDI\040' 97280
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT8K;

```



```

create regular tablespace NEWA_008 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWA\036' 51200,
DEVICE 'C:\Containers\NEWA\037' 51200,
DEVICE 'C:\Containers\NEWA\038' 51200,
DEVICE 'C:\Containers\NEWA\039' 51200,
DEVICE 'C:\Containers\NEWA\040' 51200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

-- HST
create regular tablespace HST_001 PAGESIZE 16384 managed by database
using (
DEVICE 'C:\Containers\HST\001' 83200,
DEVICE 'C:\Containers\HST\002' 83200,
DEVICE 'C:\Containers\HST\003' 83200,
DEVICE 'C:\Containers\HST\004' 83200,
DEVICE 'C:\Containers\HST\005' 83200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT16K;

create regular tablespace HST_002 PAGESIZE 16384 managed by database
using (
DEVICE 'C:\Containers\HST\006' 83200,
DEVICE 'C:\Containers\HST\007' 83200,
DEVICE 'C:\Containers\HST\008' 83200,
DEVICE 'C:\Containers\HST\009' 83200,
DEVICE 'C:\Containers\HST\010' 83200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT16K;

create regular tablespace HST_003 PAGESIZE 16384 managed by database
using (
DEVICE 'C:\Containers\HST\011' 83200,
DEVICE 'C:\Containers\HST\012' 83200,
DEVICE 'C:\Containers\HST\013' 83200,
DEVICE 'C:\Containers\HST\014' 83200,
DEVICE 'C:\Containers\HST\015' 83200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT16K;

create regular tablespace HST_004 PAGESIZE 16384 managed by database
using (
DEVICE 'C:\Containers\HST\016' 83200,
DEVICE 'C:\Containers\HST\017' 83200,
DEVICE 'C:\Containers\HST\018' 83200,
DEVICE 'C:\Containers\HST\019' 83200,
DEVICE 'C:\Containers\HST\020' 83200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT16K;

create regular tablespace HST_005 PAGESIZE 16384 managed by database
using (
DEVICE 'C:\Containers\HST\021' 83200,
DEVICE 'C:\Containers\HST\022' 83200,
DEVICE 'C:\Containers\HST\023' 83200,
DEVICE 'C:\Containers\HST\024' 83200,
DEVICE 'C:\Containers\HST\025' 83200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT16K;

create regular tablespace HST_006 PAGESIZE 16384 managed by database
using (
DEVICE 'C:\Containers\HST\026' 83200,
DEVICE 'C:\Containers\HST\027' 83200,
DEVICE 'C:\Containers\HST\028' 83200,
DEVICE 'C:\Containers\HST\029' 83200,
DEVICE 'C:\Containers\HST\030' 83200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT16K;

create regular tablespace HST_007 PAGESIZE 16384 managed by database
using (
DEVICE 'C:\Containers\HST\031' 83200,
DEVICE 'C:\Containers\HST\032' 83200,
DEVICE 'C:\Containers\HST\033' 83200,
DEVICE 'C:\Containers\HST\034' 83200,
DEVICE 'C:\Containers\HST\035' 83200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT16K;

create regular tablespace HST_008 PAGESIZE 16384 managed by database
using (
DEVICE 'C:\Containers\HST\036' 83200,
DEVICE 'C:\Containers\HST\037' 83200,
DEVICE 'C:\Containers\HST\038' 83200,
DEVICE 'C:\Containers\HST\039' 83200,
DEVICE 'C:\Containers\HST\040' 83200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULT16K;

-- NEWB
create regular tablespace NEWB_001 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\001' 51200,
DEVICE 'C:\Containers\NEWB\002' 51200,
DEVICE 'C:\Containers\NEWB\003' 51200,
DEVICE 'C:\Containers\NEWB\004' 51200,
DEVICE 'C:\Containers\NEWB\005' 51200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWB_002 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\006' 51200,
DEVICE 'C:\Containers\NEWB\007' 51200,
DEVICE 'C:\Containers\NEWB\008' 51200,
DEVICE 'C:\Containers\NEWB\009' 51200,
DEVICE 'C:\Containers\NEWB\010' 51200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWB_003 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\011' 51200,
DEVICE 'C:\Containers\NEWB\012' 51200,
DEVICE 'C:\Containers\NEWB\013' 51200,
DEVICE 'C:\Containers\NEWB\014' 51200,
DEVICE 'C:\Containers\NEWB\015' 51200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWB_004 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\016' 51200,
DEVICE 'C:\Containers\NEWB\017' 51200,
DEVICE 'C:\Containers\NEWB\018' 51200,
DEVICE 'C:\Containers\NEWB\019' 51200,
DEVICE 'C:\Containers\NEWB\020' 51200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWB_005 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\021' 51200,
DEVICE 'C:\Containers\NEWB\022' 51200,
DEVICE 'C:\Containers\NEWB\023' 51200,
DEVICE 'C:\Containers\NEWB\024' 51200,
DEVICE 'C:\Containers\NEWB\025' 51200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWB_006 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\026' 51200,
DEVICE 'C:\Containers\NEWB\027' 51200,
DEVICE 'C:\Containers\NEWB\028' 51200,
DEVICE 'C:\Containers\NEWB\029' 51200,
DEVICE 'C:\Containers\NEWB\030' 51200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

```

```

) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWB_007 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\031' 51200,
DEVICE 'C:\Containers\NEWB\032' 51200,
DEVICE 'C:\Containers\NEWB\033' 51200,
DEVICE 'C:\Containers\NEWB\034' 51200,
DEVICE 'C:\Containers\NEWB\035' 51200
) EXTENTSIZE 256 PREFETCHSIZE 0 BUFFERPOOL IBMDEFAULTBP;

create regular tablespace NEWB_008 PAGESIZE 4096 managed by database
using (
DEVICE 'C:\Containers\NEWB\036' 51200,
DEVICE 'C:\Containers\NEWB\037' 51200,
DEVICE 'C:\Containers\NEWB\038' 51200,
DEVICE 'C:\Containers\NEWB\039' 51200,
DEVICE 'C:\Containers\NEWB\040' 51200
) 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 tpcc;

alter tablespace WAR_001 bufferpool WDS1;
alter tablespace WAR_002 bufferpool WDS2;
alter tablespace WAR_003 bufferpool WDS3;
alter tablespace WAR_004 bufferpool WDS4;
alter tablespace WAR_005 bufferpool WDS5;
alter tablespace WAR_006 bufferpool WDS6;
alter tablespace WAR_007 bufferpool WDS7;
alter tablespace WAR_008 bufferpool WDS8;

alter tablespace DIS_001 bufferpool WDS1;

```

```

alter tablespace DIS_002 bufferpool WDS2;
alter tablespace DIS_003 bufferpool WDS3;
alter tablespace DIS_004 bufferpool WDS4;
alter tablespace DIS_005 bufferpool WDS5;
alter tablespace DIS_006 bufferpool WDS6;
alter tablespace DIS_007 bufferpool WDS7;
alter tablespace DIS_008 bufferpool WDS8;

alter tablespace ITM bufferpool ITM;

alter tablespace STK_001 bufferpool STK1;
alter tablespace STK_002 bufferpool STK1;
alter tablespace STK_003 bufferpool STK1;
alter tablespace STK_004 bufferpool STK1;
alter tablespace STK_005 bufferpool STK1;
alter tablespace STK_006 bufferpool STK2;
alter tablespace STK_007 bufferpool STK2;
alter tablespace STK_008 bufferpool STK2;
alter tablespace STK_009 bufferpool STK2;
alter tablespace STK_010 bufferpool STK2;
alter tablespace STK_011 bufferpool STK3;
alter tablespace STK_012 bufferpool STK3;
alter tablespace STK_013 bufferpool STK3;
alter tablespace STK_014 bufferpool STK3;
alter tablespace STK_015 bufferpool STK3;
alter tablespace STK_016 bufferpool STK4;
alter tablespace STK_017 bufferpool STK4;
alter tablespace STK_018 bufferpool STK4;
alter tablespace STK_019 bufferpool STK4;
alter tablespace STK_020 bufferpool STK4;
alter tablespace STK_021 bufferpool STK5;
alter tablespace STK_022 bufferpool STK5;
alter tablespace STK_023 bufferpool STK5;
alter tablespace STK_024 bufferpool STK5;
alter tablespace STK_025 bufferpool STK5;
alter tablespace STK_026 bufferpool STK6;
alter tablespace STK_027 bufferpool STK6;
alter tablespace STK_028 bufferpool STK6;
alter tablespace STK_029 bufferpool STK6;

```


alter tablespace STK_030 bufferpool STK6;
alter tablespace STK_031 bufferpool STK7;
alter tablespace STK_032 bufferpool STK7;
alter tablespace STK_033 bufferpool STK7;
alter tablespace STK_034 bufferpool STK7;
alter tablespace STK_035 bufferpool STK7;
alter tablespace STK_036 bufferpool STK8;
alter tablespace STK_037 bufferpool STK8;
alter tablespace STK_038 bufferpool STK8;
alter tablespace STK_039 bufferpool STK8;
alter tablespace STK_040 bufferpool STK8;

alter tablespace CST_001 bufferpool CST1;
alter tablespace CST_002 bufferpool CST1;
alter tablespace CST_003 bufferpool CST1;
alter tablespace CST_004 bufferpool CST1;
alter tablespace CST_005 bufferpool CST1;
alter tablespace CST_006 bufferpool CST2;
alter tablespace CST_007 bufferpool CST2;
alter tablespace CST_008 bufferpool CST2;
alter tablespace CST_009 bufferpool CST2;
alter tablespace CST_010 bufferpool CST2;
alter tablespace CST_011 bufferpool CST3;
alter tablespace CST_012 bufferpool CST3;
alter tablespace CST_013 bufferpool CST3;
alter tablespace CST_014 bufferpool CST3;
alter tablespace CST_015 bufferpool CST3;
alter tablespace CST_016 bufferpool CST4;
alter tablespace CST_017 bufferpool CST4;
alter tablespace CST_018 bufferpool CST4;
alter tablespace CST_019 bufferpool CST4;
alter tablespace CST_020 bufferpool CST4;
alter tablespace CST_021 bufferpool CST5;
alter tablespace CST_022 bufferpool CST5;
alter tablespace CST_023 bufferpool CST5;
alter tablespace CST_024 bufferpool CST5;
alter tablespace CST_025 bufferpool CST5;
alter tablespace CST_026 bufferpool CST6;
alter tablespace CST_027 bufferpool CST6;

alter tablespace CST_028 bufferpool CST6;
alter tablespace CST_029 bufferpool CST6;
alter tablespace CST_030 bufferpool CST6;
alter tablespace CST_031 bufferpool CST7;
alter tablespace CST_032 bufferpool CST7;
alter tablespace CST_033 bufferpool CST7;
alter tablespace CST_034 bufferpool CST7;
alter tablespace CST_035 bufferpool CST7;
alter tablespace CST_036 bufferpool CST8;
alter tablespace CST_037 bufferpool CST8;
alter tablespace CST_038 bufferpool CST8;
alter tablespace CST_039 bufferpool CST8;
alter tablespace CST_040 bufferpool CST8;

alter tablespace CSTI_001 bufferpool CSTI1;
alter tablespace CSTI_002 bufferpool CSTI1;
alter tablespace CSTI_003 bufferpool CSTI1;
alter tablespace CSTI_004 bufferpool CSTI1;
alter tablespace CSTI_005 bufferpool CSTI1;
alter tablespace CSTI_006 bufferpool CSTI2;
alter tablespace CSTI_007 bufferpool CSTI2;
alter tablespace CSTI_008 bufferpool CSTI2;
alter tablespace CSTI_009 bufferpool CSTI2;
alter tablespace CSTI_010 bufferpool CSTI2;
alter tablespace CSTI_011 bufferpool CSTI3;
alter tablespace CSTI_012 bufferpool CSTI3;
alter tablespace CSTI_013 bufferpool CSTI3;
alter tablespace CSTI_014 bufferpool CSTI3;
alter tablespace CSTI_015 bufferpool CSTI3;
alter tablespace CSTI_016 bufferpool CSTI4;
alter tablespace CSTI_017 bufferpool CSTI4;
alter tablespace CSTI_018 bufferpool CSTI4;
alter tablespace CSTI_019 bufferpool CSTI4;
alter tablespace CSTI_020 bufferpool CSTI4;
alter tablespace CSTI_021 bufferpool CSTI5;
alter tablespace CSTI_022 bufferpool CSTI5;
alter tablespace CSTI_023 bufferpool CSTI5;
alter tablespace CSTI_024 bufferpool CSTI5;

```

alter tablespace CSTI_025 bufferpool CSTI5;
alter tablespace CSTI_026 bufferpool CSTI6;
alter tablespace CSTI_027 bufferpool CSTI6;
alter tablespace CSTI_028 bufferpool CSTI6;
alter tablespace CSTI_029 bufferpool CSTI6;
alter tablespace CSTI_030 bufferpool CSTI6;
alter tablespace CSTI_031 bufferpool CSTI7;
alter tablespace CSTI_032 bufferpool CSTI7;
alter tablespace CSTI_033 bufferpool CSTI7;
alter tablespace CSTI_034 bufferpool CSTI7;
alter tablespace CSTI_035 bufferpool CSTI7;
alter tablespace CSTI_036 bufferpool CSTI8;
alter tablespace CSTI_037 bufferpool CSTI8;
alter tablespace CSTI_038 bufferpool CSTI8;
alter tablespace CSTI_039 bufferpool CSTI8;
alter tablespace CSTI_040 bufferpool CSTI8;

alter tablespace ORD_001 bufferpool OLNORDIORD1;
alter tablespace ORD_002 bufferpool OLNORDIORD2;
alter tablespace ORD_003 bufferpool OLNORDIORD3;
alter tablespace ORD_004 bufferpool OLNORDIORD4;
alter tablespace ORD_005 bufferpool OLNORDIORD5;
alter tablespace ORD_006 bufferpool OLNORDIORD6;
alter tablespace ORD_007 bufferpool OLNORDIORD7;
alter tablespace ORD_008 bufferpool OLNORDIORD8;

alter tablespace OLN_001 bufferpool OLNORDIORD1;
alter tablespace OLN_002 bufferpool OLNORDIORD2;
alter tablespace OLN_003 bufferpool OLNORDIORD3;
alter tablespace OLN_004 bufferpool OLNORDIORD4;
alter tablespace OLN_005 bufferpool OLNORDIORD5;
alter tablespace OLN_006 bufferpool OLNORDIORD6;
alter tablespace OLN_007 bufferpool OLNORDIORD7;
alter tablespace OLN_008 bufferpool OLNORDIORD8;

alter tablespace ORDI_001 bufferpool OLNORDIORD1;
alter tablespace ORDI_002 bufferpool OLNORDIORD2;
alter tablespace ORDI_003 bufferpool OLNORDIORD3;
alter tablespace ORDI_004 bufferpool OLNORDIORD4;

```

```

alter tablespace ORDI_005 bufferpool OLNORDIORD5;
alter tablespace ORDI_006 bufferpool OLNORDIORD6;
alter tablespace ORDI_007 bufferpool OLNORDIORD7;
alter tablespace ORDI_008 bufferpool OLNORDIORD8;

alter tablespace HST_001 bufferpool HST1;
alter tablespace HST_002 bufferpool HST2;
alter tablespace HST_003 bufferpool HST3;
alter tablespace HST_004 bufferpool HST4;
alter tablespace HST_005 bufferpool HST5;
alter tablespace HST_006 bufferpool HST6;
alter tablespace HST_007 bufferpool HST7;
alter tablespace HST_008 bufferpool HST8;

alter tablespace NEWA_001 bufferpool NEW1;
alter tablespace NEWA_002 bufferpool NEW2;
alter tablespace NEWA_003 bufferpool NEW3;
alter tablespace NEWA_004 bufferpool NEW4;
alter tablespace NEWA_005 bufferpool NEW5;
alter tablespace NEWA_006 bufferpool NEW6;
alter tablespace NEWA_007 bufferpool NEW7;
alter tablespace NEWA_008 bufferpool NEW8;

alter tablespace NEWB_001 bufferpool NEW1;
alter tablespace NEWB_002 bufferpool NEW2;
alter tablespace NEWB_003 bufferpool NEW3;
alter tablespace NEWB_004 bufferpool NEW4;
alter tablespace NEWB_005 bufferpool NEW5;
alter tablespace NEWB_006 bufferpool NEW6;
alter tablespace NEWB_007 bufferpool NEW7;
alter tablespace NEWB_008 bufferpool NEW8;

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 tpcc;
alter bufferpool IBMDEFAULTBP size 50;
alter bufferpool IBMDEFAULT8K size 16;
alter bufferpool IBMDEFAULT16K size 10;
alter bufferpool ITM size 1242;
alter bufferpool WDS1 size 918;
alter bufferpool WDS2 size 918;
alter bufferpool WDS3 size 918;
alter bufferpool WDS4 size 918;
alter bufferpool WDS5 size 918;
alter bufferpool WDS6 size 918;
alter bufferpool WDS7 size 918;
alter bufferpool WDS8 size 918;
alter bufferpool STK1 size 3350000;
alter bufferpool STK2 size 3350000;
alter bufferpool STK3 size 3350000;
alter bufferpool STK4 size 3350000;
alter bufferpool STK5 size 3350000;
alter bufferpool STK6 size 3350000;
alter bufferpool STK7 size 3350000;
alter bufferpool STK8 size 3350000;
alter bufferpool CST1 size 6000;
alter bufferpool CST2 size 6000;
alter bufferpool CST3 size 6000;
alter bufferpool CST4 size 6000;
alter bufferpool CST5 size 6000;
alter bufferpool CST6 size 6000;
alter bufferpool CST7 size 6000;
alter bufferpool CST8 size 6000;
alter bufferpool NEW1 size 52000;
alter bufferpool NEW2 size 52000;
alter bufferpool NEW3 size 52000;
alter bufferpool NEW4 size 52000;
alter bufferpool NEW5 size 52000;
alter bufferpool NEW6 size 52000;
alter bufferpool NEW7 size 52000;
alter bufferpool NEW8 size 52000;
alter bufferpool OLNORDIORD1 size 190000;
alter bufferpool OLNORDIORD2 size 190000;
alter bufferpool OLNORDIORD3 size 190000;
alter bufferpool OLNORDIORD4 size 190000;
alter bufferpool OLNORDIORD5 size 190000;
alter bufferpool OLNORDIORD6 size 190000;
alter bufferpool OLNORDIORD7 size 190000;
alter bufferpool OLNORDIORD8 size 190000;
alter bufferpool HST1 size 132;
alter bufferpool HST2 size 132;
alter bufferpool HST3 size 132;
alter bufferpool HST4 size 132;
alter bufferpool HST5 size 132;
alter bufferpool HST6 size 132;
alter bufferpool HST7 size 132;
alter bufferpool HST8 size 132;
alter bufferpool CSTI1 size 31000;
alter bufferpool CSTI2 size 31000;
alter bufferpool CSTI3 size 31000;
alter bufferpool CSTI4 size 31000;
alter bufferpool CSTI5 size 31000;
```

```
alter bufferpool CSTI6 size 31000;
alter bufferpool CSTI7 size 31000;
alter bufferpool CSTI8 size 31000;
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 ITM size 500 pagesize 8192;

create bufferpool WDS1 size 500 pagesize 4096;
create bufferpool STK1 size 500 pagesize 4096;
create bufferpool CST1 size 500 pagesize 4096;
create bufferpool NEW1 size 500 pagesize 4096;
create bufferpool OLNORDIORD1 size 500 pagesize 8192;
create bufferpool HST1 size 500 pagesize 16384;
create bufferpool CSTI1 size 500 pagesize 8192;
create bufferpool WDS2 size 500 pagesize 4096;
create bufferpool STK2 size 500 pagesize 4096;
create bufferpool CST2 size 500 pagesize 4096;
create bufferpool NEW2 size 500 pagesize 4096;
create bufferpool OLNORDIORD2 size 500 pagesize 8192;
create bufferpool HST2 size 500 pagesize 16384;
create bufferpool CSTI2 size 500 pagesize 8192;
create bufferpool WDS3 size 500 pagesize 4096;
create bufferpool STK3 size 500 pagesize 4096;
create bufferpool CST3 size 500 pagesize 4096;
create bufferpool NEW3 size 500 pagesize 4096;
create bufferpool OLNORDIORD3 size 500 pagesize 8192;
create bufferpool HST3 size 500 pagesize 16384;
create bufferpool CSTI3 size 500 pagesize 8192;
create bufferpool WDS4 size 500 pagesize 4096;
create bufferpool STK4 size 500 pagesize 4096;
create bufferpool CST4 size 500 pagesize 4096;
create bufferpool NEW4 size 500 pagesize 4096;
create bufferpool OLNORDIORD4 size 500 pagesize 8192;
create bufferpool HST4 size 500 pagesize 16384;
create bufferpool CSTI4 size 500 pagesize 8192;
create bufferpool WDS5 size 500 pagesize 4096;
create bufferpool STK5 size 500 pagesize 4096;
create bufferpool CST5 size 500 pagesize 4096;
create bufferpool NEW5 size 500 pagesize 4096;
create bufferpool OLNORDIORD5 size 500 pagesize 8192;
create bufferpool HST5 size 500 pagesize 16384;
create bufferpool CSTI5 size 500 pagesize 8192;
create bufferpool WDS6 size 500 pagesize 4096;
create bufferpool STK6 size 500 pagesize 4096;
create bufferpool CST6 size 500 pagesize 4096;
create bufferpool NEW6 size 500 pagesize 4096;
create bufferpool OLNORDIORD6 size 500 pagesize 8192;
create bufferpool HST6 size 500 pagesize 16384;
create bufferpool CSTI6 size 500 pagesize 8192;
```

```

create bufferpool WDS7 size 500 pagesize 4096;
create bufferpool STK7 size 500 pagesize 4096;
create bufferpool CST7 size 500 pagesize 4096;
create bufferpool NEW7 size 500 pagesize 4096;
create bufferpool OLNORDIORD7 size 500 pagesize 8192;
create bufferpool HST7 size 500 pagesize 16384;
create bufferpool CSTI7 size 500 pagesize 8192;
create bufferpool WDS8 size 500 pagesize 4096;
create bufferpool STK8 size 500 pagesize 4096;
create bufferpool CST8 size 500 pagesize 4096;
create bufferpool NEW8 size 500 pagesize 4096;
create bufferpool OLNORDIORD8 size 500 pagesize 8192;
create bufferpool HST8 size 500 pagesize 16384;
create bufferpool CSTI8 size 500 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 - 2005
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
-----

```

```

drop database tpcc;
create database tpcc collate using identity;

```

altdbsp_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 CSTI_008 prefetchsize 0;
alter tablespace CSTI_009 prefetchsize 0;
alter tablespace CSTI_010 prefetchsize 0;
alter tablespace CSTI_011 prefetchsize 0;
alter tablespace CSTI_012 prefetchsize 0;
alter tablespace CSTI_013 prefetchsize 0;
alter tablespace CSTI_014 prefetchsize 0;
alter tablespace CSTI_015 prefetchsize 0;
alter tablespace CSTI_016 prefetchsize 0;
alter tablespace CSTI_017 prefetchsize 0;
alter tablespace CSTI_018 prefetchsize 0;
alter tablespace CSTI_019 prefetchsize 0;
alter tablespace CSTI_020 prefetchsize 0;
alter tablespace CSTI_021 prefetchsize 0;
alter tablespace CSTI_022 prefetchsize 0;
alter tablespace CSTI_023 prefetchsize 0;
alter tablespace CSTI_024 prefetchsize 0;
alter tablespace CSTI_025 prefetchsize 0;
alter tablespace CSTI_026 prefetchsize 0;
alter tablespace CSTI_027 prefetchsize 0;

```

```

alter tablespace CSTI_028 prefetchsize 0;
alter tablespace CSTI_029 prefetchsize 0;
alter tablespace CSTI_030 prefetchsize 0;
alter tablespace CSTI_031 prefetchsize 0;
alter tablespace CSTI_032 prefetchsize 0;
alter tablespace CSTI_033 prefetchsize 0;
alter tablespace CSTI_034 prefetchsize 0;
alter tablespace CSTI_035 prefetchsize 0;
alter tablespace CSTI_036 prefetchsize 0;
alter tablespace CSTI_037 prefetchsize 0;
alter tablespace CSTI_038 prefetchsize 0;
alter tablespace CSTI_039 prefetchsize 0;
alter tablespace CSTI_040 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 CST_008 prefetchsize 0;
alter tablespace CST_009 prefetchsize 0;
alter tablespace CST_010 prefetchsize 0;
alter tablespace CST_011 prefetchsize 0;
alter tablespace CST_012 prefetchsize 0;
alter tablespace CST_013 prefetchsize 0;
alter tablespace CST_014 prefetchsize 0;
alter tablespace CST_015 prefetchsize 0;
alter tablespace CST_016 prefetchsize 0;
alter tablespace CST_017 prefetchsize 0;
alter tablespace CST_018 prefetchsize 0;
alter tablespace CST_019 prefetchsize 0;
alter tablespace CST_020 prefetchsize 0;
alter tablespace CST_021 prefetchsize 0;
alter tablespace CST_022 prefetchsize 0;
alter tablespace CST_023 prefetchsize 0;
alter tablespace CST_024 prefetchsize 0;
alter tablespace CST_025 prefetchsize 0;
alter tablespace CST_026 prefetchsize 0;
alter tablespace CST_027 prefetchsize 0;
alter tablespace CST_028 prefetchsize 0;
alter tablespace CST_029 prefetchsize 0;
alter tablespace CST_030 prefetchsize 0;
alter tablespace CST_031 prefetchsize 0;
alter tablespace CST_032 prefetchsize 0;
alter tablespace CST_033 prefetchsize 0;
alter tablespace CST_034 prefetchsize 0;
alter tablespace CST_035 prefetchsize 0;
alter tablespace CST_036 prefetchsize 0;
alter tablespace CST_037 prefetchsize 0;
alter tablespace CST_038 prefetchsize 0;
alter tablespace CST_039 prefetchsize 0;
alter tablespace CST_040 prefetchsize 0;
alter tablespace DIS_001 prefetchsize 0;
alter tablespace DIS_002 prefetchsize 0;
alter tablespace DIS_003 prefetchsize 0;
alter tablespace DIS_004 prefetchsize 0;
alter tablespace DIS_005 prefetchsize 0;
alter tablespace DIS_006 prefetchsize 0;
alter tablespace DIS_007 prefetchsize 0;
alter tablespace DIS_008 prefetchsize 0;
alter tablespace HST_001 prefetchsize 0;
alter tablespace HST_002 prefetchsize 0;
alter tablespace HST_003 prefetchsize 0;
alter tablespace HST_004 prefetchsize 0;
alter tablespace HST_005 prefetchsize 0;
alter tablespace HST_006 prefetchsize 0;
alter tablespace HST_007 prefetchsize 0;

```



```

alter tablespace WAR_001 prefetchsize 4096;
alter tablespace WAR_002 prefetchsize 4096;
alter tablespace WAR_003 prefetchsize 4096;
alter tablespace WAR_004 prefetchsize 4096;
alter tablespace WAR_005 prefetchsize 4096;
alter tablespace WAR_006 prefetchsize 4096;
alter tablespace WAR_007 prefetchsize 4096;
alter tablespace WAR_008 prefetchsize 4096;
connect reset;

```

crconst_customer_all.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 544);
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 545 AND 1088);
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 1089 AND 1632);
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 1633 AND 2176);
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 2177 AND 2720);
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 2721 AND 3264);
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 BETWEEN 3265 AND 3808);
SET INTEGRITY FOR CUSTOMER7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER8 OFF;
ALTER TABLE CUSTOMER8 DROP CONSTRAINT CUSTOMER8CKC;
ALTER TABLE CUSTOMER8 ADD CONSTRAINT CUSTOMER8CKC
CHECK (C_W_ID BETWEEN 3809 AND 4352);
SET INTEGRITY FOR CUSTOMER8 ALL IMMEDIATE UNCHECKED;
connect reset;

```

```

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER9 OFF;
ALTER TABLE CUSTOMER9 DROP CONSTRAINT CUSTOMER9CKC;
ALTER TABLE CUSTOMER9 ADD CONSTRAINT CUSTOMER9CKC
CHECK (C_W_ID BETWEEN 4353 AND 4896);
SET INTEGRITY FOR CUSTOMER9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER10 OFF;
ALTER TABLE CUSTOMER10 DROP CONSTRAINT CUSTOMER10CKC;
ALTER TABLE CUSTOMER10 ADD CONSTRAINT CUSTOMER10CKC
CHECK (C_W_ID BETWEEN 4897 AND 5440);
SET INTEGRITY FOR CUSTOMER10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER11 OFF;
ALTER TABLE CUSTOMER11 DROP CONSTRAINT CUSTOMER11CKC;
ALTER TABLE CUSTOMER11 ADD CONSTRAINT CUSTOMER11CKC
CHECK (C_W_ID BETWEEN 5441 AND 5984);
SET INTEGRITY FOR CUSTOMER11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER12 OFF;
ALTER TABLE CUSTOMER12 DROP CONSTRAINT CUSTOMER12CKC;
ALTER TABLE CUSTOMER12 ADD CONSTRAINT CUSTOMER12CKC
CHECK (C_W_ID BETWEEN 5985 AND 6528);
SET INTEGRITY FOR CUSTOMER12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER13 OFF;
ALTER TABLE CUSTOMER13 DROP CONSTRAINT CUSTOMER13CKC;
ALTER TABLE CUSTOMER13 ADD CONSTRAINT CUSTOMER13CKC
CHECK (C_W_ID BETWEEN 6529 AND 7072);
SET INTEGRITY FOR CUSTOMER13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER14 OFF;
ALTER TABLE CUSTOMER14 DROP CONSTRAINT CUSTOMER14CKC;
ALTER TABLE CUSTOMER14 ADD CONSTRAINT CUSTOMER14CKC
CHECK (C_W_ID BETWEEN 7073 AND 7616);
SET INTEGRITY FOR CUSTOMER14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER15 OFF;
ALTER TABLE CUSTOMER15 DROP CONSTRAINT CUSTOMER15CKC;
ALTER TABLE CUSTOMER15 ADD CONSTRAINT CUSTOMER15CKC
CHECK (C_W_ID BETWEEN 7617 AND 8160);
SET INTEGRITY FOR CUSTOMER15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER16 OFF;
ALTER TABLE CUSTOMER16 DROP CONSTRAINT CUSTOMER16CKC;
ALTER TABLE CUSTOMER16 ADD CONSTRAINT CUSTOMER16CKC
CHECK (C_W_ID BETWEEN 8161 AND 8704);
SET INTEGRITY FOR CUSTOMER16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER17 OFF;
ALTER TABLE CUSTOMER17 DROP CONSTRAINT CUSTOMER17CKC;
ALTER TABLE CUSTOMER17 ADD CONSTRAINT CUSTOMER17CKC
CHECK (C_W_ID BETWEEN 8705 AND 9248);
SET INTEGRITY FOR CUSTOMER17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER18 OFF;
ALTER TABLE CUSTOMER18 DROP CONSTRAINT CUSTOMER18CKC;
ALTER TABLE CUSTOMER18 ADD CONSTRAINT CUSTOMER18CKC
CHECK (C_W_ID BETWEEN 9249 AND 9792);

```

```

SET INTEGRITY FOR CUSTOMER18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER19 OFF;
ALTER TABLE CUSTOMER19 DROP CONSTRAINT CUSTOMER19CKC;
ALTER TABLE CUSTOMER19 ADD CONSTRAINT CUSTOMER19CKC
CHECK (C_W_ID BETWEEN 9793 AND 10336);
SET INTEGRITY FOR CUSTOMER19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER20 OFF;
ALTER TABLE CUSTOMER20 DROP CONSTRAINT CUSTOMER20CKC;
ALTER TABLE CUSTOMER20 ADD CONSTRAINT CUSTOMER20CKC
CHECK (C_W_ID BETWEEN 10337 AND 10880);
SET INTEGRITY FOR CUSTOMER20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER21 OFF;
ALTER TABLE CUSTOMER21 DROP CONSTRAINT CUSTOMER21CKC;
ALTER TABLE CUSTOMER21 ADD CONSTRAINT CUSTOMER21CKC
CHECK (C_W_ID BETWEEN 10881 AND 11424);
SET INTEGRITY FOR CUSTOMER21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER22 OFF;
ALTER TABLE CUSTOMER22 DROP CONSTRAINT CUSTOMER22CKC;
ALTER TABLE CUSTOMER22 ADD CONSTRAINT CUSTOMER22CKC
CHECK (C_W_ID BETWEEN 11425 AND 11968);
SET INTEGRITY FOR CUSTOMER22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER23 OFF;
ALTER TABLE CUSTOMER23 DROP CONSTRAINT CUSTOMER23CKC;
ALTER TABLE CUSTOMER23 ADD CONSTRAINT CUSTOMER23CKC
CHECK (C_W_ID BETWEEN 11969 AND 12512);
SET INTEGRITY FOR CUSTOMER23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER24 OFF;
ALTER TABLE CUSTOMER24 DROP CONSTRAINT CUSTOMER24CKC;
ALTER TABLE CUSTOMER24 ADD CONSTRAINT CUSTOMER24CKC
CHECK (C_W_ID BETWEEN 12513 AND 13056);
SET INTEGRITY FOR CUSTOMER24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER25 OFF;
ALTER TABLE CUSTOMER25 DROP CONSTRAINT CUSTOMER25CKC;
ALTER TABLE CUSTOMER25 ADD CONSTRAINT CUSTOMER25CKC
CHECK (C_W_ID BETWEEN 13057 AND 13600);
SET INTEGRITY FOR CUSTOMER25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER26 OFF;
ALTER TABLE CUSTOMER26 DROP CONSTRAINT CUSTOMER26CKC;
ALTER TABLE CUSTOMER26 ADD CONSTRAINT CUSTOMER26CKC
CHECK (C_W_ID BETWEEN 13601 AND 14144);
SET INTEGRITY FOR CUSTOMER26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER27 OFF;
ALTER TABLE CUSTOMER27 DROP CONSTRAINT CUSTOMER27CKC;
ALTER TABLE CUSTOMER27 ADD CONSTRAINT CUSTOMER27CKC
CHECK (C_W_ID BETWEEN 14145 AND 14688);
SET INTEGRITY FOR CUSTOMER27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER28 OFF;
ALTER TABLE CUSTOMER28 DROP CONSTRAINT CUSTOMER28CKC;
ALTER TABLE CUSTOMER28 ADD CONSTRAINT CUSTOMER28CKC
CHECK (C_W_ID BETWEEN 14689 AND 15232);
SET INTEGRITY FOR CUSTOMER28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER29 OFF;
ALTER TABLE CUSTOMER29 DROP CONSTRAINT CUSTOMER29CKC;
ALTER TABLE CUSTOMER29 ADD CONSTRAINT CUSTOMER29CKC
CHECK (C_W_ID BETWEEN 15233 AND 15776);
SET INTEGRITY FOR CUSTOMER29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER30 OFF;
ALTER TABLE CUSTOMER30 DROP CONSTRAINT CUSTOMER30CKC;
ALTER TABLE CUSTOMER30 ADD CONSTRAINT CUSTOMER30CKC
CHECK (C_W_ID BETWEEN 15777 AND 16320);
SET INTEGRITY FOR CUSTOMER30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER31 OFF;
ALTER TABLE CUSTOMER31 DROP CONSTRAINT CUSTOMER31CKC;
ALTER TABLE CUSTOMER31 ADD CONSTRAINT CUSTOMER31CKC
CHECK (C_W_ID BETWEEN 16321 AND 16864);
SET INTEGRITY FOR CUSTOMER31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER32 OFF;
ALTER TABLE CUSTOMER32 DROP CONSTRAINT CUSTOMER32CKC;
ALTER TABLE CUSTOMER32 ADD CONSTRAINT CUSTOMER32CKC
CHECK (C_W_ID BETWEEN 16865 AND 17408);
SET INTEGRITY FOR CUSTOMER32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER33 OFF;
ALTER TABLE CUSTOMER33 DROP CONSTRAINT CUSTOMER33CKC;
ALTER TABLE CUSTOMER33 ADD CONSTRAINT CUSTOMER33CKC
CHECK (C_W_ID BETWEEN 17409 AND 17952);
SET INTEGRITY FOR CUSTOMER33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER34 OFF;
ALTER TABLE CUSTOMER34 DROP CONSTRAINT CUSTOMER34CKC;
ALTER TABLE CUSTOMER34 ADD CONSTRAINT CUSTOMER34CKC
CHECK (C_W_ID BETWEEN 17953 AND 18496);
SET INTEGRITY FOR CUSTOMER34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER35 OFF;
ALTER TABLE CUSTOMER35 DROP CONSTRAINT CUSTOMER35CKC;
ALTER TABLE CUSTOMER35 ADD CONSTRAINT CUSTOMER35CKC
CHECK (C_W_ID BETWEEN 18497 AND 19040);
SET INTEGRITY FOR CUSTOMER35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER36 OFF;
ALTER TABLE CUSTOMER36 DROP CONSTRAINT CUSTOMER36CKC;
ALTER TABLE CUSTOMER36 ADD CONSTRAINT CUSTOMER36CKC
CHECK (C_W_ID BETWEEN 19041 AND 19584);
SET INTEGRITY FOR CUSTOMER36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER37 OFF;
ALTER TABLE CUSTOMER37 DROP CONSTRAINT CUSTOMER37CKC;
ALTER TABLE CUSTOMER37 ADD CONSTRAINT CUSTOMER37CKC
CHECK (C_W_ID BETWEEN 19585 AND 20128);
SET INTEGRITY FOR CUSTOMER37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;

```



```

SET INTEGRITY FOR CUSTOMER38 OFF;
ALTER TABLE CUSTOMER38 DROP CONSTRAINT CUSTOMER38CKC;
ALTER TABLE CUSTOMER38 ADD CONSTRAINT CUSTOMER38CKC
CHECK (C_W_ID BETWEEN 20129 AND 20672);
SET INTEGRITY FOR CUSTOMER38 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER39 OFF;
ALTER TABLE CUSTOMER39 DROP CONSTRAINT CUSTOMER39CKC;
ALTER TABLE CUSTOMER39 ADD CONSTRAINT CUSTOMER39CKC
CHECK (C_W_ID BETWEEN 20673 AND 21216);
SET INTEGRITY FOR CUSTOMER39 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER40 OFF;
ALTER TABLE CUSTOMER40 DROP CONSTRAINT CUSTOMER40CKC;
ALTER TABLE CUSTOMER40 ADD CONSTRAINT CUSTOMER40CKC
CHECK (C_W_ID >= 21217);
SET INTEGRITY FOR CUSTOMER40 ALL IMMEDIATE UNCHECKED;
connect reset;

```

crconst_district_all.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT1 OFF;
ALTER TABLE DISTRICT1 DROP CONSTRAINT DISTRICT1CKC;
ALTER TABLE DISTRICT1 ADD CONSTRAINT DISTRICT1CKC CHECK
(D_W_ID BETWEEN 1 AND 2720);
SET INTEGRITY FOR DISTRICT1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT2 OFF;
ALTER TABLE DISTRICT2 DROP CONSTRAINT DISTRICT2CKC;
ALTER TABLE DISTRICT2 ADD CONSTRAINT DISTRICT2CKC CHECK
(D_W_ID BETWEEN 2721 AND 5440);
SET INTEGRITY FOR DISTRICT2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT3 OFF;
ALTER TABLE DISTRICT3 DROP CONSTRAINT DISTRICT3CKC;
ALTER TABLE DISTRICT3 ADD CONSTRAINT DISTRICT3CKC CHECK
(D_W_ID BETWEEN 5441 AND 8160);
SET INTEGRITY FOR DISTRICT3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT4 OFF;
ALTER TABLE DISTRICT4 DROP CONSTRAINT DISTRICT4CKC;
ALTER TABLE DISTRICT4 ADD CONSTRAINT DISTRICT4CKC CHECK
(D_W_ID BETWEEN 8161 AND 10880);
SET INTEGRITY FOR DISTRICT4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT5 OFF;
ALTER TABLE DISTRICT5 DROP CONSTRAINT DISTRICT5CKC;
ALTER TABLE DISTRICT5 ADD CONSTRAINT DISTRICT5CKC CHECK
(D_W_ID BETWEEN 10881 AND 13600);
SET INTEGRITY FOR DISTRICT5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT6 OFF;
ALTER TABLE DISTRICT6 DROP CONSTRAINT DISTRICT6CKC;
ALTER TABLE DISTRICT6 ADD CONSTRAINT DISTRICT6CKC CHECK
(D_W_ID BETWEEN 13601 AND 16320);
SET INTEGRITY FOR DISTRICT6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT7 OFF;
ALTER TABLE DISTRICT7 DROP CONSTRAINT DISTRICT7CKC;

```

```

ALTER TABLE DISTRICT7 ADD CONSTRAINT DISTRICT7CKC CHECK
(D_W_ID BETWEEN 16321 AND 19040);
SET INTEGRITY FOR DISTRICT7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT8 OFF;
ALTER TABLE DISTRICT8 DROP CONSTRAINT DISTRICT8CKC;
ALTER TABLE DISTRICT8 ADD CONSTRAINT DISTRICT8CKC CHECK
(D_W_ID >= 19041);
SET INTEGRITY FOR DISTRICT8 ALL IMMEDIATE UNCHECKED;
connect reset;

```

crconst_history_all.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY1 OFF;
ALTER TABLE HISTORY1 DROP CONSTRAINT HISTORY1CKC;
ALTER TABLE HISTORY1 ADD CONSTRAINT HISTORY1CKC CHECK
(H_W_ID BETWEEN 1 AND 2720);
SET INTEGRITY FOR HISTORY1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY2 OFF;
ALTER TABLE HISTORY2 DROP CONSTRAINT HISTORY2CKC;
ALTER TABLE HISTORY2 ADD CONSTRAINT HISTORY2CKC CHECK
(H_W_ID BETWEEN 2721 AND 5440);
SET INTEGRITY FOR HISTORY2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY3 OFF;
ALTER TABLE HISTORY3 DROP CONSTRAINT HISTORY3CKC;
ALTER TABLE HISTORY3 ADD CONSTRAINT HISTORY3CKC CHECK
(H_W_ID BETWEEN 5441 AND 8160);
SET INTEGRITY FOR HISTORY3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY4 OFF;
ALTER TABLE HISTORY4 DROP CONSTRAINT HISTORY4CKC;
ALTER TABLE HISTORY4 ADD CONSTRAINT HISTORY4CKC CHECK
(H_W_ID BETWEEN 8161 AND 10880);
SET INTEGRITY FOR HISTORY4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY5 OFF;
ALTER TABLE HISTORY5 DROP CONSTRAINT HISTORY5CKC;
ALTER TABLE HISTORY5 ADD CONSTRAINT HISTORY5CKC CHECK
(H_W_ID BETWEEN 10881 AND 13600);
SET INTEGRITY FOR HISTORY5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY6 OFF;
ALTER TABLE HISTORY6 DROP CONSTRAINT HISTORY6CKC;
ALTER TABLE HISTORY6 ADD CONSTRAINT HISTORY6CKC CHECK
(H_W_ID BETWEEN 13601 AND 16320);
SET INTEGRITY FOR HISTORY6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY7 OFF;
ALTER TABLE HISTORY7 DROP CONSTRAINT HISTORY7CKC;
ALTER TABLE HISTORY7 ADD CONSTRAINT HISTORY7CKC CHECK
(H_W_ID BETWEEN 16321 AND 19040);
SET INTEGRITY FOR HISTORY7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY8 OFF;
ALTER TABLE HISTORY8 DROP CONSTRAINT HISTORY8CKC;

```

```
ALTER TABLE HISTORY8 ADD CONSTRAINT HISTORY8CKC CHECK
(H_W_ID >= 19041);
SET INTEGRITY FOR HISTORY8 ALL IMMEDIATE UNCHECKED;
connect reset;
```

crconst_new_order_all.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 2720) AND
(NO_O_ID <= 3675));
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 2721 AND 5440)
AND (NO_O_ID <= 3675));
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 5441 AND 8160)
AND (NO_O_ID <= 3675));
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 8161 AND 10880)
AND (NO_O_ID <= 3675));
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 10881 AND 13600)
AND (NO_O_ID <= 3675));
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 13601 AND 16320)
AND (NO_O_ID <= 3675));
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 BETWEEN 16321 AND 19040)
AND (NO_O_ID <= 3675));
```

```
SET INTEGRITY FOR NEW_ORDERA7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA8 OFF;
ALTER TABLE NEW_ORDERA8 DROP CONSTRAINT
NEW_ORDERA8CKC;
ALTER TABLE NEW_ORDERA8 ADD CONSTRAINT
NEW_ORDERA8CKC CHECK ((NO_W_ID >= 19041) AND (NO_O_ID <=
3675));
SET INTEGRITY FOR NEW_ORDERA8 ALL IMMEDIATE UNCHECKED;
connect reset;
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 2720) AND
(NO_O_ID >= 3676));
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 2721 AND 5440)
AND (NO_O_ID >= 3676));
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 5441 AND 8160)
AND (NO_O_ID >= 3676));
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 8161 AND 10880)
AND (NO_O_ID >= 3676));
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 10881 AND 13600)
AND (NO_O_ID >= 3676));
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 13601 AND 16320)
AND (NO_O_ID >= 3676));
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 BETWEEN 16321 AND 19040)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB8 OFF;
ALTER TABLE NEW_ORDERB8 DROP CONSTRAINT
NEW_ORDERB8CKC;
ALTER TABLE NEW_ORDERB8 ADD CONSTRAINT
NEW_ORDERB8CKC CHECK ((NO_W_ID >= 19041) AND (NO_O_ID >=
3676));
SET INTEGRITY FOR NEW_ORDERB8 ALL IMMEDIATE UNCHECKED;
connect reset;

```

crconst_order_line_all.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 2720);
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 2721 AND 5440);
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 5441 AND 8160);
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 8161 AND 10880);
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 10881 AND 13600);
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 13601 AND 16320);
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 BETWEEN 16321 AND 19040);
SET INTEGRITY FOR ORDER_LINE7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE8 OFF;
ALTER TABLE ORDER_LINE8 DROP CONSTRAINT
ORDER_LINE8CKC;
ALTER TABLE ORDER_LINE8 ADD CONSTRAINT ORDER_LINE8CKC
CHECK (OL_W_ID >= 19041);
SET INTEGRITY FOR ORDER_LINE8 ALL IMMEDIATE UNCHECKED;
connect reset;

```

crconst_orders_all.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS1 OFF;
ALTER TABLE ORDERS1 DROP CONSTRAINT ORDERS1CKC;
ALTER TABLE ORDERS1 ADD CONSTRAINT ORDERS1CKC CHECK
(O_W_ID BETWEEN 1 AND 2720);
SET INTEGRITY FOR ORDERS1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS2 OFF;
ALTER TABLE ORDERS2 DROP CONSTRAINT ORDERS2CKC;
ALTER TABLE ORDERS2 ADD CONSTRAINT ORDERS2CKC CHECK
(O_W_ID BETWEEN 2721 AND 5440);
SET INTEGRITY FOR ORDERS2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS3 OFF;
ALTER TABLE ORDERS3 DROP CONSTRAINT ORDERS3CKC;
ALTER TABLE ORDERS3 ADD CONSTRAINT ORDERS3CKC CHECK
(O_W_ID BETWEEN 5441 AND 8160);
SET INTEGRITY FOR ORDERS3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS4 OFF;
ALTER TABLE ORDERS4 DROP CONSTRAINT ORDERS4CKC;
ALTER TABLE ORDERS4 ADD CONSTRAINT ORDERS4CKC CHECK
(O_W_ID BETWEEN 8161 AND 10880);
SET INTEGRITY FOR ORDERS4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS5 OFF;
ALTER TABLE ORDERS5 DROP CONSTRAINT ORDERS5CKC;
ALTER TABLE ORDERS5 ADD CONSTRAINT ORDERS5CKC CHECK
(O_W_ID BETWEEN 10881 AND 13600);
SET INTEGRITY FOR ORDERS5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS6 OFF;
ALTER TABLE ORDERS6 DROP CONSTRAINT ORDERS6CKC;
ALTER TABLE ORDERS6 ADD CONSTRAINT ORDERS6CKC CHECK
(O_W_ID BETWEEN 13601 AND 16320);
SET INTEGRITY FOR ORDERS6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS7 OFF;
ALTER TABLE ORDERS7 DROP CONSTRAINT ORDERS7CKC;
ALTER TABLE ORDERS7 ADD CONSTRAINT ORDERS7CKC CHECK
(O_W_ID BETWEEN 16321 AND 19040);
SET INTEGRITY FOR ORDERS7 ALL IMMEDIATE UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS8 OFF;
ALTER TABLE ORDERS8 DROP CONSTRAINT ORDERS8CKC;
ALTER TABLE ORDERS8 ADD CONSTRAINT ORDERS8CKC CHECK
(O_W_ID >= 19041);
SET INTEGRITY FOR ORDERS8 ALL IMMEDIATE UNCHECKED;
connect reset;

```

crconst_stock_all.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 544);
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 545 AND 1088);
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 1089 AND 1632);
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 1633 AND 2176);
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 2177 AND 2720);
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 2721 AND 3264);
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 BETWEEN 3265 AND 3808);
SET INTEGRITY FOR STOCK7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK8 OFF;
ALTER TABLE STOCK8 DROP CONSTRAINT STOCK8CKC;
ALTER TABLE STOCK8 ADD CONSTRAINT STOCK8CKC CHECK
(S_W_ID BETWEEN 3809 AND 4352);
SET INTEGRITY FOR STOCK8 ALL IMMEDIATE UNCHECKED;
connect reset;

```

```

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK9 OFF;
ALTER TABLE STOCK9 DROP CONSTRAINT STOCK9CKC;
ALTER TABLE STOCK9 ADD CONSTRAINT STOCK9CKC CHECK
(S_W_ID BETWEEN 4353 AND 4896);
SET INTEGRITY FOR STOCK9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK10 OFF;
ALTER TABLE STOCK10 DROP CONSTRAINT STOCK10CKC;
ALTER TABLE STOCK10 ADD CONSTRAINT STOCK10CKC CHECK
(S_W_ID BETWEEN 4897 AND 5440);
SET INTEGRITY FOR STOCK10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK11 OFF;
ALTER TABLE STOCK11 DROP CONSTRAINT STOCK11CKC;
ALTER TABLE STOCK11 ADD CONSTRAINT STOCK11CKC CHECK
(S_W_ID BETWEEN 5441 AND 5984);
SET INTEGRITY FOR STOCK11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK12 OFF;
ALTER TABLE STOCK12 DROP CONSTRAINT STOCK12CKC;
ALTER TABLE STOCK12 ADD CONSTRAINT STOCK12CKC CHECK
(S_W_ID BETWEEN 5985 AND 6528);
SET INTEGRITY FOR STOCK12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK13 OFF;
ALTER TABLE STOCK13 DROP CONSTRAINT STOCK13CKC;
ALTER TABLE STOCK13 ADD CONSTRAINT STOCK13CKC CHECK
(S_W_ID BETWEEN 6529 AND 7072);
SET INTEGRITY FOR STOCK13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK14 OFF;
ALTER TABLE STOCK14 DROP CONSTRAINT STOCK14CKC;
ALTER TABLE STOCK14 ADD CONSTRAINT STOCK14CKC CHECK
(S_W_ID BETWEEN 7073 AND 7616);
SET INTEGRITY FOR STOCK14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK15 OFF;
ALTER TABLE STOCK15 DROP CONSTRAINT STOCK15CKC;
ALTER TABLE STOCK15 ADD CONSTRAINT STOCK15CKC CHECK
(S_W_ID BETWEEN 7617 AND 8160);
SET INTEGRITY FOR STOCK15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK16 OFF;
ALTER TABLE STOCK16 DROP CONSTRAINT STOCK16CKC;
ALTER TABLE STOCK16 ADD CONSTRAINT STOCK16CKC CHECK
(S_W_ID BETWEEN 8161 AND 8704);
SET INTEGRITY FOR STOCK16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK17 OFF;
ALTER TABLE STOCK17 DROP CONSTRAINT STOCK17CKC;
ALTER TABLE STOCK17 ADD CONSTRAINT STOCK17CKC CHECK
(S_W_ID BETWEEN 8705 AND 9248);
SET INTEGRITY FOR STOCK17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK18 OFF;
ALTER TABLE STOCK18 DROP CONSTRAINT STOCK18CKC;
ALTER TABLE STOCK18 ADD CONSTRAINT STOCK18CKC CHECK
(S_W_ID BETWEEN 9249 AND 9792);

```

```

SET INTEGRITY FOR STOCK18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK19 OFF;
ALTER TABLE STOCK19 DROP CONSTRAINT STOCK19CKC;
ALTER TABLE STOCK19 ADD CONSTRAINT STOCK19CKC CHECK
(S_W_ID BETWEEN 9793 AND 10336);
SET INTEGRITY FOR STOCK19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK20 OFF;
ALTER TABLE STOCK20 DROP CONSTRAINT STOCK20CKC;
ALTER TABLE STOCK20 ADD CONSTRAINT STOCK20CKC CHECK
(S_W_ID BETWEEN 10337 AND 10880);
SET INTEGRITY FOR STOCK20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK21 OFF;
ALTER TABLE STOCK21 DROP CONSTRAINT STOCK21CKC;
ALTER TABLE STOCK21 ADD CONSTRAINT STOCK21CKC CHECK
(S_W_ID BETWEEN 10881 AND 11424);
SET INTEGRITY FOR STOCK21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK22 OFF;
ALTER TABLE STOCK22 DROP CONSTRAINT STOCK22CKC;
ALTER TABLE STOCK22 ADD CONSTRAINT STOCK22CKC CHECK
(S_W_ID BETWEEN 11425 AND 11968);
SET INTEGRITY FOR STOCK22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK23 OFF;
ALTER TABLE STOCK23 DROP CONSTRAINT STOCK23CKC;
ALTER TABLE STOCK23 ADD CONSTRAINT STOCK23CKC CHECK
(S_W_ID BETWEEN 11969 AND 12512);
SET INTEGRITY FOR STOCK23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK24 OFF;
ALTER TABLE STOCK24 DROP CONSTRAINT STOCK24CKC;
ALTER TABLE STOCK24 ADD CONSTRAINT STOCK24CKC CHECK
(S_W_ID BETWEEN 12513 AND 13056);
SET INTEGRITY FOR STOCK24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK25 OFF;
ALTER TABLE STOCK25 DROP CONSTRAINT STOCK25CKC;
ALTER TABLE STOCK25 ADD CONSTRAINT STOCK25CKC CHECK
(S_W_ID BETWEEN 13057 AND 13600);
SET INTEGRITY FOR STOCK25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK26 OFF;
ALTER TABLE STOCK26 DROP CONSTRAINT STOCK26CKC;
ALTER TABLE STOCK26 ADD CONSTRAINT STOCK26CKC CHECK
(S_W_ID BETWEEN 13601 AND 14144);
SET INTEGRITY FOR STOCK26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK27 OFF;
ALTER TABLE STOCK27 DROP CONSTRAINT STOCK27CKC;
ALTER TABLE STOCK27 ADD CONSTRAINT STOCK27CKC CHECK
(S_W_ID BETWEEN 14145 AND 14688);
SET INTEGRITY FOR STOCK27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK28 OFF;
ALTER TABLE STOCK28 DROP CONSTRAINT STOCK28CKC;

```

```

ALTER TABLE STOCK28 ADD CONSTRAINT STOCK28CKC CHECK
(S_W_ID BETWEEN 14689 AND 15232);
SET INTEGRITY FOR STOCK28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK29 OFF;
ALTER TABLE STOCK29 DROP CONSTRAINT STOCK29CKC;
ALTER TABLE STOCK29 ADD CONSTRAINT STOCK29CKC CHECK
(S_W_ID BETWEEN 15233 AND 15776);
SET INTEGRITY FOR STOCK29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK30 OFF;
ALTER TABLE STOCK30 DROP CONSTRAINT STOCK30CKC;
ALTER TABLE STOCK30 ADD CONSTRAINT STOCK30CKC CHECK
(S_W_ID BETWEEN 15777 AND 16320);
SET INTEGRITY FOR STOCK30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK31 OFF;
ALTER TABLE STOCK31 DROP CONSTRAINT STOCK31CKC;
ALTER TABLE STOCK31 ADD CONSTRAINT STOCK31CKC CHECK
(S_W_ID BETWEEN 16321 AND 16864);
SET INTEGRITY FOR STOCK31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK32 OFF;
ALTER TABLE STOCK32 DROP CONSTRAINT STOCK32CKC;
ALTER TABLE STOCK32 ADD CONSTRAINT STOCK32CKC CHECK
(S_W_ID BETWEEN 16865 AND 17408);
SET INTEGRITY FOR STOCK32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK33 OFF;
ALTER TABLE STOCK33 DROP CONSTRAINT STOCK33CKC;
ALTER TABLE STOCK33 ADD CONSTRAINT STOCK33CKC CHECK
(S_W_ID BETWEEN 17409 AND 17952);
SET INTEGRITY FOR STOCK33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK34 OFF;
ALTER TABLE STOCK34 DROP CONSTRAINT STOCK34CKC;
ALTER TABLE STOCK34 ADD CONSTRAINT STOCK34CKC CHECK
(S_W_ID BETWEEN 17953 AND 18496);
SET INTEGRITY FOR STOCK34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK35 OFF;
ALTER TABLE STOCK35 DROP CONSTRAINT STOCK35CKC;
ALTER TABLE STOCK35 ADD CONSTRAINT STOCK35CKC CHECK
(S_W_ID BETWEEN 18497 AND 19040);
SET INTEGRITY FOR STOCK35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK36 OFF;
ALTER TABLE STOCK36 DROP CONSTRAINT STOCK36CKC;
ALTER TABLE STOCK36 ADD CONSTRAINT STOCK36CKC CHECK
(S_W_ID BETWEEN 19041 AND 19584);
SET INTEGRITY FOR STOCK36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK37 OFF;
ALTER TABLE STOCK37 DROP CONSTRAINT STOCK37CKC;
ALTER TABLE STOCK37 ADD CONSTRAINT STOCK37CKC CHECK
(S_W_ID BETWEEN 19585 AND 20128);
SET INTEGRITY FOR STOCK37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;

```

```

SET INTEGRITY FOR STOCK38 OFF;
ALTER TABLE STOCK38 DROP CONSTRAINT STOCK38CKC;
ALTER TABLE STOCK38 ADD CONSTRAINT STOCK38CKC CHECK
(S_W_ID BETWEEN 20129 AND 20672);
SET INTEGRITY FOR STOCK38 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK39 OFF;
ALTER TABLE STOCK39 DROP CONSTRAINT STOCK39CKC;
ALTER TABLE STOCK39 ADD CONSTRAINT STOCK39CKC CHECK
(S_W_ID BETWEEN 20673 AND 21216);
SET INTEGRITY FOR STOCK39 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK40 OFF;
ALTER TABLE STOCK40 DROP CONSTRAINT STOCK40CKC;
ALTER TABLE STOCK40 ADD CONSTRAINT STOCK40CKC CHECK
(S_W_ID >= 21217);
SET INTEGRITY FOR STOCK40 ALL IMMEDIATE UNCHECKED;
connect reset;

```

crconst_warehouse_all.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE1 OFF;
ALTER TABLE WAREHOUSE1 DROP CONSTRAINT
WAREHOUSE1CKC;
ALTER TABLE WAREHOUSE1 ADD CONSTRAINT WAREHOUSE1CKC
CHECK (W_ID BETWEEN 1 AND 2720);
SET INTEGRITY FOR WAREHOUSE1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE2 OFF;
ALTER TABLE WAREHOUSE2 DROP CONSTRAINT
WAREHOUSE2CKC;
ALTER TABLE WAREHOUSE2 ADD CONSTRAINT WAREHOUSE2CKC
CHECK (W_ID BETWEEN 2721 AND 5440);
SET INTEGRITY FOR WAREHOUSE2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE3 OFF;
ALTER TABLE WAREHOUSE3 DROP CONSTRAINT
WAREHOUSE3CKC;
ALTER TABLE WAREHOUSE3 ADD CONSTRAINT WAREHOUSE3CKC
CHECK (W_ID BETWEEN 5441 AND 8160);
SET INTEGRITY FOR WAREHOUSE3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE4 OFF;
ALTER TABLE WAREHOUSE4 DROP CONSTRAINT
WAREHOUSE4CKC;
ALTER TABLE WAREHOUSE4 ADD CONSTRAINT WAREHOUSE4CKC
CHECK (W_ID BETWEEN 8161 AND 10880);
SET INTEGRITY FOR WAREHOUSE4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE5 OFF;
ALTER TABLE WAREHOUSE5 DROP CONSTRAINT
WAREHOUSE5CKC;
ALTER TABLE WAREHOUSE5 ADD CONSTRAINT WAREHOUSE5CKC
CHECK (W_ID BETWEEN 10881 AND 13600);
SET INTEGRITY FOR WAREHOUSE5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE6 OFF;
ALTER TABLE WAREHOUSE6 DROP CONSTRAINT
WAREHOUSE6CKC;

```

```

ALTER TABLE WAREHOUSE6 ADD CONSTRAINT WAREHOUSE6CKC
CHECK (W_ID BETWEEN 13601 AND 16320);
SET INTEGRITY FOR WAREHOUSE6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE7 OFF;
ALTER TABLE WAREHOUSE7 DROP CONSTRAINT
WAREHOUSE7CKC;
ALTER TABLE WAREHOUSE7 ADD CONSTRAINT WAREHOUSE7CKC
CHECK (W_ID BETWEEN 16321 AND 19040);
SET INTEGRITY FOR WAREHOUSE7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE8 OFF;
ALTER TABLE WAREHOUSE8 DROP CONSTRAINT
WAREHOUSE8CKC;
ALTER TABLE WAREHOUSE8 ADD CONSTRAINT WAREHOUSE8CKC
CHECK (W_ID >= 19041);
SET INTEGRITY FOR WAREHOUSE8 ALL IMMEDIATE UNCHECKED;
connect reset;

```

cridx_cust_idxb_all.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;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB8;
CREATE INDEX CUST_IDXB8

```

```

        ON CUSTOMER8(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_IDXB9;
CREATE INDEX CUST_IDXB9
        ON CUSTOMER9(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_IDXB10;
CREATE INDEX CUST_IDXB10
        ON CUSTOMER10(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_IDXB11;
CREATE INDEX CUST_IDXB11
        ON CUSTOMER11(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_IDXB12;
CREATE INDEX CUST_IDXB12
        ON CUSTOMER12(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_IDXB13;
CREATE INDEX CUST_IDXB13
        ON CUSTOMER13(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_IDXB14;
CREATE INDEX CUST_IDXB14
        ON CUSTOMER14(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_IDXB15;
CREATE INDEX CUST_IDXB15
        ON CUSTOMER15(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_IDXB16;
CREATE INDEX CUST_IDXB16
        ON CUSTOMER16(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_IDXB17;
CREATE INDEX CUST_IDXB17
        ON CUSTOMER17(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_IDXB18;
CREATE INDEX CUST_IDXB18
        ON CUSTOMER18(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_IDXB19;
CREATE INDEX CUST_IDXB19
        ON CUSTOMER19(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_IDXB20;
CREATE INDEX CUST_IDXB20
        ON CUSTOMER20(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_IDXB21;
CREATE INDEX CUST_IDXB21
        ON CUSTOMER21(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_IDXB22;
CREATE INDEX CUST_IDXB22
        ON CUSTOMER22(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_IDXB23;
CREATE INDEX CUST_IDXB23
        ON CUSTOMER23(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_IDXB24;
CREATE INDEX CUST_IDXB24
        ON CUSTOMER24(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_IDXB25;
CREATE INDEX CUST_IDXB25
        ON CUSTOMER25(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_IDXB26;
CREATE INDEX CUST_IDXB26
        ON CUSTOMER26(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_IDXB27;
CREATE INDEX CUST_IDXB27
        ON CUSTOMER27(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_IDXB28;
CREATE INDEX CUST_IDXB28
        ON CUSTOMER28(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_IDXB29;
CREATE INDEX CUST_IDXB29
        ON CUSTOMER29(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_IDXB30;
CREATE INDEX CUST_IDXB30
        ON CUSTOMER30(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_IDXB31;
CREATE INDEX CUST_IDXB31
    ON CUSTOMER31(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_IDXB32;
CREATE INDEX CUST_IDXB32
    ON CUSTOMER32(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_IDXB33;
CREATE INDEX CUST_IDXB33
    ON CUSTOMER33(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_IDXB34;
CREATE INDEX CUST_IDXB34
    ON CUSTOMER34(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_IDXB35;
CREATE INDEX CUST_IDXB35
    ON CUSTOMER35(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_IDXB36;
CREATE INDEX CUST_IDXB36
    ON CUSTOMER36(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_IDXB37;
CREATE INDEX CUST_IDXB37
    ON CUSTOMER37(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_IDXB38;
CREATE INDEX CUST_IDXB38
    ON CUSTOMER38(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_IDXB39;
CREATE INDEX CUST_IDXB39
    ON CUSTOMER39(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_IDXB40;
CREATE INDEX CUST_IDXB40
    ON CUSTOMER40(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;

```

criidx_ordr_idxb_all.ddl

```

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB1;
CREATE INDEX ORDR_IDXB1
    ON ORDERS1(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

```

```

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB2;
CREATE INDEX ORDR_IDXB2
    ON ORDERS2(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB3;
CREATE INDEX ORDR_IDXB3
    ON ORDERS3(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB4;
CREATE INDEX ORDR_IDXB4
    ON ORDERS4(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB5;
CREATE INDEX ORDR_IDXB5
    ON ORDERS5(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB6;
CREATE INDEX ORDR_IDXB6
    ON ORDERS6(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB7;
CREATE INDEX ORDR_IDXB7
    ON ORDERS7(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB8;
CREATE INDEX ORDR_IDXB8
    ON ORDERS8(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

```

crtb_customer_all.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 544,
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 545 ENDING AT 1088,
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 1089 ENDING AT 1632,
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 1633 ENDING AT 2176,
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 2177 ENDING AT 2720,
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 2721 ENDING AT 3264,
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 3265 ENDING AT 3808,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER8;
CREATE TABLE CUSTOMER8

```

```

(
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_008
INDEX IN CSTI_008
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 3809 ENDING AT 4352,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER9;
CREATE TABLE CUSTOMER9

```

```

(
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_009
INDEX IN CSTI_009
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 4353 ENDING AT 4896,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

connect reset;

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER10;
CREATE TABLE CUSTOMER10

```

```

(
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_010
INDEX IN CSTI_010
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 4897 ENDING AT 5440,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

connect reset;

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER11;
CREATE TABLE CUSTOMER11

```

```

(
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_011
INDEX IN CSTI_011
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 5441 ENDING AT 5984,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

connect reset;

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER12;
CREATE TABLE CUSTOMER12

```

```

(
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_012
INDEX IN CSTI_012
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 5985 ENDING AT 6528,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

connect reset;

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER13;
CREATE TABLE CUSTOMER13

```

```

(
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_013
INDEX IN CSTI_013
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 6529 ENDING AT 7072,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE CUSTOMER14;

CREATE TABLE CUSTOMER14

```

(
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_014
INDEX IN CSTI_014
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 7073 ENDING AT 7616,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE CUSTOMER15;

CREATE TABLE CUSTOMER15

```

(
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_015
INDEX IN CSTI_015
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 7617 ENDING AT 8160,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE CUSTOMER16;

CREATE TABLE CUSTOMER16

```

(
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_016
INDEX IN CSTI_016
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 8161 ENDING AT 8704,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE CUSTOMER17;

CREATE TABLE CUSTOMER17

```

(
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_017
INDEX IN CSTI_017
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 8705 ENDING AT 9248,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER18;
CREATE TABLE CUSTOMER18

```

```

(
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_018
INDEX IN CSTI_018
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 9249 ENDING AT 9792,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER19;
CREATE TABLE CUSTOMER19
(

```

```

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_019
INDEX IN CSTI_019
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 9793 ENDING AT 10336,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER20;
CREATE TABLE CUSTOMER20

```

```

(
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_020
INDEX IN CSTI_020
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 10337 ENDING AT 10880,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER21;

```

```

CREATE TABLE CUSTOMER21
(
  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_021
INDEX IN CSTI_021
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 10881 ENDING AT 11424,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

connect reset;

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER22;
CREATE TABLE CUSTOMER22

```

```

(
  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_022
INDEX IN CSTI_022
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 11425 ENDING AT 11968,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

connect reset;

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER23;
CREATE TABLE CUSTOMER23

```

```

(
  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_023
INDEX IN CSTI_023
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 11969 ENDING AT 12512,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

connect reset;

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER24;
CREATE TABLE CUSTOMER24

```

```

(
  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_024
INDEX IN CSTI_024
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 12513 ENDING AT 13056,
  C_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER25;
CREATE TABLE CUSTOMER25
(
  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_025
INDEX IN CSTI_025
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 13057 ENDING AT 13600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER26;
CREATE TABLE CUSTOMER26
(
  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_026
INDEX IN CSTI_026
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 13601 ENDING AT 14144,

```

```

  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER27;
CREATE TABLE CUSTOMER27
(
  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_027
INDEX IN CSTI_027
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 14145 ENDING AT 14688,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER28;
CREATE TABLE CUSTOMER28
(
  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_028
INDEX IN CSTI_028
ORGANIZE BY KEY SEQUENCE (

```

```

C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 14689 ENDING AT 15232,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER29;
CREATE TABLE CUSTOMER29
(
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_029
INDEX IN CSTI_029
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 15233 ENDING AT 15776,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER30;
CREATE TABLE CUSTOMER30
(
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_030

```

```

INDEX IN CSTI_030
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 15777 ENDING AT 16320,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER31;
CREATE TABLE CUSTOMER31
(
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_031
INDEX IN CSTI_031
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 16321 ENDING AT 16864,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER32;
CREATE TABLE CUSTOMER32
(
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_032
INDEX IN CSTI_032
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 16865 ENDING AT 17408,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER33;
CREATE TABLE CUSTOMER33

```

```

(
  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_033
INDEX IN CSTI_033
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 17409 ENDING AT 17952,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER34;
CREATE TABLE CUSTOMER34

```

```

(
  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_034
INDEX IN CSTI_034
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 17953 ENDING AT 18496,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER35;
CREATE TABLE CUSTOMER35

```

```

(
  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_035
INDEX IN CSTI_035
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 18497 ENDING AT 19040,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER36;
CREATE TABLE CUSTOMER36

```

```

(
  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_036
INDEX IN CSTI_036
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 19041 ENDING AT 19584,
C_D_ID STARTING FROM 1 ENDING AT 10
)

```

ALLOW OVERFLOW;

connect reset;

connect to TPCC in share mode;

DROP TABLE CUSTOMER37;

CREATE TABLE CUSTOMER37

```

(
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_037
INDEX IN CSTI_037
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 19585 ENDING AT 20128,
C_D_ID STARTING FROM 1 ENDING AT 10
)

```

ALLOW OVERFLOW;

connect reset;

connect to TPCC in share mode;

DROP TABLE CUSTOMER38;

CREATE TABLE CUSTOMER38

```

(
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_038
INDEX IN CSTI_038
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 20129 ENDING AT 20672,
C_D_ID STARTING FROM 1 ENDING AT 10
)

```

ALLOW OVERFLOW;

connect reset;

connect to TPCC in share mode;

DROP TABLE CUSTOMER39;

CREATE TABLE CUSTOMER39

```

(
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_039
INDEX IN CSTI_039
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 20673 ENDING AT 21216,
C_D_ID STARTING FROM 1 ENDING AT 10
)

```

ALLOW OVERFLOW;

connect reset;

connect to TPCC in share mode;

DROP TABLE CUSTOMER40;

CREATE TABLE CUSTOMER40

```

(
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_040
INDEX IN CSTI_040
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 21217 ENDING AT 21760,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

crtb_district_all.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT1;
CREATE TABLE DISTRICT1

```

```

(
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_001
INDEX IN DIS_001
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 1 ENDING AT 2720
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT2;
CREATE TABLE DISTRICT2

```

```

(
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_002
INDEX IN DIS_002
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 2721 ENDING AT 5440
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT3;
CREATE TABLE DISTRICT3
(
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_003
INDEX IN DIS_003
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 5441 ENDING AT 8160
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT4;
CREATE TABLE DISTRICT4

```

```

(
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_004
INDEX IN DIS_004
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 8161 ENDING AT 10880
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT5;
CREATE TABLE DISTRICT5

```

```

(
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_005
INDEX IN DIS_005

```

```

ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 10881 ENDING AT 13600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT6;
CREATE TABLE DISTRICT6
(
  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_006
INDEX IN DIS_006
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 13601 ENDING AT 16320
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT7;
CREATE TABLE DISTRICT7
(
  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_007
INDEX IN DIS_007
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 16321 ENDING AT 19040
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT8;
CREATE TABLE DISTRICT8
(
  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_008
INDEX IN DIS_008
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 19041 ENDING AT 21760
)
ALLOW OVERFLOW;
connect reset;

crtb_history_all.ddl

connect to TPCC in share mode;
DROP TABLE HISTORY1;
CREATE TABLE HISTORY1
(
  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_001
INDEX IN HST_001;
ALTER TABLE HISTORY1 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY2;
CREATE TABLE HISTORY2
(
  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_002
INDEX IN HST_002;
ALTER TABLE HISTORY2 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY3;
CREATE TABLE HISTORY3
(
  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_003
INDEX IN HST_003;
ALTER TABLE HISTORY3 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY4;
CREATE TABLE HISTORY4
(

```

```

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_004
INDEX IN HST_004;

```

ALTER TABLE HISTORY4 APPEND ON;

connect reset;

connect to TPCC in share mode;

DROP TABLE HISTORY5;

CREATE TABLE HISTORY5

```

(
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_005
INDEX IN HST_005;

```

ALTER TABLE HISTORY5 APPEND ON;

connect reset;

connect to TPCC in share mode;

DROP TABLE HISTORY6;

CREATE TABLE HISTORY6

```

(
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_006
INDEX IN HST_006;

```

ALTER TABLE HISTORY6 APPEND ON;

connect reset;

connect to TPCC in share mode;

DROP TABLE HISTORY7;

CREATE TABLE HISTORY7

```

(
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_007
INDEX IN HST_007;

```

ALTER TABLE HISTORY7 APPEND ON;

connect reset;

connect to TPCC in share mode;

DROP TABLE HISTORY8;

CREATE TABLE HISTORY8

```

(

```

```

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_008
INDEX IN HST_008;

```

ALTER TABLE HISTORY8 APPEND ON;

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_M_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_new_order_all.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 2720,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
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 2721 ENDING AT 5440,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
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 5441 ENDING AT 8160,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
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 8161 ENDING AT 10880,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
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 10881 ENDING AT 13600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
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 13601 ENDING AT 16320,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
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 16321 ENDING AT 19040,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA8;
CREATE TABLE NEW_ORDERA8
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN NEWA_008
INDEX IN NEWA_008
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 19041 ENDING AT 21760,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
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 2720,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
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 2721 ENDING AT 5440,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
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 5441 ENDING AT 8160,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
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 8161 ENDING AT 10880,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
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 10881 ENDING AT 13600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
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 13601 ENDING AT 16320,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
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 16321 ENDING AT 19040,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB8;
CREATE TABLE NEW_ORDERB8
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN NEWB_008
INDEX IN NEWB_008
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 19041 ENDING AT 21760,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
ALLOW OVERFLOW;
connect reset;

crtb_orders_all.ddl
connect to TPCC in share mode;
DROP TABLE ORDERS1;
CREATE TABLE ORDERS1
(
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_001
INDEX IN ORDI_001
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 1 ENDING AT 2720,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS2;

```

```

CREATE TABLE ORDERS2
(
  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_002
INDEX IN ORDI_002
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 2721 ENDING AT 5440,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS3;
CREATE TABLE ORDERS3
(
  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_003
INDEX IN ORDI_003
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 5441 ENDING AT 8160,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS4;
CREATE TABLE ORDERS4
(
  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_004
INDEX IN ORDI_004
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 8161 ENDING AT 10880,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS5;
CREATE TABLE ORDERS5
(

```

```

  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_005
INDEX IN ORDI_005
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 10881 ENDING AT 13600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS6;
CREATE TABLE ORDERS6
(
  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_006
INDEX IN ORDI_006
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 13601 ENDING AT 16320,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS7;
CREATE TABLE ORDERS7
(
  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_007
INDEX IN ORDI_007
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 16321 ENDING AT 19040,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS8;
CREATE TABLE ORDERS8
(
  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_008
INDEX IN ORDI_008
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 19041 ENDING AT 21760,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

connect reset;

crtb_order_line_all.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 2720,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3675,
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 2721 ENDING AT 5440,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3675,
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 5441 ENDING AT 8160,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3675,
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 8161 ENDING AT 10880,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

```

connect reset;

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINES;
CREATE TABLE ORDER_LINES

```

```

(
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 10881 ENDING AT 13600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  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 13601 ENDING AT 16320,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  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 16321 ENDING AT 19040,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE8;
CREATE TABLE ORDER_LINE8
(
  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_008
INDEX IN OLN_008
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 19041 ENDING AT 21760,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;

crtb_stock_all.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 544
)
)
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 10000,
S_W_ID STARTING FROM 545 ENDING AT 1088
)
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 1089 ENDING AT 1632
)
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 1633 ENDING AT 2176
)

```

```

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 2177 ENDING AT 2720
)
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 2721 ENDING AT 3264
)
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 3265 ENDING AT 3808
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK8;
CREATE TABLE STOCK8
(
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_008
INDEX IN STK_008
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 3809 ENDING AT 4352
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK9;
CREATE TABLE STOCK9
(
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_009
INDEX IN STK_009
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 4353 ENDING AT 4896
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK10;
CREATE TABLE STOCK10
(
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_010
INDEX IN STK_010
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 4897 ENDING AT 5440
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK11;
CREATE TABLE STOCK11
(
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_011
INDEX IN STK_011
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 5441 ENDING AT 5984
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK12;
CREATE TABLE STOCK12
(
  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_012
INDEX IN STK_012
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 5985 ENDING AT 6528
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK13;
CREATE TABLE STOCK13
(
  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_013
INDEX IN STK_013
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 6529 ENDING AT 7072
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE STOCK14;
CREATE TABLE STOCK14
(
  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_014
INDEX IN STK_014
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 7073 ENDING AT 7616
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK15;
CREATE TABLE STOCK15
(
  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_015
INDEX IN STK_015
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 7617 ENDING AT 8160
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK16;
CREATE TABLE STOCK16
(
  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_016
INDEX IN STK_016
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 8161 ENDING AT 8704
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK17;
CREATE TABLE STOCK17
(
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_017
INDEX IN STK_017
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 8705 ENDING AT 9248
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK18;
CREATE TABLE STOCK18
(
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_018
INDEX IN STK_018
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 9249 ENDING AT 9792
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK19;
CREATE TABLE STOCK19
(
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_019
INDEX IN STK_019
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 9793 ENDING AT 10336
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK20;
CREATE TABLE STOCK20
(
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_020
INDEX IN STK_020

```

```

ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 10337 ENDING AT 10880
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK21;
CREATE TABLE STOCK21
(
  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_021
INDEX IN STK_021
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 10881 ENDING AT 11424
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK22;
CREATE TABLE STOCK22
(
  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_022
INDEX IN STK_022
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 11425 ENDING AT 11968
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK23;

```

```

CREATE TABLE STOCK23
(
  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_023
INDEX IN STK_023
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 11969 ENDING AT 12512
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK24;
CREATE TABLE STOCK24
(
  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_024
INDEX IN STK_024
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 12513 ENDING AT 13056
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK25;
CREATE TABLE STOCK25
(
  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_025
INDEX IN STK_025
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 13057 ENDING AT 13600
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK26;
CREATE TABLE STOCK26

```

```

(
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_026
INDEX IN STK_026
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 13601 ENDING AT 14144
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK27;
CREATE TABLE STOCK27

```

```

(
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_027
INDEX IN STK_027
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 14145 ENDING AT 14688
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK28;
CREATE TABLE STOCK28

```

```

(
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_028
INDEX IN STK_028
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 14689 ENDING AT 15232
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK29;
CREATE TABLE STOCK29

```

```

(
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_029
INDEX IN STK_029
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,

```



```

        S_W_ID STARTING FROM 15233 ENDING AT 15776
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK30;
CREATE TABLE STOCK30
(
    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_030
INDEX IN STK_030
ORGANIZE BY KEY SEQUENCE (
    S_I_ID STARTING FROM 1 ENDING AT 100000,
    S_W_ID STARTING FROM 15777 ENDING AT 16320
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK31;
CREATE TABLE STOCK31
(
    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_031
INDEX IN STK_031
ORGANIZE BY KEY SEQUENCE (
    S_I_ID STARTING FROM 1 ENDING AT 100000,
    S_W_ID STARTING FROM 16321 ENDING AT 16864
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK32;
CREATE TABLE STOCK32
(

```

```

    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_032
INDEX IN STK_032
ORGANIZE BY KEY SEQUENCE (
    S_I_ID STARTING FROM 1 ENDING AT 100000,
    S_W_ID STARTING FROM 16865 ENDING AT 17408
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK33;
CREATE TABLE STOCK33
(
    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_033
INDEX IN STK_033
ORGANIZE BY KEY SEQUENCE (
    S_I_ID STARTING FROM 1 ENDING AT 100000,
    S_W_ID STARTING FROM 17409 ENDING AT 17952
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK34;
CREATE TABLE STOCK34
(
    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_034
INDEX IN STK_034
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 17953 ENDING AT 18496
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE STOCK35;

CREATE TABLE STOCK35

```

(
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_035

INDEX IN STK_035

ORGANIZE BY KEY SEQUENCE (

S_I_ID STARTING FROM 1 ENDING AT 10000,

S_W_ID STARTING FROM 18497 ENDING AT 19040

)

ALLOW OVERFLOW;

connect reset;

connect to TPCC in share mode;

DROP TABLE STOCK36;

CREATE TABLE STOCK36

```

(
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_036

INDEX IN STK_036

ORGANIZE BY KEY SEQUENCE (

S_I_ID STARTING FROM 1 ENDING AT 10000,

S_W_ID STARTING FROM 19041 ENDING AT 19584

)

ALLOW OVERFLOW;

connect reset;

connect to TPCC in share mode;

DROP TABLE STOCK37;

CREATE TABLE STOCK37

```

(
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_037

INDEX IN STK_037

ORGANIZE BY KEY SEQUENCE (

S_I_ID STARTING FROM 1 ENDING AT 10000,

S_W_ID STARTING FROM 19585 ENDING AT 20128

)

ALLOW OVERFLOW;

connect reset;

connect to TPCC in share mode;

DROP TABLE STOCK38;

CREATE TABLE STOCK38

```

(
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_038

INDEX IN STK_038

ORGANIZE BY KEY SEQUENCE (

S_I_ID STARTING FROM 1 ENDING AT 10000,

S_W_ID STARTING FROM 20129 ENDING AT 20672

)

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK39;
CREATE TABLE STOCK39
(
  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_039
INDEX IN STK_039
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 20673 ENDING AT 21216
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK40;
CREATE TABLE STOCK40
(
  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_040
INDEX IN STK_040
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 21217 ENDING AT 21760
)
ALLOW OVERFLOW;

```

connect reset;

crtb_warehouse_all.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE1;

```

```

CREATE TABLE WAREHOUSE1
(
  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_001
INDEX IN WAR_001
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 1 ENDING AT 2720
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE2;
CREATE TABLE WAREHOUSE2
(
  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_002
INDEX IN WAR_002
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 2721 ENDING AT 5440
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE3;
CREATE TABLE WAREHOUSE3
(
  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_003
INDEX IN WAR_003
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 5441 ENDING AT 8160
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE4;
CREATE TABLE WAREHOUSE4
(
  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_004
INDEX IN WAR_004
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 8161 ENDING AT 10880
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE5;
CREATE TABLE WAREHOUSE5
(
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_005
INDEX IN WAR_005
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 10881 ENDING AT 13600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE6;
CREATE TABLE WAREHOUSE6
(
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_006
INDEX IN WAR_006
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 13601 ENDING AT 16320
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE7;
CREATE TABLE WAREHOUSE7
(
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_007
INDEX IN WAR_007
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 16321 ENDING AT 19040
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE8;
CREATE TABLE WAREHOUSE8
(
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_008
INDEX IN WAR_008
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 19041 ENDING AT 21760
)
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 UNION ALL
SELECT * FROM CUSTOMER8 UNION ALL
SELECT * FROM CUSTOMER9 UNION ALL
SELECT * FROM CUSTOMER10 UNION ALL

```

```

SELECT * FROM CUSTOMER11 UNION ALL
SELECT * FROM CUSTOMER12 UNION ALL
SELECT * FROM CUSTOMER13 UNION ALL
SELECT * FROM CUSTOMER14 UNION ALL
SELECT * FROM CUSTOMER15 UNION ALL
SELECT * FROM CUSTOMER16 UNION ALL
SELECT * FROM CUSTOMER17 UNION ALL
SELECT * FROM CUSTOMER18 UNION ALL
SELECT * FROM CUSTOMER19 UNION ALL
SELECT * FROM CUSTOMER20 UNION ALL
SELECT * FROM CUSTOMER21 UNION ALL
SELECT * FROM CUSTOMER22 UNION ALL
SELECT * FROM CUSTOMER23 UNION ALL
SELECT * FROM CUSTOMER24 UNION ALL
SELECT * FROM CUSTOMER25 UNION ALL
SELECT * FROM CUSTOMER26 UNION ALL
SELECT * FROM CUSTOMER27 UNION ALL
SELECT * FROM CUSTOMER28 UNION ALL
SELECT * FROM CUSTOMER29 UNION ALL
SELECT * FROM CUSTOMER30 UNION ALL
SELECT * FROM CUSTOMER31 UNION ALL
SELECT * FROM CUSTOMER32 UNION ALL
SELECT * FROM CUSTOMER33 UNION ALL
SELECT * FROM CUSTOMER34 UNION ALL
SELECT * FROM CUSTOMER35 UNION ALL
SELECT * FROM CUSTOMER36 UNION ALL
SELECT * FROM CUSTOMER37 UNION ALL
SELECT * FROM CUSTOMER38 UNION ALL
SELECT * FROM CUSTOMER39 UNION ALL
SELECT * FROM CUSTOMER40
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

crvw_district.ddl

```

connect to TPCC in share mode;
DROP VIEW DISTRICT;
CREATE VIEW DISTRICT
(D_NEXT_O_ID,
D_TAX,
D_YTD,
D_NAME,
D_STREET_1,
D_STREET_2,
D_CITY,
D_STATE,
D_ZIP,
D_ID,
D_W_ID
) AS SELECT * FROM DISTRICT1 UNION ALL
SELECT * FROM DISTRICT2 UNION ALL
SELECT * FROM DISTRICT3 UNION ALL
SELECT * FROM DISTRICT4 UNION ALL
SELECT * FROM DISTRICT5 UNION ALL
SELECT * FROM DISTRICT6 UNION ALL
SELECT * FROM DISTRICT7 UNION ALL
SELECT * FROM DISTRICT8
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

crvw_history.ddl

```

connect to TPCC in share mode;
DROP VIEW HISTORY;

```

```

CREATE VIEW HISTORY
(H_C_ID,
H_C_D_ID,
H_C_W_ID,
H_D_ID,
H_W_ID,
H_DATE,
H_AMOUNT,
H_DATA
) AS SELECT * FROM HISTORY1 UNION ALL
SELECT * FROM HISTORY2 UNION ALL
SELECT * FROM HISTORY3 UNION ALL
SELECT * FROM HISTORY4 UNION ALL
SELECT * FROM HISTORY5 UNION ALL
SELECT * FROM HISTORY6 UNION ALL
SELECT * FROM HISTORY7 UNION ALL
SELECT * FROM HISTORY8
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_ORDERA8 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 UNION ALL
SELECT * FROM NEW_ORDERB8
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 UNION ALL
SELECT * FROM ORDER_LINE8
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

crvw_orders.ddl

```

connect to TPCC in share mode;
DROP VIEW ORDERS;
CREATE VIEW ORDERS
(O_C_ID,
 O_ENTRY_D,
 O_CARRIER_ID,
 O_OL_CNT,
 O_ALL_LOCAL,
 O_ID,
 O_W_ID,
 O_D_ID
) AS SELECT * FROM ORDERS1 UNION ALL
SELECT * FROM ORDERS2 UNION ALL
SELECT * FROM ORDERS3 UNION ALL
SELECT * FROM ORDERS4 UNION ALL
SELECT * FROM ORDERS5 UNION ALL
SELECT * FROM ORDERS6 UNION ALL
SELECT * FROM ORDERS7 UNION ALL
SELECT * FROM ORDERS8
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 UNION ALL
SELECT * FROM STOCK8 UNION ALL
SELECT * FROM STOCK9 UNION ALL

```

```

SELECT * FROM STOCK10 UNION ALL
SELECT * FROM STOCK11 UNION ALL
SELECT * FROM STOCK12 UNION ALL
SELECT * FROM STOCK13 UNION ALL
SELECT * FROM STOCK14 UNION ALL
SELECT * FROM STOCK15 UNION ALL
SELECT * FROM STOCK16 UNION ALL
SELECT * FROM STOCK17 UNION ALL
SELECT * FROM STOCK18 UNION ALL
SELECT * FROM STOCK19 UNION ALL
SELECT * FROM STOCK20 UNION ALL
SELECT * FROM STOCK21 UNION ALL
SELECT * FROM STOCK22 UNION ALL
SELECT * FROM STOCK23 UNION ALL
SELECT * FROM STOCK24 UNION ALL
SELECT * FROM STOCK25 UNION ALL
SELECT * FROM STOCK26 UNION ALL
SELECT * FROM STOCK27 UNION ALL
SELECT * FROM STOCK28 UNION ALL
SELECT * FROM STOCK29 UNION ALL
SELECT * FROM STOCK30 UNION ALL
SELECT * FROM STOCK31 UNION ALL
SELECT * FROM STOCK32 UNION ALL
SELECT * FROM STOCK33 UNION ALL
SELECT * FROM STOCK34 UNION ALL
SELECT * FROM STOCK35 UNION ALL
SELECT * FROM STOCK36 UNION ALL
SELECT * FROM STOCK37 UNION ALL
SELECT * FROM STOCK38 UNION ALL
SELECT * FROM STOCK39 UNION ALL
SELECT * FROM STOCK40
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

crvw_warehouse.ddl

```

connect to TPCC in share mode;
DROP VIEW WAREHOUSE;
CREATE VIEW WAREHOUSE
(W_NAME,
 W_STREET_1,
 W_STREET_2,
 W_CITY,
 W_STATE,
 W_ZIP,
 W_TAX,
 W_YTD,
 W_ID
) AS SELECT * FROM WAREHOUSE1 UNION ALL
SELECT * FROM WAREHOUSE2 UNION ALL
SELECT * FROM WAREHOUSE3 UNION ALL
SELECT * FROM WAREHOUSE4 UNION ALL
SELECT * FROM WAREHOUSE5 UNION ALL
SELECT * FROM WAREHOUSE6 UNION ALL
SELECT * FROM WAREHOUSE7 UNION ALL
SELECT * FROM WAREHOUSE8
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

gen_customer_all.bat

```

C:\tpc-c.ibm\dbgen\gendata.exe -t -r 1 544 -f1
C:\flats\flat_001\customer_001_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t -r 545 1088 -f1
C:\flats\flat_002\customer_002_1.dat

```

```

C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 1089 1632 -f1
C:\flats\flat_003\customer_003_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 1633 2176 -f1
C:\flats\flat_004\customer_004_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 2177 2720 -f1
C:\flats\flat_005\customer_005_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 2721 3264 -f1
C:\flats\flat_006\customer_006_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 3265 3808 -f1
C:\flats\flat_007\customer_007_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 3809 4352 -f1
C:\flats\flat_008\customer_008_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 4353 4896 -f1
C:\flats\flat_009\customer_009_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 4897 5440 -f1
C:\flats\flat_010\customer_010_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 5441 5984 -f1
C:\flats\flat_011\customer_011_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 5985 6528 -f1
C:\flats\flat_012\customer_012_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 6529 7072 -f1
C:\flats\flat_013\customer_013_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 7073 7616 -f1
C:\flats\flat_014\customer_014_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 7617 8160 -f1
C:\flats\flat_015\customer_015_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 8161 8704 -f1
C:\flats\flat_016\customer_016_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 8705 9248 -f1
C:\flats\flat_017\customer_017_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 9249 9792 -f1
C:\flats\flat_018\customer_018_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 9793 10336 -f1
C:\flats\flat_019\customer_019_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 10337 10880 -f1
C:\flats\flat_020\customer_020_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 10881 11424 -f1
C:\flats\flat_021\customer_021_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 11425 11968 -f1
C:\flats\flat_022\customer_022_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 11969 12512 -f1
C:\flats\flat_023\customer_023_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 12513 13056 -f1
C:\flats\flat_024\customer_024_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 13057 13600 -f1
C:\flats\flat_025\customer_025_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 13601 14144 -f1
C:\flats\flat_026\customer_026_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 14145 14688 -f1
C:\flats\flat_027\customer_027_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 14689 15232 -f1
C:\flats\flat_028\customer_028_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 15233 15776 -f1
C:\flats\flat_029\customer_029_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 15777 16320 -f1
C:\flats\flat_030\customer_030_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 16321 16864 -f1
C:\flats\flat_031\customer_031_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 16865 17408 -f1
C:\flats\flat_032\customer_032_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 17409 17952 -f1
C:\flats\flat_033\customer_033_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 17953 18496 -f1
C:\flats\flat_034\customer_034_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 18497 19040 -f1
C:\flats\flat_035\customer_035_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 19041 19584 -f1
C:\flats\flat_036\customer_036_1.dat

```

```

C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 19585 20128 -f1
C:\flats\flat_037\customer_037_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 20129 20672 -f1
C:\flats\flat_038\customer_038_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 20673 21216 -f1
C:\flats\flat_039\customer_039_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 7 -r 21217 21760 -f1
C:\flats\flat_040\customer_040_1.dat

```

gen_district_all.bat

```

C:\tpc-c.ibm\dbgen\gendata.exe -t 4 -r 1 2720 -f1
C:\flats\flat_001\district_001_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 4 -r 2721 5440 -f1
C:\flats\flat_002\district_002_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 4 -r 5441 8160 -f1
C:\flats\flat_003\district_003_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 4 -r 8161 10880 -f1
C:\flats\flat_004\district_004_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 4 -r 10881 13600 -f1
C:\flats\flat_005\district_005_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 4 -r 13601 16320 -f1
C:\flats\flat_006\district_006_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 4 -r 16321 19040 -f1
C:\flats\flat_007\district_007_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 4 -r 19041 21760 -f1
C:\flats\flat_008\district_008_1.dat

```

gen_history_all.bat

```

C:\tpc-c.ibm\dbgen\gendata.exe -t 8 -r 1 2720 -f1
C:\flats\flat_001\history_001_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 8 -r 2721 5440 -f1
C:\flats\flat_002\history_002_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 8 -r 5441 8160 -f1
C:\flats\flat_003\history_003_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 8 -r 8161 10880 -f1
C:\flats\flat_004\history_004_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 8 -r 10881 13600 -f1
C:\flats\flat_005\history_005_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 8 -r 13601 16320 -f1
C:\flats\flat_006\history_006_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 8 -r 16321 19040 -f1
C:\flats\flat_007\history_007_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 8 -r 19041 21760 -f1
C:\flats\flat_008\history_008_1.dat

```

gen_item_1.bat

```

C:\tpc-c.ibm\dbgen\gendata.exe -t 5 -f1 C:\flats\flat\item_1.dat

```

gen_new_order_all.bat

```

C:\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 1 2720 -f1
C:\flats\flat_001\neworder_001_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 2721 5440 -f1
C:\flats\flat_002\neworder_002_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 5441 8160 -f1
C:\flats\flat_003\neworder_003_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 8161 10880 -f1
C:\flats\flat_004\neworder_004_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 10881 13600 -f1
C:\flats\flat_005\neworder_005_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 13601 16320 -f1
C:\flats\flat_006\neworder_006_1.dat

```

```
C:\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 16321 19040 -f1
C:\flats\flat_007\neworder_007_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 11 -r 19041 21760 -f1
C:\flats\flat_008\neworder_008_1.dat
```

gen_orders_all.bat

```
C:\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 1 2720 -f1
C:\flats\flat_001\orders_001_1.dat -f2 C:\flats\flat_001\orderline_001_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 2721 5440 -f1
C:\flats\flat_002\orders_002_1.dat -f2 C:\flats\flat_002\orderline_002_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 5441 8160 -f1
C:\flats\flat_003\orders_003_1.dat -f2 C:\flats\flat_003\orderline_003_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 8161 10880 -f1
C:\flats\flat_004\orders_004_1.dat -f2 C:\flats\flat_004\orderline_004_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 10881 13600 -f1
C:\flats\flat_005\orders_005_1.dat -f2 C:\flats\flat_005\orderline_005_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 13601 16320 -f1
C:\flats\flat_006\orders_006_1.dat -f2 C:\flats\flat_006\orderline_006_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 16321 19040 -f1
C:\flats\flat_007\orders_007_1.dat -f2 C:\flats\flat_007\orderline_007_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 9 -r 19041 21760 -f1
C:\flats\flat_008\orders_008_1.dat -f2 C:\flats\flat_008\orderline_008_1.dat
```

gen_stock_all.bat

```
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 1 544 -f1
C:\flats\flat_001\stock_001_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 545 1088 -f1
C:\flats\flat_002\stock_002_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 1089 1632 -f1
C:\flats\flat_003\stock_003_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 1633 2176 -f1
C:\flats\flat_004\stock_004_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 2177 2720 -f1
C:\flats\flat_005\stock_005_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 2721 3264 -f1
C:\flats\flat_006\stock_006_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 3265 3808 -f1
C:\flats\flat_007\stock_007_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 3809 4352 -f1
C:\flats\flat_008\stock_008_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 4353 4896 -f1
C:\flats\flat_009\stock_009_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 4897 5440 -f1
C:\flats\flat_010\stock_010_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 5441 5984 -f1
C:\flats\flat_011\stock_011_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 5985 6528 -f1
C:\flats\flat_012\stock_012_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 6529 7072 -f1
C:\flats\flat_013\stock_013_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 7073 7616 -f1
C:\flats\flat_014\stock_014_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 7617 8160 -f1
C:\flats\flat_015\stock_015_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 8161 8704 -f1
C:\flats\flat_016\stock_016_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 8705 9248 -f1
C:\flats\flat_017\stock_017_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 9249 9792 -f1
C:\flats\flat_018\stock_018_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 9793 10336 -f1
C:\flats\flat_019\stock_019_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 10337 10880 -f1
C:\flats\flat_020\stock_020_1.dat
```

```
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 10881 11424 -f1
C:\flats\flat_021\stock_021_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 11425 11968 -f1
C:\flats\flat_022\stock_022_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 11969 12512 -f1
C:\flats\flat_023\stock_023_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 12513 13056 -f1
C:\flats\flat_024\stock_024_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 13057 13600 -f1
C:\flats\flat_025\stock_025_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 13601 14144 -f1
C:\flats\flat_026\stock_026_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 14145 14688 -f1
C:\flats\flat_027\stock_027_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 14689 15232 -f1
C:\flats\flat_028\stock_028_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 15233 15776 -f1
C:\flats\flat_029\stock_029_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 15777 16320 -f1
C:\flats\flat_030\stock_030_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 16321 16864 -f1
C:\flats\flat_031\stock_031_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 16865 17408 -f1
C:\flats\flat_032\stock_032_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 17409 17952 -f1
C:\flats\flat_033\stock_033_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 17953 18496 -f1
C:\flats\flat_034\stock_034_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 18497 19040 -f1
C:\flats\flat_035\stock_035_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 19041 19584 -f1
C:\flats\flat_036\stock_036_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 19585 20128 -f1
C:\flats\flat_037\stock_037_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 20129 20672 -f1
C:\flats\flat_038\stock_038_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 20673 21216 -f1
C:\flats\flat_039\stock_039_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 6 -r 21217 21760 -f1
C:\flats\flat_040\stock_040_1.dat
```

gen_warehouse_all.bat

```
C:\tpc-c.ibm\dbgen\gendata.exe -t 3 -r 1 2720 -f1
C:\flats\flat_001\warehouse_001_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 3 -r 2721 5440 -f1
C:\flats\flat_002\warehouse_002_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 3 -r 5441 8160 -f1
C:\flats\flat_003\warehouse_003_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 3 -r 8161 10880 -f1
C:\flats\flat_004\warehouse_004_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 3 -r 10881 13600 -f1
C:\flats\flat_005\warehouse_005_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 3 -r 13601 16320 -f1
C:\flats\flat_006\warehouse_006_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 3 -r 16321 19040 -f1
C:\flats\flat_007\warehouse_007_1.dat
C:\tpc-c.ibm\dbgen\gendata.exe -t 3 -r 19041 21760 -f1
C:\flats\flat_008\warehouse_008_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
16320000 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
16320000 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
16320000 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
16320000 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
16320000 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
16320000 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
16320000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_008\customer_008_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_009\customer_009_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER9;
COMMIT WORK;

```

```

CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_010\customer_010_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_011\customer_011_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_012\customer_012_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_013\customer_013_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_014\customer_014_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_015\customer_015_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_016\customer_016_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_017\customer_017_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER18 ACTIVATE NOT LOGGED INITIALLY;

```



```

IMPORT FROM C:\flats\flat_035\customer_035_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_036\customer_036_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_037\customer_037_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_038\customer_038_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_039\customer_039_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER40 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_040\customer_040_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
16320000 INSERT INTO CUSTOMER40;
COMMIT WORK;
CONNECT RESET;

```

load_district_all.ddl

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_001\district_001_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_002\district_002_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_003\district_003_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;

```

```

IMPORT FROM C:\flats\flat_004\district_004_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_005\district_005_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_006\district_006_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_007\district_007_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_008\district_008_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT8;
COMMIT WORK;
CONNECT RESET;

```

load_history_all.ddl

```

connect to TPCC in share mode;
LOAD FROM C:\flats\flat_001\history_001_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY1
NONRECOVERABLE DATA BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM C:\flats\flat_002\history_002_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY2
NONRECOVERABLE DATA BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM C:\flats\flat_003\history_003_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY3
NONRECOVERABLE DATA BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM C:\flats\flat_004\history_004_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY4
NONRECOVERABLE DATA BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM C:\flats\flat_005\history_005_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY5
NONRECOVERABLE DATA BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM C:\flats\flat_006\history_006_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY6
NONRECOVERABLE DATA BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM C:\flats\flat_007\history_007_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY7
NONRECOVERABLE DATA BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;

```

```
LOAD FROM C:\flats\flat_008\history_008_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY8
NONRECOVERABLE DATA BUFFER 5000 CPU_PARALLELISM 4 ;
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
897600000 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
897600000 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
897600000 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
897600000 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
897600000 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
897600000 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
897600000 INSERT INTO ORDER_LINE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE8 ACTIVATE NOT LOGGED INITIALLY;
```

```
IMPORT FROM C:\flats\flat_008\orderline_008_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
897600000 INSERT INTO ORDER_LINE8;
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 1000 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 1000 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 1000 INSERT INTO NEW_ORDERA3;
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 1000 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 1000 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 1000 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 1000 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 1000 INSERT INTO NEW_ORDERA7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_008\neworder_008_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO NEW_ORDERA8;
COMMIT WORK;
CONNECT RESET;
```

load_item_1.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_orders_all.ddl

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_001\orders_001_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_002\orders_002_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_003\orders_003_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_004\orders_004_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_005\orders_005_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_006\orders_006_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_007\orders_007_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_008\orders_008_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO ORDERS8;
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 54400000
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 54400000
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 54400000
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 54400000
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 54400000
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 54400000
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 54400000
INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_008\stock_008_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_009\stock_009_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK10 ACTIVATE NOT LOGGED INITIALLY;

```



```

IMPORT FROM C:\flats\flat_027\stock_027_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_028\stock_028_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_029\stock_029_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_030\stock_030_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_031\stock_031_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_032\stock_032_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_033\stock_033_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_034\stock_034_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_035\stock_035_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK35;
COMMIT WORK;

```

```

CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_036\stock_036_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_037\stock_037_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_038\stock_038_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_039\stock_039_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK40 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM C:\flats\flat_040\stock_040_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 54400000
INSERT INTO STOCK40;
COMMIT WORK;
CONNECT RESET;

```

load_warehouse_all.ddl

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_001\warehouse_001_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_002\warehouse_002_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_003\warehouse_003_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_004\warehouse_004_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE4;
COMMIT WORK;
CONNECT RESET;

```

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_005\warehouse_005_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_006\warehouse_006_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_007\warehouse_007_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM C:\flats\flat_008\warehouse_008_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE8;
COMMIT WORK;
CONNECT RESET;

```

rnst_customer_all.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;

```

```

RUNSTATS ON TABLE TPCC.CUSTOMER11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;

```



```

RUNSTATS ON TABLE TPCC.CUSTOMER28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER40 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

rnst_district_all.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;

```

```

RUNSTATS ON TABLE TPCC.DISTRICT4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT8 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

rnst_history_all.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY8 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

rnst_item.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ITEM AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

rnst_new_order_all.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA1 AND INDEXES ALL;

```

```

COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB8 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

rnst_order_line_all.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE1 AND INDEXES ALL;
COMMIT WORK;

```

```

connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE8 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

rnst_orders_all.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS8 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

rnst_stock_all.ddl


```

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK40 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

rnst_warehouse_all.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE8 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

dbgen\gendata.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.
*****
*****/

/*
 * gendata.c - Generate data for TPC-C database
 */

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <sqlutil.h>
/* NT named pipe support */
#include <windows.h>
#include <time.h>

#include "platform.h"
#include "db2tpcc.h"
#include "tpccrnd.h"
#include "tpccmisc.h"
#include "lval.h"

/* PROTOTYPES. */
void gen_dist_tbl( void );
void gen_cust_tbl( void );
void gen_hist_tbl( void );
void gen_nu_ord_tbl( void );
void gen_ordr_tbl( void );
void gen_item_tbl( void );
void gen_stock_tbl( void );
void gen_ware_tbl( void );

int i, j;
double timestamp1, timestamp2, elapse;
int rc, rc1, rc2;

int using_range = 0;
int using_npipe = 0;
int using_rctload = 0;
int quiet_mode = 0;
sqlint32 ware_start=-1, ware_end=-1;

char fmtWare[] = "%s|%s|%s|%s|%s|%s|%s|%d|I64d|\n";
char fmtDist[] = "%d|%d|I64d|%s|%s|%s|%s|%s|%s|%d|\n";
char fmtItem[] = "%s|%d|%s|%d|\n";
char fmtStock[] =
"%d|%d|%d|%d|%s|%s|%s|%s|%s|%s|%s|%s|%s|%d|\n";
char fmtCust[] =
"%d|%s|%s|%s|I64d|I64d|%s|%s|%d|%s|%s|%s|%s|%s|%d|%d|I64d|
I64d|\n";
char fmtHist[] = "%d|%d|%d|%d|%d|I64d|\n";
char fmtOrdr[] = "%d|I64d|%d|%d|%d|%d|\n";
char fmtOLine[] = "%I64d|%d|%d|%d|%s|%d|%d|\n";
char fmtNewOrd[] = "%d|%d|\n";
void InitFormatStrings(char delim);
void ScalingReport(void);

int outtype1 = 0;
int outtype2 = 0;
char *outname1 = NULL;
char *outname2 = NULL;

```

```

/*-----*/
/* main */
/*-----*/
int main (int argc, char *argv[])
{
    int option = -1;
    char *delim = NULL;

    /*
    ****
    * *
    /* Compute Warehouse Ranges */
    /*
    ****
    * *
    ware_start = 1;
    ware_end = WAREHOUSES;

    /*
    ****
    * *
    /* Process Command Line Arguments */
    /*
    ****
    * *

    /* Valid Command Line Options
    *-----
    * Table Option:      -t <table>      (-t 3 for warehouse)
    * Output Column Delimiter: -d <char>  (-d ' ', -d '|', etc)
    * Output to File:    -f[n] <file>    (-f customer.dat)
    * Output to Pipe:    -p[n] <pipe>     (-p tpcpipe.000)
    * Warehouse Range:   -r <start> <end> (-r 1 100)
    * Scaling Report:    -s
    * Quiet Mode:        -q
    *
    * The -f[n] and/or -p[n] options are required.
    * The -t, -d, -r, -s and -q options are optional.
    *
    * If -d is omitted, the vertical bar (pipe) symbol (|) will be used.
    * If -r is omitted, the range [1..WAREHOUSES] will be used.
    *
    * Due to the TPC-C spec requiring that orders and orderline be
    * generated at the same time, there is an extension to the -f and -p
    * options to specify one of the two output streams for each argument.
    *
    * -f1 orders.dat -f2 orderline.dat will output to two files
    * -f1 orders.dat -p2 tpcpipe.000 will output to a file and a pipe
    *
    * -f1/-p1 specifies the destination for the orders table
    * -f2/-p2 specifies the destination for the orderline table
    *
    */

    /* Read Arguments */
    for (i=1; i<argc; i++)
    {
        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], "-f1") == 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();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {
        fprintf(stdout, "\nITEM table generated in %8.2f seconds.\n\n", elapse);
        fflush(stdout);
    } else {
        fprintf(stderr, "\nITEM table FAILED at (1 %d) after %8.2f
seconds.\n\n", item_num, elapse);
        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 = 3000000;

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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nWAREHOUSE table generated in %8.2f
seconds.\n\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nWAREHOUSE table FAILED at (W %d) after %8.2f
seconds.\n\n", ware_num, elapsed);
    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\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 currtmstp;
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");
currtmstp = 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-rtload) */
            if (using_rtload == 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,
    currtmstp,
    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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nCUSTOMER table generated in %8.2f
seconds.\n\n", elapsed);
    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, elapsed);
    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 currtmstp;

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

```

```

timestamp1 = current_time();

rc = GenericOpen(&hnd, outtype1, outname1);
if (rc != 0) { goto hist_done; }

currtmstp = 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,
                                currtmstp,
                                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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nHISTORY table generated in %8.2f seconds.\n\n", elapsed);
    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, elapsed);
    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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nNEW_ORDER table generated in %8.2f
seconds.\n\n", elapsed);
    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, elapsed);
    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_carrier_id;
sqlint32 ord_ol_cnt;
sqlint32 oline_ol_num;
sqlint32 oline_item_num;

sqlint32 oline_amount;
char oline_dist_info[25];
sqlint64 nulltmstamp = 0;
sqlint64 currtmstamp;

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; }

currtmstamp = 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_carrier_id = rand_integer( 1, 10 ) ;
else
ord_carrier_id = 0;

cust_num = random_1_3000();
ord_ol_cnt =
rand_integer(MIN_OL_PER_ORDER,MAX_OL_PER_ORDER);

numBytes = sprintf(Buffer, fmtOrdr,
cust_num,
currtmstamp,
ord_carrier_id,
ord_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_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) ? currtmstamp : nulltmstamp),
((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)

LDFLAGS = $(LDFLAGS_EXEC) $(LDFLAGS_LIB)

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

```

```

OBJS =          tpcrnd$(OBJEXT) \
$(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT)
OBJ_EEE =       $(TPCC_ROOT)/Src.Common/tpccclwh$(OBJEXT)

EXEC =          gendata$(BINEXT)

#
#####
#####
# End-User Targets
#
#####
#####

all:            $(EXEC)

clean:          - $(ERASE) *$(OBJEXT) $(EXEC)

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

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

$(EXEC):
$(LDFLAGS_OUT)$@

#
#####
#####
# 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\tpcrnd.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[] =

"0123456789abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ
WXYZ";

static char *last_name_parts[] =
{
  "BAR",
  "OUGHT",
  "ABLE",
  "PRI",
  "PRES",
  "ESE",
  "ANTI",
  "CALLY",
  "ATION",
  "EING"
};

/*
*****
*****
 * 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);
}

/*
*****
*****
 * 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);
}

/*
*****
*****
* NUrand_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);
}

/*
*****
*****
* 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 21760
#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 */
/*
*****
*****/
#ifndef INVALID_HANDLE_VALUE
#define INVALID_HANDLE_VALUE ((DWORD)-1)
#endif

#ifndef 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/Winx64 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 /MD -GS- -DWIN64 -J -Zp8
-DREG_KIT_METHOD
CFLAGS_OUT=/Fo
CFLAGS_DEBUG=

# Linker Configuration (MSVC)

```

```
LD_EXEC=link.exe
LD_STORP=link.exe
LD_FLAGS_EXEC=
LD_FLAGS_SHLIB=/DLL
LD_FLAGS_STORP=$(LD_FLAGS_SHLIB) /DEF:rpctpc.def
LD_FLAGS_LIB=/LIBPATH:$(TPCC_SQLLIB)\lib
/LIBPATH:"C:\MsSDKx64\lib\amd64" db2api.lib winmm.lib
LD_FLAGS_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)
UTIL_OBJ_DB2 = tpcctx$(OBJEXT)
```

```
#
#####
# User Targets
#
#####
```

```
all: dbgen connect $(UTIL_OBJ_DB2) disconnect
```

```
dbgen: $(UTIL_OBJ)
```

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

```
#
#####
# Helper Targets
#
#####
```

```
connect:
       - db2 connect to $(TPCC_DBNAME)
```

```
disconnect:
       - db2 connect reset
       - db2 terminate
```

```
rebind: connect
        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$(OBJEXT): tpccdbg.c
tpccctx$(OBJEXT): tpccctx.c
tpccmisc$(OBJEXT): tpccmisc.c

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

dbgen\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 - 2005
** 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 <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=CK050901

@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=%HOME%\tpc-c.ibm
```

```
set TPCC_SQLLIB=%HOME%\sqllib
```

```
set TPCC_RUNDATA=%HOME%\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) = 4.329822e-007

Max number of concurrently active databases (NUMDB) = 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) = 660

Agent pool size (NUM_POOLAGENTS) = 0

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) =
 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
 Number of FCM connection entries (FCM_NUM_CONNECT) =
 AUTOMATIC
 Number of FCM message anchors (FCM_NUM_ANCHORS) =
 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) = MAC0
 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) = 5000
 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) = 10000
 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) = 660
 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) = 120
 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 = S0000167.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) = 1817
 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

DB2_RESOURCE_POLICY=c:\tpc-c.ibm\cfg\aff8.cfg
 DB2_SELUDI_COMM_BUFFER=Y
 DB2_USE_ALTERNATE_PAGE_CLEANING=YES
 DB2_MAX_NON_TABLE_LOCKS=500
 DB2_LGPAGE_BP=ON
 DB2_TRUSTED_BINDIN=ON
 DB2_KEEPTABLELOCK=ON
 DB2_NO_FORK_CHECK=ON
 DB2_FMP_COMM_HEAPSZ=0
 DB2_APM_PERFORMANCE=ALL
 DB2_ENABLE_BUFDP=OFF
 DB2_PINNED_BP=ON
 DB2_SELECTIVITY=ON
 DB2ASSUMEUPDATE=ON
 DB2CHECKCLIENTINTERVAL=0
 DB2_HASH_JOIN=OFF
 DB2CHKSQLDA=OFF
 DB2_COLLECT_TS_REC_INFO=false
 DB2COMM=tcPIP
 DB2CHKPTR=OFF

Aff8.cfg

```
<RESOURCE_POLICY>
<DATABASE_RESOURCE_POLICY>
<DBNAME>TPCC</DBNAME>
<METHOD>CPUMASK</METHOD>
<RESOURCE_BINDING>
<RESOURCE>3</RESOURCE>
<DBMEM_PERCENTAGE>0.0</DBMEM_PERCENTAGE>
<SERVICE_NAME>50001</SERVICE_NAME>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>12</RESOURCE>
<DBMEM_PERCENTAGE>0.0</DBMEM_PERCENTAGE>
<SERVICE_NAME>50002</SERVICE_NAME>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>48</RESOURCE>
<DBMEM_PERCENTAGE>0.0</DBMEM_PERCENTAGE>
<SERVICE_NAME>50003</SERVICE_NAME>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>192</RESOURCE>
<DBMEM_PERCENTAGE>0.0</DBMEM_PERCENTAGE>
<SERVICE_NAME>50004</SERVICE_NAME>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>768</RESOURCE>
<DBMEM_PERCENTAGE>0.0</DBMEM_PERCENTAGE>
<SERVICE_NAME>50005</SERVICE_NAME>
```

```
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>3072</RESOURCE>
<DBMEM_PERCENTAGE>0.0</DBMEM_PERCENTAGE>
<SERVICE_NAME>50006</SERVICE_NAME>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>12288</RESOURCE>
<DBMEM_PERCENTAGE>0.0</DBMEM_PERCENTAGE>
<SERVICE_NAME>50007</SERVICE_NAME>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>49152</RESOURCE>
<DBMEM_PERCENTAGE>0.0</DBMEM_PERCENTAGE>
<SERVICE_NAME>50008</SERVICE_NAME>
</RESOURCE_BINDING>
</DATABASE_RESOURCE_POLICY>
</RESOURCE_POLICY>
```

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
- Windows Time
- Wireless Configuration

System Information Report

System Information report written at: 02/17/06 23:16:02

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 Build 3790
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	DB2SERV1
System Manufacturer	IBM
System Model	IBM x3950-[88724RZ]-
System Type	x64-based PC
Processor	EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
Processor	EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
Processor	EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
Processor	EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
Processor	EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
Processor	EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
Processor	EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
Processor	EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
Processor	EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz

Processor EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
 Processor EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
 Processor EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
 Processor EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
 Processor EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
 Processor EM64T Family 15 Model 4 Stepping 8 GenuineIntel ~3003 Mhz
 BIOS Version/Date IBM -[ZUE148AUS-1.09]-, 1/12/2006
 SMBIOS Version 2.3
 Windows Directory C:\WINDOWS
 System Directory C:\WINDOWS\system32
 Boot Device \Device\HarddiskVolume242
 Locale United States
 Hardware Abstraction Layer Version = "5.2.3790.1830
 (srv03_sp1_rtm.050324-1447)"
 User NameDB2SERV1\TPCC
 Time ZoneEastern Standard Time
 Total Physical Memory 130,558.83 MB
 Available Physical Memory 124.06 GB
 Total Virtual Memory 126.62 GB
 Available Virtual Memory 126.06 GB
 Page File Space 2.00 GB
 Page File C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource Device
 Memory Address 0xF8900000-0xF89FFFFF PCI bus
 Memory Address 0xF8900000-0xF89FFFFF Broadcom NetXtreme Gigabit Ethernet #2

I/O Port 0x00000000-0x00001FFF PCI bus
 I/O Port 0x00000000-0x00001FFF Direct memory access controller

Memory Address 0xF0000000-0xF7FFFFFF PCI bus
 Memory Address 0xF0000000-0xF7FFFFFF Radeon 7000 / RADEON VE Family (Microsoft Corporation)

I/O Port 0x00004400-0x000045FF PCI bus
 I/O Port 0x00004400-0x000045FF QLogic Fibre Channel Adapter

I/O Port 0x00002600-0x000027FF PCI bus
 I/O Port 0x00002600-0x000027FF QLogic Fibre Channel Adapter

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 0xF8800000-0xF88FFFFF PCI bus
 Memory Address 0xF8800000-0xF88FFFFF Radeon 7000 / RADEON VE Family (Microsoft Corporation)

I/O Port 0x00002200-0x000023FF PCI bus
 I/O Port 0x00002200-0x000023FF QLogic Fibre Channel Adapter

I/O Port 0x00002000-0x000021FF PCI bus
 I/O Port 0x00002000-0x000021FF QLogic Fibre Channel Adapter

I/O Port 0x00004600-0x000047FF PCI bus
 I/O Port 0x00004600-0x000047FF QLogic Fibre Channel Adapter

Memory Address 0xF8E00000-0xF8EFFFFF PCI bus
 Memory Address 0xF8E00000-0xF8EFFFFF PCI standard
 PCI-to-PCI bridge

I/O Port 0x00004200-0x000043FF PCI bus
 I/O Port 0x00004200-0x000043FF QLogic Fibre Channel Adapter

Memory Address 0xEF000000-0xEF0FFFFF PCI bus
 Memory Address 0xEF000000-0xEF0FFFFF NEC PCI to USB Open Host Controller

I/O Port 0x00002400-0x000025FF PCI bus
 I/O Port 0x00002400-0x000025FF QLogic Fibre Channel Adapter

I/O Port 0x00004800-0x000049FF PCI bus
 I/O Port 0x00004800-0x000049FF QLogic Fibre Channel Adapter

Memory Address 0xEF100000-0xEF1FFFFF PCI bus
 Memory Address 0xEF100000-0xEF1FFFFF Broadcom NetXtreme Gigabit Ethernet #3

Memory Address 0xA0000-0xBFFFF PCI bus
 Memory Address 0xA0000-0xBFFFF Radeon 7000 / RADEON VE Family (Microsoft Corporation)

IRQ 92 NEC PCI to USB Open Host Controller
 IRQ 92 NEC PCI to USB Open Host Controller
 IRQ 92 Standard Enhanced PCI to USB Host Controller

[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 PS/2 Keyboard OK
 0x00000064-0x00000064 Standard 101/102-Key or Microsoft Natural PS/2 Keyboard OK
 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
0x00002400-0x000025FF	PCI bus	OK
0x00002400-0x000025FF	QLogic Fibre Channel Adapter	OK
0x00002600-0x000027FF	PCI bus	OK
0x00002600-0x000027FF	QLogic Fibre Channel Adapter	OK
0x00003000-0x00003FFF	PCI bus	OK
0x00004000-0x000041FF	PCI bus	OK
0x00004200-0x000043FF	PCI bus	OK
0x00004200-0x000043FF	QLogic Fibre Channel Adapter	OK
0x00004400-0x000045FF	PCI bus	OK
0x00004400-0x000045FF	QLogic Fibre Channel Adapter	OK
0x00004600-0x000047FF	PCI bus	OK
0x00004600-0x000047FF	QLogic Fibre Channel Adapter	OK
0x00004800-0x000049FF	PCI bus	OK
0x00004800-0x000049FF	QLogic Fibre Channel Adapter	OK
0x00005000-0x00005FFF	PCI bus	OK
0x00006000-0x00006FFF	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 53	QLogic Fibre Channel Adapter	OK
IRQ 55	IBM ServeRAID 6M Controller	OK
IRQ 92	NEC PCI to USB Open Host Controller	OK
IRQ 92	NEC PCI to USB Open Host Controller	OK
IRQ 92	Standard Enhanced PCI to USB Host Controller	OK
IRQ 96	Broadcom NetXtreme Gigabit Ethernet #3	OK
IRQ 100	Broadcom NetXtreme Gigabit Ethernet #4	OK
IRQ 90	QLogic Fibre Channel Adapter	OK
IRQ 91	QLogic Fibre Channel Adapter	OK
IRQ 124	QLogic Fibre Channel Adapter	OK

IRQ 125 QLogic Fibre Channel Adapter OK

[Memory]

Resource	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	Radeon 7000 / RADEON VE Family (Microsoft Corporation)	OK
0xF0000000-0xF7FFFFFF	PCI bus	OK
0xF0000000-0xF7FFFFFF	Radeon 7000 / RADEON VE Family (Microsoft Corporation)	OK
0xF8800000-0xF88FFFFFF	PCI bus	OK
0xF8800000-0xF88FFFFFF	Radeon 7000 / RADEON VE Family (Microsoft Corporation)	OK
0xF8810000-0xF8810FFF	NEC PCI to USB Open Host Controller	OK
0xF8811000-0xF8811FFF	NEC PCI to USB Open Host Controller	OK
0xF8812000-0xF88120FF	Standard Enhanced PCI to USB Host Controller	OK
0x0400-0x04FF	System board	OK
0x100000-0x7FFFFFFF	Memory Module	OK
0xF8900000-0xF89FFFFFF	PCI bus	OK
0xF8900000-0xF89FFFFFF	Broadcom NetXtreme Gigabit Ethernet #2	OK
0xF8910000-0xF891FFFF	Broadcom NetXtreme Gigabit Ethernet	OK
0xF8A00000-0xF8AFFFFF	PCI bus	OK
0xF8A20000-0xF8A20FFF	QLogic Fibre Channel Adapter	OK
0xF8B00000-0xF8BFFFFF	PCI bus	OK
0xF8B20000-0xF8B20FFF	QLogic Fibre Channel Adapter	OK
0xF8C00000-0xF8CFFFFF	PCI bus	OK
0xF8C20000-0xF8C20FFF	QLogic Fibre Channel Adapter	OK
0xF8D00000-0xF8DFFFFF	PCI bus	OK
0xF8D20000-0xF8D20FFF	QLogic Fibre Channel Adapter	OK
0xF8000000-0xF83FFFFFF	PCI bus	OK
0xF8400000-0xF87FFFFFF	PCI bus	OK
0xF8E00000-0xF8EFFFFFF	PCI bus	OK
0xF8E00000-0xF8EFFFFFF	PCI standard PCI-to-PCI bridge	OK
0xF8E80000-0xF8E80FFF	IBM ServeRAID 6M Controller	OK
0xEF000000-0xEF0FFFFFF	PCI bus	OK
0xEF000000-0xEF0FFFFFF	NEC PCI to USB Open Host Controller	OK
0xEF001000-0xEF001FFF	NEC PCI to USB Open Host Controller	OK
0xEF002000-0xEF0020FF	Standard Enhanced PCI to USB Host Controller	OK
0xEF100000-0xEF1FFFFFF	PCI bus	OK
0xEF100000-0xEF1FFFFFF	Broadcom NetXtreme Gigabit Ethernet #3	OK
0xEF110000-0xEF11FFFF	Broadcom NetXtreme Gigabit Ethernet #4	OK
0xEF200000-0xEF2FFFFFF	PCI bus	OK
0xEF220000-0xEF220FFF	QLogic Fibre Channel Adapter	OK
0xEF300000-0xEF3FFFFFF	PCI bus	OK
0xEF320000-0xEF320FFF	QLogic Fibre Channel Adapter	OK
0xEF400000-0xEF4FFFFFF	PCI bus	OK
0xEF420000-0xEF420FFF	QLogic Fibre Channel Adapter	OK
0xEF500000-0xEF5FFFFFF	PCI bus	OK
0xEF520000-0xEF520FFF	QLogic Fibre Channel Adapter	OK
0xEE800000-0xEEBFFFFFF	PCI bus	OK
0xEF800000-0xEF8FFFFFF	PCI bus	OK
0xEEC00000-0xEEFFFFFF	PCI bus	OK
0xEFC00000-0xEFFFFFFF	PCI bus	OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC Version	Manufacturer Size	Description Creation Date	Status	File
OK	C:\WINDOWS\system32\MSADP32.ACM	Microsoft Corporation 5.2.3790.1830		(srv03_sp1_rtm.050324-1447) 23.50 KB (24,064 bytes) 3/25/2005 7:00 AM
OK	C:\WINDOWS\system32\IMAADP32.ACM	Microsoft Corporation 5.2.3790.1830		(srv03_sp1_rtm.050324-1447) 24.00 KB (24,576 bytes) 3/25/2005 7:00 AM
OK	C:\WINDOWS\system32\MSGSM32.ACM	Microsoft Corporation 5.2.3790.1830		(srv03_sp1_rtm.050324-1447) 34.50 KB (35,328 bytes) 3/25/2005 7:00 AM
OK	C:\WINDOWS\system32\TSSOFT32.ACM	DSP GROUP, INC. 1.01		(13,824 bytes) 3/25/2005 7:00 AM
OK	C:\WINDOWS\system32\MSG711.ACM	Microsoft Corporation 5.2.3790.1830		(srv03_sp1_rtm.050324-1447) 13.50 KB (13,824 bytes) 3/25/2005 7:00 AM

[Video Codecs]

CODEC Version	Manufacturer Size	Description Creation Date	Status	File
OK	C:\WINDOWS\system32\TSBYUV.DLL	Microsoft Corporation 5.2.3790.1830		(srv03_sp1_rtm.050324-1447) 12.50 KB (12,800 bytes) 3/24/2005 12:34 PM
OK	C:\WINDOWS\system32\MSYUV.DLL	Microsoft Corporation 5.2.3790.1830		(srv03_sp1_rtm.050324-1447) 21.00 KB (21,504 bytes) 3/24/2005 12:21 PM
OK	C:\WINDOWS\system32\MSVIDC32.DLL	Microsoft Corporation 5.2.3790.1830		(srv03_sp1_rtm.050324-1447) 43.00 KB (44,032 bytes) 3/25/2005 7:00 AM
OK	C:\WINDOWS\system32\IYUV_32.DLL	Microsoft Corporation 5.2.3790.1830		(srv03_sp1_rtm.050324-1447) 52.50 KB (53,760 bytes) 3/24/2005 12:19 PM
OK	C:\WINDOWS\system32\MSRLE32.DLL	Microsoft Corporation 5.2.3790.1830		(srv03_sp1_rtm.050324-1447) 15.50 KB (15,872 bytes) 3/25/2005 7:00 AM

[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\CDROMMATSHITA_DVD-ROM_SR-8178_____PJ22_____
 \5&A8D2D22&0&0.0.0

[Sound Device]

Item Value

[Display]

Item Value
 Name Radeon 7000 / RADEON VE Family (Microsoft Corporation)
 PNP Device ID
 PCI\VEN_1002&DEV_5159&SUBSYS_02C81014&REV_00\3&267A616A&0&08
 Adapter Type ATI display adapter (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 6.14.10.6509
 INF File atiixpag.inf (ati2mtag_RV100 section)
 Color Planes 1
 Color Table Entries 4294967296
 Resolution 1024 x 768 x 75 hertz
 Bits/Pixel 32
 Memory Address 0xF0000000-0xF7FFFFFFF
 I/O Port 0x00001800-0x000018FF
 Memory Address 0xF8800000-0xF8FFFFFFF
 IRQ Channel IRQ 16
 I/O Port 0x000003B0-0x000003BB
 I/O Port 0x000003C0-0x000003DF
 Memory Address 0xA0000-0xBFFFF

[Infrared]

Item Value

[Input]

[Keyboard]

Item Value
 Description USB Human Interface Device
 Name Enhanced (101- or 102-key)
 Layout 00000409
 PNP Device ID
 USB\VID_04B3&PID_4001&MI_00\6&25AF40CE&0&0000
 Number of Function Keys 12
 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
 Description USB Human Interface Device
 Name Enhanced (101- or 102-key)
 Layout 00000409
 PNP Device ID
 USB\VID_04B3&PID_4001&MI_00\6&25AF40CE&0&0000

Number of Function Keys 12

[Pointing Device]

Item Value
Hardware Type USB Human Interface Device
Number of Buttons 5
Status OK
PNP Device ID
USB\VID_04B3&PID_4001&MI_01\6&25AF40CE&0&0001
Power Management Supported No
Double Click Threshold 6
Handedness Right Handed Operation

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

Hardware Type USB Human Interface Device
Number of Buttons 5
Status OK
PNP Device ID
USB\VID_04B3&PID_4001&MI_01\6&25AF40CE&0&0001
Power Management Supported No
Double Click Threshold 6
Handedness Right Handed Operation

[Modem]

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 2/17/2006 4:07 PM
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 0xF8910000-0xF891FFFF
IRQ Channel IRQ 28

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 2/17/2006 4:07 PM
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 0xF8900000-0xF89FFFFF
IRQ Channel IRQ 24

Name [00000003] RAS Async Adapter
Adapter Type Not Available
Product Type RAS Async Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 2/17/2006 4:07 PM

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_L2TPMINIPOINT\0000
Last Reset 2/17/2006 4:07 PM

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_PPTPMINIPOINT\0000
Last Reset 2/17/2006 4:07 PM

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_PPPOEMINIPOINT\0000
 Last Reset 2/17/2006 4:07 PM
 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_PTIMINIPOINT\0000
 Last Reset 2/17/2006 4:07 PM
 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 2/17/2006 4:07 PM
 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

Name [00000009] 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&1D521019&0&08
 Last Reset 2/17/2006 4:07 PM
 Index 9
 Service Name b57nd
 IP Address 0.0.0.0
 IP Subnet 0.0.0.0
 Default IP Gateway Not Available
 DHCP Enabled Yes
 DHCP Server
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available

MAC Address 00:0D:60:98:10:2C
 Memory Address 0xEF100000-0xEF1FFFFFF
 IRQ Channel IRQ 96

Name [00000010] 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&1D521019&0&09
 Last Reset 2/17/2006 4:07 PM
 Index 10
 Service Name b57nd
 IP Address 192.168.122.210
 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:60:98:10:2D
 Memory Address 0xEF110000-0xEF11FFFF
 IRQ Channel IRQ 100

[Protocol]

Item	Value
Name	MSAFD Tcpi [TCP/IP]
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	16 bytes
Maximum Message Size	0 bytes
Message Oriented	No
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	Yes
Supports Graceful Closing	Yes
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD Tcpi [UDP/IP]
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.93 KB (65,467 bytes)
Message Oriented	Yes
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	Yes

Name	RSVP UDP Service Provider
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No

Maximum Address Size 16 bytes
 Maximum Message Size 63.93 KB (65,467 bytes)
 Message Oriented Yes
 Minimum Address Size 16 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting Yes
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption Yes
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting Yes

Name RSVP TCP Service Provider
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 16 bytes
 Maximum Message Size 0 bytes
 Message Oriented No
 Minimum Address Size 16 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption Yes
 Supports Expedited Data Yes
 Supports Graceful Closing Yes
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

[WinSock]

Item Value
 File c:\windows\system32\wsock32.dll
 Size 24.50 KB (25,088 bytes)
 Version 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)

[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

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

[Parallel]

Item Value

[Storage]

[Drives]

Item Value
 Drive C:
 Description Local Fixed Disk
 Compressed No
 File System NTFS
 Size 33.90 GB (36,398,149,632 bytes)
 Free Space 8.42 GB (9,042,464,768 bytes)
 Volume Name
 Volume Serial Number F484F62B

Drive D:
 Description CD-ROM Disc

Drive E:
 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 L:
 Description Local Fixed Disk
 Compressed Not Available
 File System Not Available
 Size Not Available
 Free Space Not Available
 Volume Name Not Available
 Volume Serial Number Not Available

[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 12
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 6
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #20, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12

SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 6
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #21, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0

SCSI Logical Unit 5
 SCSI Port 6
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #22, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0

SCSI Logical Unit 7
 SCSI Port 6
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #23, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0

SCSI Logical Unit 9
 SCSI Port 6
 SCSI Target ID 0
 Sectors/Track 63

Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #24, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 8
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #30, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 8
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #31, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 4
 SCSI Port 8
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255

Partition Disk #32, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 6
 SCSI Port 8
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255

Partition Disk #33, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 8
 SCSI Port 8
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255

Partition Disk #34, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 4
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255

Partition Disk #10, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 4
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #11, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 4
 SCSI Port 4
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #12, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 6
 SCSI Port 4
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #13, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 8
 SCSI Port 4
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #14, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 9
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #35, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 9
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #36, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 5
 SCSI Port 9
 SCSI Target ID 0

Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #37, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 7
 SCSI Port 9
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #38, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 9
 SCSI Port 9
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #39, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 3
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965

Tracks/Cylinder 255
 Partition Disk #5, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 3
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #6, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 5
 SCSI Port 3
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #7, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 7
 SCSI Port 3
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #8, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 9
 SCSI Port 3
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #9, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 5
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #15, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 5
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #16, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 5
 SCSI Port 5
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #17, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 7
 SCSI Port 5
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #18, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 9
 SCSI Port 5
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #19, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 13
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 2

SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #0, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 2
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #1, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 4
 SCSI Port 2
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #2, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 6
 SCSI Port 2
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795

Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #3, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 8
 SCSI Port 2
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #4, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 7
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #25, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 7
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #26, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 4
 SCSI Port 7
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #27, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 6
 SCSI Port 7
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #28, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 12
 SCSI Bus 0
 SCSI Logical Unit 8
 SCSI Port 7
 SCSI Target ID 0
 Sectors/Track 63
 Size 467.61 GB (502,095,767,040 bytes)
 Total Cylinders 61,043
 Total Sectors 980,655,795
 Total Tracks 15,565,965
 Tracks/Cylinder 255
 Partition Disk #29, Partition #0
 Partition Size 467.61 GB (502,087,541,760 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 10
 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 #40, 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 10
 SCSI Target ID 1
 Sectors/Track 63
 Size 339.01 GB (364,009,766,400 bytes)
 Total Cylinders 44,255
 Total Sectors 710,956,575
 Total Tracks 11,285,025
 Tracks/Cylinder 255
 Partition Disk #41, Partition #0
 Partition Size 339.00 GB (364,001,541,120 bytes)
 Partition Starting Offset 8,225,280 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 0xF8A20000-0xF8A20FFF
 IRQ Channel IRQ 18

Name QLogic Fibre Channel Adapter
 Manufacturer QLogic
 Status OK
 PNP Device ID
 PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&29E81982&&08
 I/O Port 0x00002200-0x000023FF
 Memory Address 0xF8B20000-0xF8B20FFF
 IRQ Channel IRQ 19

Name QLogic Fibre Channel Adapter
 Manufacturer QLogic
 Status OK
 PNP Device ID
 PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&172E68DD&&0&08
 I/O Port 0x00002400-0x000025FF

Memory Address 0xF8C20000-0xF8C20FFF
 IRQ Channel IRQ 52

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 0x00002600-0x000027FF
 Memory Address 0xF8D20000-0xF8D20FFF
 IRQ Channel IRQ 53

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 0xF8E80000-0xF8E80FFF
 IRQ Channel IRQ 55

Name QLogic Fibre Channel Adapter
 Manufacturer QLogic
 Status OK
 PNP Device ID
 PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&300BC0BE&0&08
 I/O Port 0x00004200-0x000043FF
 Memory Address 0xEF220000-0xEF220FFF
 IRQ Channel IRQ 90

Name QLogic Fibre Channel Adapter
 Manufacturer QLogic
 Status OK
 PNP Device ID
 PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&72AA75C&0&08
 I/O Port 0x00004400-0x000045FF
 Memory Address 0xEF320000-0xEF320FFF
 IRQ Channel IRQ 91

Name QLogic Fibre Channel Adapter
 Manufacturer QLogic
 Status OK
 PNP Device ID
 PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&19E45801&0&08
 I/O Port 0x00004600-0x000047FF
 Memory Address 0xEF420000-0xEF420FFF
 IRQ Channel IRQ 124

Name QLogic Fibre Channel Adapter
 Manufacturer QLogic
 Status OK
 PNP Device ID
 PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&2C9E08A6&0&08
 I/O Port 0x00004800-0x000049FF
 Memory Address 0xEF520000-0xEF520FFF
 IRQ Channel IRQ 125

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

Name Primary IDE Channel
 Manufacturer (Standard IDE ATA/ATAPI controllers)
 Status OK
 PNP Device ID PCIIDE\IDECHANNEL\4&101988B2&0&0
 I/O Port 0x000001F0-0x000001F7
 I/O Port 0x000003F6-0x000003F6
 IRQ Channel IRQ 14

Name Secondary IDE Channel
 Manufacturer (Standard IDE ATA/ATAPI controllers)
 Status OK
 PNP Device ID PCIIDE\IDECHANNEL\4&101988B2&0&1
 I/O Port 0x00000170-0x00000177
 I/O Port 0x00000376-0x00000376

[Printing]

Name	Driver	Port Name	Server Name
------	--------	-----------	-------------

[Problem Devices]

Device	PNP Device ID	Error Code
--------	---------------	------------

[USB]

Device	PNP Device ID
NEC PCI to USB Open Host Controller	PCI\VEN_1033&DEV_0035&SUBSYS_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
NEC PCI to USB Open Host Controller	PCI\VEN_1033&DEV_0035&SUBSYS_00351033&REV_43\3&A985F74&0&18
NEC PCI to USB Open Host Controller	PCI\VEN_1033&DEV_0035&SUBSYS_00351033&REV_43\3&A985F74&0&19
Standard Enhanced PCI to USB Host Controller	PCI\VEN_1033&DEV_00E0&SUBSYS_00E01033&REV_04\3&A985F74&0&1A

[Software Environment]

[System Drivers]

Name	Description	File	Type	Started	Start Mode
State	Status	Error Control	Accept	Pause	Accept Stop
abiosdsk	Abiosdsk	Not Available	Kernel Driver		No
Disabled	Stopped	OK	Ignore	No	No
acpi	Microsoft ACPI Driver				
c:\windows\system32\drivers\acpi.sys			Kernel Driver		Yes
Boot	Running	OK	Normal	No	Yes

Terminal Server Mouse Driver	No	SYSTEM	5.2.3790.1830		pmxdrv	Not Available	LEGACYDRIVER	Not Available
10/1/2002 (Standard system devices)		machine.inf	Not		Not Available	Not Available	Not Available	Not
Available	ROOT\RDP_MOU\0000				Available	ROOT\LEGACY_PMXDRV\0000		
Terminal Server Keyboard Driver	No	SYSTEM			Partition Manager	Not Available	LEGACYDRIVER	Not
5.2.3790.1830	10/1/2002 (Standard system devices)	machine.inf			Available	Not Available	Not Available	Not Available
Not Available	ROOT\RDP_KBD\0000				Not Available	ROOT\LEGACY_PARTMGR\0000		
Terminal Server Device Redirector	No	SYSTEM			NetBios over Tcpi	Not Available	LEGACYDRIVER	Not
5.2.3790.1830	10/1/2002 (Standard system devices)	machine.inf			Available	Not Available	Not Available	Not Available
Not Available	ROOT\RDPDR\0000				Not Available	ROOT\LEGACY_NETBT\0000		
Direct Parallel	No	NET	5.2.3790.1830	10/1/2002	NDProxy	Not Available	LEGACYDRIVER	Not Available
Microsoft netrasa.inf	Not Available	ROOT\MS_PTMINIPORT\0000			Not Available	Not Available	Not Available	Not
WAN Miniport (PPTP)	No	NET	5.2.3790.1830		Available	ROOT\LEGACY_NDPROXY\0000		
10/1/2002 Microsoft netrasa.inf	Not Available	ROOT\MS_PPTMINIPORT\0000			NDIS Usermode I/O Protocol	Not Available	LEGACYDRIVER	
WAN Miniport (PPPOE)	No	NET	5.2.3790.1830		Not Available	Not Available	Not Available	Not
10/1/2002 Microsoft netrasa.inf	Not Available	ROOT\MS_PPPOEMINIPORT\0000			Available	Not Available	ROOT\LEGACY_NDISUIO\0000	
WAN Miniport (IP)	No	NET	5.2.3790.1830	10/1/2002	Remote Access NDIS TAPI Driver	Not Available	LEGACYDRIVER	Not Available
Microsoft netrasa.inf	Not Available	ROOT\MS_NDISWANIP\0000			Available	Not Available	Not Available	Not
WAN Miniport (L2TP)	No	NET	5.2.3790.1830		Available	Not Available	Not Available	Not
10/1/2002 Microsoft netrasa.inf	Not Available	ROOT\MS_L2TPMINIPORT\0000			NDIS System Driver	Not Available	LEGACYDRIVER	Not
Video Codecs	No	MEDIA	5.2.3790.1830	10/1/2002	Available	Not Available	Not Available	Not Available
(Standard system devices)	wave.inf	Not Available			Not Available	ROOT\LEGACY_NDIS\0000		
Legacy Video Capture Devices	No	MEDIA	5.2.3790.1830		mountmgr	Not Available	LEGACYDRIVER	Not Available
10/1/2002 (Standard system devices)	wave.inf	Not Available			Not Available	Not Available	Not Available	Not
Media Control Devices	No	MEDIA	5.2.3790.1830		Available	ROOT\LEGACY_MOUNTMGR\0000		
10/1/2002 (Standard system devices)	wave.inf	Not Available			mnmdd	Not Available	LEGACYDRIVER	Not Available
Legacy Audio Drivers	No	MEDIA	5.2.3790.1830	10/1/2002	Not Available	Not Available	Not Available	Not
(Standard system devices)	wave.inf	Not Available			Available	ROOT\LEGACY_MNMDD\0000		
Audio Codecs	No	MEDIA	5.2.3790.1830	10/1/2002	ksecdd	Not Available	LEGACYDRIVER	Not Available
(Standard system devices)	wave.inf	Not Available			Not Available	Not Available	Not Available	Not
Remote Access IP ARP Driver	Not Available	LEGACYDRIVER			Available	ROOT\LEGACY_KSECDD\0000		
Not Available	Not Available	Not Available	Not		IPSEC driver	Not Available	LEGACYDRIVER	Not
Available	Not Available	ROOT\LEGACY_WANARP\0000			Available	Not Available	Not Available	Not Available
volsnap	Not Available	LEGACYDRIVER	Not Available		Not Available	ROOT\LEGACY_IPSEC\0000		
Not Available	Not Available	Not Available	Not		IP Network Address Translator	Not Available	LEGACYDRIVER	
Available	ROOT\LEGACY_VOLSNAP\0000				Not Available	Not Available	Not Available	Not
VGA Display Controller.	Not Available	LEGACYDRIVER			Available	Not Available	ROOT\LEGACY_IPNAT\0000	
Not Available	Not Available	Not Available	Not		Generic Packet Classifier	Not Available	LEGACYDRIVER	
Available	Not Available	ROOT\LEGACY_VGASAVE\0000			Not Available	Not Available	Not Available	Not
TDTCP	Not Available	LEGACYDRIVER	Not Available		Available	Not Available	ROOT\LEGACY_GPC\0000	
Not Available	Not Available	Not Available	Not		Fips	Not Available	LEGACYDRIVER	Not Available
Available	ROOT\LEGACY_TDTCP\0000				Not Available	Not Available	Not Available	Not
TCP/IP Protocol Driver	Not Available	LEGACYDRIVER			Available	ROOT\LEGACY_FIPS\0000		
Not Available	Not Available	Not Available	Not		em	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	ROOT\LEGACY_TCPIP\0000			Not Available	Not Available	Not Available	Not
tbs	Not Available	LEGACYDRIVER	Not Available		Available	ROOT\LEGACY_EM\0000		
Not Available	Not Available	Not Available	Not		dmload	Not Available	LEGACYDRIVER	Not Available
Available	ROOT\LEGACY_TBS\0000				Not Available	Not Available	Not Available	Not
Security Driver	Not Available	LEGACYDRIVER	Not		Available	ROOT\LEGACY_DMLOAD\0000		
Not Available	Not Available	Not Available	Not Available		dmboot	Not Available	LEGACYDRIVER	Not Available
Not Available	ROOT\LEGACY_SECDRV\0000				Not Available	Not Available	Not Available	Not
RDPWD	Not Available	LEGACYDRIVER	Not Available		Available	ROOT\LEGACY_DMBOOT\0000		
Not Available	Not Available	Not Available	Not		CRC Disk Filter Driver	Not Available	LEGACYDRIVER	
Available	ROOT\LEGACY_RDPWD\0000				Not Available	Not Available	Not Available	Not
RDPD	Not Available	LEGACYDRIVER	Not Available		Available	Not Available	ROOT\LEGACY_CRCDISK\0000	
Not Available	Not Available	Not Available	Not		Available	Not Available	LEGACYDRIVER	Not
Available	ROOT\LEGACY_RDPD\0000				Available	Not Available	LEGACYDRIVER	Not
Remote Access Auto Connection Driver	Not Available	LEGACYDRIVER	Not Available		Not Available	Not Available	Not Available	Not
LEGACYDRIVER	Not Available	Not Available	Not		Beep	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available			Not Available	Not Available	Not Available	Not
Available	ROOT\LEGACY_RASACD\0000				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 5.2.3790.1830	10/1/2002	SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&29C1039C&0&003
Microsoft volume.inf	Not Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEF7463DCOFFSET874857C00LENGTH7D0400				Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available
Generic volume	No	VOLUME 5.2.3790.1830	10/1/2002	SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&29C1039C&0&0001
Microsoft volume.inf	Not Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEF7463DCOFFSET8437EB400LENGTH31064A00				QLogic Fibre Channel Adapter No SCSIADAPTER 9.0.1.64 2/18/2005 QLogic oem0.inf Not Available
Generic volume	No	VOLUME 5.2.3790.1830	10/1/2002	PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&2C9E08A6&0&08
Microsoft volume.inf	Not Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEF7463DCOFFSET813F07200LENGTH2F8DC400				PCI standard host CPU bridge No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available
Generic volume	No	VOLUME 5.2.3790.1830	10/1/2002	PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_02\3&2C9E08A6&0&00
Microsoft volume.inf	Not Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEF7463DCOFFSET807313E00LENGTHCBBE600				PCI bus No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available
Generic volume	No	VOLUME 5.2.3790.1830	10/1/2002	ACPI\PNP0A03\15
Microsoft volume.inf	Not Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEF7463DCOFFSET7D1436200LENGTH35ED5E00				Qlogic processor device No SYSTEM 5.2.3790.1830 10/1/2002 QLOGIC scsidev.inf Not Available
Generic volume	No	VOLUME 5.2.3790.1830	10/1/2002	SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_4&3343B59&0&07F0
Microsoft volume.inf	Not Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEF7463DCOFFSET48B668800LENGTH345DC5C00				Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available
Generic volume	No	VOLUME 5.2.3790.1830	10/1/2002	SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3343B59&0&0009
Microsoft volume.inf	Not Available			
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEF7463DCOFFSET7E0000LENGTH48AE80A00				Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available
Volume Manager	No	SYSTEM 5.2.3790.1830	10/1/2002	SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3343B59&0&0005
(Standard system devices)		machine.inf	Not Available	
ROOT\FTDISK\0000				Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available
Logical Disk Manager	No	SYSTEM 5.2.3790.1830	10/1/2002	SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3343B59&0&0003
(Standard system devices)		machine.inf	Not Available	
ROOT\DMIO\0000				Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available
PCI standard host CPU bridge	No	SYSTEM 5.2.3790.1830		SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3343B59&0&0001
10/1/2002 (Standard system devices)		machine.inf	Not Available	
Available				QLogic Fibre Channel Adapter No SCSIADAPTER 9.0.1.64 2/18/2005 QLogic oem0.inf Not Available
PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_02\3&36B90202&0&00				PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&19E45801&0&08
PCI bus	No	SYSTEM 5.2.3790.1830	10/1/2002 (Standard system devices)	PCI standard host CPU bridge No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\17		machine.inf	Not Available	
PCI standard host CPU bridge	No	SYSTEM 5.2.3790.1830		PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_02\3&19E45801&0&00
10/1/2002 (Standard system devices)		machine.inf	Not Available	
Available				PCI bus No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_02\3&3BCEF44&0&00				ACPI\PNP0A03\14
PCI bus	No	SYSTEM 5.2.3790.1830	10/1/2002 (Standard system devices)	Qlogic processor device No SYSTEM 5.2.3790.1830 10/1/2002 QLOGIC scsidev.inf Not Available
ACPI\PNP0A03\16		machine.inf	Not Available	
Qlogic processor device	No	SYSTEM 5.2.3790.1830		SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_4&29C1039C&0&07F0
10/1/2002 QLOGIC		scsidev.inf	Not Available	
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_4&29C1039C&0&07F0				Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available
Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002	SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&A64063C&0&008
(Standard disk drives)		disk.inf	Not Available	
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&29C1039C&0&009				Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available
Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002	SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&A64063C&0&006
(Standard disk drives)		disk.inf	Not Available	
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&29C1039C&0&007				Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available
Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002	SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&A64063C&0&004
(Standard disk drives)		disk.inf	Not Available	
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&29C1039C&0&005				Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available
Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002	SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&A64063C&0&002
(Standard disk drives)		disk.inf	Not Available	
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&29C1039C&0&000				Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available
Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002	SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&A64063C&0&000
(Standard disk drives)		disk.inf	Not Available	

QLogic Fibre Channel Adapter	No	SCSIADAPTER	9.0.1.64	PCI\VEN_1166&DEV_0227&SUBSYS_00000000&REV_00\3&A985F74&0&7B
2/18/2005 QLogic oem0.inf	Not Available			ServerWorks Champion CSB6 - SouthBridge 6
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&72AA75C&0&08				No SYSTEM
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1830	5.2.3790.1830 10/1/2002 ServerWorks (RCC) machine.inf
10/1/2002 (Standard system devices)		machine.inf	Not Available	Not Available
Available				PCI\VEN_1166&DEV_0203&SUBSYS_00000000&REV_A0\3&A985F74&0&78
PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_02\3&72AA75C&0&00				USB Root Hub
PCI bus	No	SYSTEM	5.2.3790.1830	No USB 5.2.3790.1830 10/1/2002
system devices)		machine.inf	Not Available	(Standard USB Host Controller) usbport.inf Not Available
ACPI\PNP0A03\13				USB\ROOT_HUB20\4&26176FDA&0
Qlogic processor device	No	SYSTEM	5.2.3790.1830	Standard Enhanced PCI to USB Host Controller
10/1/2002 QLOGIC scsasidev.inf	Not Available			No USB
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_4&266D7871&0&07F0				5.2.3790.1830 10/1/2002 (Standard USB Host Controller) usbport.inf
Disk drive	No	DISKDRIVE	5.2.3790.1830	Not Available
(Standard disk drives) disk.inf	Not Available			PCI\VEN_1033&DEV_00E0&SUBSYS_00E01033&REV_04\3&A985F74&0&1A
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&266D7871&0&009				HID-compliant device
Disk drive	No	DISKDRIVE	5.2.3790.1830	No HIDCLASS 5.2.3790.1830
(Standard disk drives) disk.inf	Not Available			10/1/2002 (Standard system devices) input.inf Not Available
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&266D7871&0&007				HID\VID_04B3&PID_4001&MI_02\7&25DE2139&0&0000
Disk drive	No	DISKDRIVE	5.2.3790.1830	USB Human Interface Device
(Standard disk drives) disk.inf	Not Available			No HIDCLASS
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&266D7871&0&005				5.2.3790.1830 10/1/2002 (Standard system devices) input.inf
Disk drive	No	DISKDRIVE	5.2.3790.1830	Not Available
(Standard disk drives) disk.inf	Not Available			USB\VID_04B3&PID_4001&MI_02\6&2BDC6642&0&0002
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&266D7871&0&003				HID-compliant mouse
Disk drive	No	DISKDRIVE	5.2.3790.1830	No MOUSE 5.2.3790.1830 10/1/2002
(Standard disk drives) disk.inf	Not Available			Microsoft msmouse.inf Not Available
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&266D7871&0&001				HID\VID_04B3&PID_4001&MI_01\7&37A42F0&0&0000
QLogic Fibre Channel Adapter	No	SCSIADAPTER	9.0.1.64	USB Human Interface Device
2/18/2005 QLogic oem0.inf	Not Available			No HIDCLASS
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&300BC0BE&0&08				5.2.3790.1830 10/1/2002 (Standard system devices) input.inf
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1830	Not Available
10/1/2002 (Standard system devices)		machine.inf	Not Available	USB\VID_04B3&PID_4001&MI_01\6&2BDC6642&0&0001
Available				HID Keyboard Device
PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_02\3&300BC0BE&0&00				No KEYBOARD
PCI bus	No	SYSTEM	5.2.3790.1830	5.2.3790.1830 10/1/2002 (Standard keyboards) keyboard.inf
system devices)		machine.inf	Not Available	Not Available
ACPI\PNP0A03\12				HID\VID_04B3&PID_4001&MI_00\7&1B3DCF35&0&0000
Broadcom NetXtreme Gigabit Ethernet	No	NET	8.39.1.0	USB Human Interface Device
7/21/2005 Broadcom oem2.inf	Not Available			No HIDCLASS
PCI\VEN_14E4&DEV_1648&SUBSYS_02E71014&REV_10\3&1D521019&0&09				5.2.3790.1830 10/1/2002 (Standard system devices) input.inf
Broadcom NetXtreme Gigabit Ethernet	No	NET	8.39.1.0	Not Available
7/21/2005 Broadcom oem2.inf	Not Available			USB\VID_04B3&PID_4001&MI_00\6&2BDC6642&0&0000
PCI\VEN_14E4&DEV_1648&SUBSYS_02E71014&REV_10\3&1D521019&0&08				USB Composite Device
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1830	No USB 5.2.3790.1830
10/1/2002 (Standard system devices)		machine.inf	Not Available	10/1/2002 (Standard USB Host Controller) usb.inf Not Available
Available				USB\VID_04B3&PID_4001\000D6046DB25
PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_02\3&1D521019&0&00				USB Root Hub
PCI bus	No	SYSTEM	5.2.3790.1830	No USB 5.2.3790.1830 10/1/2002
system devices)		machine.inf	Not Available	(Standard USB Host Controller) usbport.inf Not Available
ACPI\PNP0A03\11				USB\ROOT_HUB\4&14A79F01&0
Serverworks Champion CSB6 - SouthBridge 6 LPC	No	SYSTEM	5.2.3790.1830	NEC PCI to USB Open Host Controller
5.2.3790.1830 10/1/2002 ServerWorks (RCC) machine.inf	Not Available			No USB
Not Available				5.2.3790.1830 10/1/2002 NEC usbport.inf Not Available
				PCI\VEN_1033&DEV_0035&SUBSYS_00351033&REV_43\3&A985F74&0&19
				USB Root Hub
				No USB 5.2.3790.1830 10/1/2002
				(Standard USB Host Controller) usbport.inf Not Available
				USB\ROOT_HUB\4&27614FA6&0
				NEC PCI to USB Open Host Controller
				No USB
				5.2.3790.1830 10/1/2002 NEC usbport.inf Not Available
				PCI\VEN_1033&DEV_0035&SUBSYS_00351033&REV_43\3&A985F74&0&18
				PCI standard host CPU bridge
				No SYSTEM 5.2.3790.1830
				10/1/2002 (Standard system devices) machine.inf Not Available
				Available
				PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_02\3&A985F74&0&00
				PCI bus
				No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices)
				machine.inf Not Available
				ACPI\PNP0A03\10
				Memory Module
				No MEMORY 5.2.3790.1830 10/1/2002
				Microsoft memory.inf Not Available ACPI\PNP0C80\2
				Intel Processor
				No PROCESSOR 5.2.3790.1830
				10/1/2002 Intel cpu.inf Not Available
				ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4\15

Intel Processor	No	PROCESSOR	5.2.3790.1830	Qlogic processor device	No	SYSTEM	5.2.3790.1830
10/1/2002 Intel	cpu.inf	Not Available		10/1/2002 QLOGIC	scsidev.inf	Not Available	
ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4\14				SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_4&1D1C1BB3&0&07F0			
Intel Processor	No	PROCESSOR	5.2.3790.1830	Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002
10/1/2002 Intel	cpu.inf	Not Available		(Standard disk drives) disk.inf	Not Available		
ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4\13				SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&1D1C1BB3&0&08			
Intel Processor	No	PROCESSOR	5.2.3790.1830	Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002
10/1/2002 Intel	cpu.inf	Not Available		(Standard disk drives) disk.inf	Not Available		
ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4\12				SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&1D1C1BB3&0&06			
Intel Processor	No	PROCESSOR	5.2.3790.1830	Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002
10/1/2002 Intel	cpu.inf	Not Available		(Standard disk drives) disk.inf	Not Available		
ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4\11				SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&1D1C1BB3&0&04			
Intel Processor	No	PROCESSOR	5.2.3790.1830	Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002
10/1/2002 Intel	cpu.inf	Not Available		(Standard disk drives) disk.inf	Not Available		
ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4\10				SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&1D1C1BB3&0&02			
Intel Processor	No	PROCESSOR	5.2.3790.1830	Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002
10/1/2002 Intel	cpu.inf	Not Available		(Standard disk drives) disk.inf	Not Available		
ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4\9				SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&1D1C1BB3&0&00			
Intel Processor	No	PROCESSOR	5.2.3790.1830	QLogic Fibre Channel Adapter	No	SCSIADAPTER	9.0.1.64
10/1/2002 Intel	cpu.inf	Not Available		2/18/2005 QLogic	oem0.inf	Not Available	
ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4\8				PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&474B838&0&08			
IBM Dummy Device	No	SYSTEM	5.2.3790.1830	PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1830
10/1/2002 IBM	scsidev.inf	Not Available		10/1/2002 (Standard system devices)	machine.inf	Not Available	
IBM	scsidev.inf	Not Available		Available			
SCSI\BRIDGE&VEN_IBM&PROD_DUMMY_DEVICE&REV_7.10\5&804C5&0&300				PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&474B838&0&00			
SCSI Processor Device	No	SYSTEM	5.2.3790.1830	PCI bus	No	SYSTEM	5.2.3790.1830
10/1/2002 IBM	scsidev.inf	Not Available		10/1/2002 (Standard system devices)	machine.inf	Not Available	
SCSI\PROCESSOR&VEN_IBM&PROD_EXP400___S320&REV_D110\5&804C5&0&2F0				ACPI\PNP0A03\5			
SCSI Processor Device	No	SYSTEM	5.2.3790.1830	Qlogic processor device	No	SYSTEM	5.2.3790.1830
10/1/2002 IBM	scsidev.inf	Not Available		10/1/2002 QLOGIC	scsidev.inf	Not Available	
SCSI\PROCESSOR&VEN_IBM&PROD_EXP400___S320&REV_D110\5&804C5&0&1F0				SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_4&17455A85&0&07F0			
Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002	Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002
(Standard disk drives) disk.inf	Not Available			(Standard disk drives) disk.inf	Not Available		
SCSI\DISK&VEN_IBM&PROD_SERVERAID&REV_6.10\5&804C5&0&010				SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&17455A85&0&009			
Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002	Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002
(Standard disk drives) disk.inf	Not Available			(Standard disk drives) disk.inf	Not Available		
SCSI\DISK&VEN_IBM&PROD_SERVERAID&REV_6.10\5&804C5&0&000				SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&17455A85&0&007			
IBM ServeRAID 6M Controller	No	SCSIADAPTER	5.2.3790.1830	Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002
10/1/2002 IBM Corporation	pnpscsi.inf	Not Available		(Standard disk drives) disk.inf	Not Available		
Available				SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&17455A85&0&005			
PCI\VEN_9005&DEV_0250&SUBSYS_02791014&REV_02\4&29C8B970&0&4008				Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002
PCI standard PCI-to-PCI bridge	No	SYSTEM	5.2.3790.1830	(Standard disk drives) disk.inf	Not Available		
10/1/2002 (Standard system devices)	machine.inf	Not Available		SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&17455A85&0&001			
Available				Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\3&20FEA912&0&08				(Standard disk drives) disk.inf	Not Available		
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1830	SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&17455A85&0&003			
10/1/2002 (Standard system devices)	machine.inf	Not Available		Disk drive No	DISKDRIVE	5.2.3790.1830	10/1/2002
Available				(Standard disk drives) disk.inf	Not Available		
PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&20FEA912&0&00				SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&17455A85&0&001			
PCI bus	No	SYSTEM	5.2.3790.1830	QLogic Fibre Channel Adapter	No	SCSIADAPTER	9.0.1.64
10/1/2002 (Standard system devices)	machine.inf	Not Available		2/18/2005 QLogic	oem0.inf	Not Available	
Available				PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&172E68DD&0&08			
ACPI\PNP0A03\7				PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1830
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1830	10/1/2002 (Standard system devices)	machine.inf	Not Available	
10/1/2002 (Standard system devices)	machine.inf	Not Available		Available			
Available							
PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&E44F86D&0&00							
PCI bus	No	SYSTEM	5.2.3790.1830				
10/1/2002 (Standard system devices)	machine.inf	Not Available					
Available							
ACPI\PNP0A03\6							

PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&172E68DD&0&00	PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&1070020&0&00
PCI bus No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available	PCI bus No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\4	ACPI\PNP0A03\2
QLogic processor device No SYSTEM 5.2.3790.1830 10/1/2002 QLOGIC scsidev.inf Not Available	Broadcom NetXtreme Gigabit Ethernet No NET 8.39.1.0 7/21/2005 Broadcom oem2.inf Not Available
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&232FAFE&0&07F0	PCI\VEN_14E4&DEV_1648&SUBSYS_02E71014&REV_10\3&13C0B0C5&0&09
Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available	Broadcom NetXtreme Gigabit Ethernet No NET 8.39.1.0 7/21/2005 Broadcom oem2.inf Not Available
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&232FAFE&0&008	PCI\VEN_14E4&DEV_1648&SUBSYS_02E71014&REV_10\3&13C0B0C5&0&08
Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available	PCI standard host CPU bridge No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&232FAFE&0&006	PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&13C0B0C5&0&00
Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available	PCI bus No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&232FAFE&0&002	ACPI\PNP0A03\1
Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available	Memory Module No MEMORY5.2.3790.1830 10/1/2002 Microsoft memory.inf Not Available ACPI\PNP0C80\0
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&232FAFE&0&000	Intel Processor No PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not Available
QLogic Fibre Channel Adapter No SCSIADAPTER 9.0.1.64 2/18/2005 QLogic oem0.inf Not Available	ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_7
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&29E81982&0&08	Intel Processor No PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not Available
PCI standard host CPU bridge No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available	ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_6
PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&29E81982&0&00	Intel Processor No PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not Available
PCI bus No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available	ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_5
ACPI\PNP0A03\3	Intel Processor No PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not Available
QLogic processor device No SYSTEM 5.2.3790.1830 10/1/2002 QLOGIC scsidev.inf Not Available	ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_4
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_4&3B4E3515&0&07F0	Intel Processor No PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not Available
Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available	ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_3
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3B4E3515&0&008	Intel Processor No PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not Available
Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available	ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_2
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3B4E3515&0&006	Intel Processor No PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not Available
Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available	ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_1
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3B4E3515&0&004	Intel Processor No PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not Available
Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available	ACPI\GENUINEINTEL_-_EM64T_FAMILY_15_MODEL_4_0
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3B4E3515&0&002	ACPI Fixed Feature Button No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available
Disk drive No DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives) disk.inf Not Available	Available ACPI\FIXEDBUTTON\2&DABA3FF&0
SCSI\DISK&VEN_IBM&PROD_1742-900&REV_0520\4&3B4E3515&0&000	System board No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available
QLogic Fibre Channel Adapter No SCSIADAPTER 9.0.1.64 2/18/2005 QLogic oem0.inf Not Available	ACPI\PNP0C01\1
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&1070020&0&08	IBM Active PCI Device No SYSTEM 5.2.1.0 8/30/2005 IBM Corporation oem3.inf Not Available
PCI standard host CPU bridge No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available	ACPI\IBM37D4\2&DABA3FF&0
Available	Motherboard resources No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available
	Available ACPI\PNP0C02\3
	Numeric data processor No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available
	Available ACPI\PNP0C04\4&13245C1&0
	System speaker No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available
	ACPI\PNP0800\4&13245C1&0
	System CMOS/real time clock No SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available
	Available ACPI\PNP0B00\4&13245C1&0

System timer No SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
ACPI\PNP0100\4&13245C1&0
Direct memory access controller No SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\PNP0200\4&13245C1&0
Advanced programmable interrupt controller No SYSTEM
5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf
Not Available ACPI\PNP0003\4&13245C1&0
Communications Port No PORTS 5.2.3790.1830 10/1/2002
(Standard port types) mspports.inf Not Available
ACPI\PNP0501\2
Communications Port No PORTS 5.2.3790.1830 10/1/2002
(Standard port types) mspports.inf Not Available
ACPI\PNP0501\1
PS/2 Compatible Mouse No MOUSE 5.2.3790.1830
10/1/2002 Microsoft msmouse.inf Not Available
ACPI\PNP0F13\4&13245C1&0
Standard 101/102-Key or Microsoft Natural PS/2 Keyboard No
KEYBOARD 5.2.3790.1830 10/1/2002 (Standard keyboards)
keyboard.inf Not Available ACPI\PNP0303\4&13245C1&0
Serverworks Champion CSB6 - SouthBridge 6 LPC No SYSTEM
5.2.3790.1830 10/1/2002 ServerWorks (RCC) machine.inf
Not Available
PCI\VEN_1166&DEV_0227&SUBSYS_00000000&REV_00\3&267A616A&
0&7B
Secondary IDE Channel No HDC 5.2.3790.1830
10/1/2002 (Standard IDE ATA/ATAPI controllers) mshdc.inf Not
Available PCI\IDE\IDECHANNEL\4&101988B2&0&1
CD-ROM Drive No CDROM 5.2.3790.1830 10/1/2002
(Standard CD-ROM drives) cdrom.inf Not Available
IDE\CDROM\MATSHITA_DVD-ROM_SR-8178 _____ PJ22 _____
\5&A8D2D22&0&0.0.0
Primary IDE Channel No HDC 5.2.3790.1830 10/1/2002
(Standard IDE ATA/ATAPI controllers) mshdc.inf Not Available
PCI\IDE\IDECHANNEL\4&101988B2&0&0
Standard Dual Channel PCI IDE Controller No HDC
5.2.3790.1830 10/1/2002 (Standard IDE ATA/ATAPI controllers)
mshdc.inf Not Available
PCI\VEN_1166&DEV_0213&SUBSYS_02121166&REV_A0\3&267A616A&
0&79
ServerWorks Champion CSB6 - SouthBridge 6 No SYSTEM
5.2.3790.1830 10/1/2002 ServerWorks (RCC) machine.inf
Not Available
PCI\VEN_1166&DEV_0203&SUBSYS_00000000&REV_A0\3&267A616A&
0&78
USB Root Hub No USB 5.2.3790.1830 10/1/2002
(Standard USB Host Controller) usbport.inf Not Available
USB\ROOT_HUB20\4&2B778F81&0
Standard Enhanced PCI to USB Host Controller No USB
5.2.3790.1830 10/1/2002 (Standard USB Host Controller) usbport.inf
Not Available
PCI\VEN_1033&DEV_00E0&SUBSYS_00E01033&REV_04\3&267A616A&
0&1A
HID-compliant device No HIDCLASS 5.2.3790.1830
10/1/2002 (Standard system devices) input.inf Not Available
HID\VID_04B3&PID_4001&MI_02\7&8DF652D&0&0000
USB Human Interface Device No HIDCLASS
5.2.3790.1830 10/1/2002 (Standard system devices) input.inf
Not Available
USB\VID_04B3&PID_4001&MI_02\6&25AF40CE&0&0002
HID-compliant mouse No MOUSE 5.2.3790.1830 10/1/2002
Microsoft msmouse.inf Not Available
HID\VID_04B3&PID_4001&MI_01\7&1AF7D895&0&0000
USB Human Interface Device No HIDCLASS
5.2.3790.1830 10/1/2002 (Standard system devices) input.inf
Not Available
USB\VID_04B3&PID_4001&MI_01\6&25AF40CE&0&0001

HID Keyboard Device No KEYBOARD
5.2.3790.1830 10/1/2002 (Standard keyboards) keyboard.inf
Not Available
HID\VID_04B3&PID_4001&MI_00\7&3344C50&0&0000
USB Human Interface Device No HIDCLASS
5.2.3790.1830 10/1/2002 (Standard system devices) input.inf
Not Available
USB\VID_04B3&PID_4001&MI_00\6&25AF40CE&0&0000
USB Composite Device No USB 5.2.3790.1830
10/1/2002 (Standard USB Host Controller) usb.inf Not Available
USB\VID_04B3&PID_4001\000D6046DA09
USB Root Hub No USB 5.2.3790.1830 10/1/2002
(Standard USB Host Controller) usbport.inf Not Available
USB\ROOT_HUB\4&2DDBD7B&0
NEC PCI to USB Open Host Controller No USB
5.2.3790.1830 10/1/2002 NEC usbport.inf Not Available
PCI\VEN_1033&DEV_0035&SUBSYS_00351033&REV_43\3&267A616A&
0&19
USB Root Hub No USB 5.2.3790.1830 10/1/2002
(Standard USB Host Controller) usbport.inf Not Available
USB\ROOT_HUB\4&15976E20&0
NEC PCI to USB Open Host Controller No USB
5.2.3790.1830 10/1/2002 NEC usbport.inf Not Available
PCI\VEN_1033&DEV_0035&SUBSYS_00351033&REV_43\3&267A616A&
0&18
Plug and Play Monitor No MONITOR
5.2.3790.1830 10/1/2002 (Standard monitor types) monitor.inf
Not Available
DISPLAY\AVO0000\4&36FA8DD8&0&10000080&00&01
Plug and Play Monitor No MONITOR
5.2.3790.1830 10/1/2002 (Standard monitor types) monitor.inf
Not Available
DISPLAY\IBM029A\4&36FA8DD8&0&10000082&00&01
Radeon 7000 / RADEON VE Family (Microsoft Corporation) No
DISPLAY 6.14.10.6508 12/3/2004 ATI Technologies Inc.
atiixpag.inf Not Available
PCI\VEN_1002&DEV_5159&SUBSYS_02C81014&REV_00\3&267A616A&
0&08
PCI standard host CPU bridge No SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices) machine.inf Not
Available
PCI\VEN_1014&DEV_02A1&SUBSYS_00000000&REV_01\3&267A616A&
0&00
PCI bus No SYSTEM 5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\0
Microsoft ACPI-Compliant System No SYSTEM
5.2.3790.1830 10/1/2002 Microsoft acpi.inf Not Available
ACPI_HAL\PNP0C08\0
ACPI Multiprocessor x64-based PC No COMPUTER
5.2.3790.1830 10/1/2002 (Standard computers) hal.inf Not
Available ROOT\ACPI_HAL\0000
Not Available Not Available Not Available Not
Available Not Available Not Available Not Available
Not Available 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>
DB2TEMPDIR	C:\SQLLIB\	<SYSTEM>
FP_NO_HOST_CHECK	NO	<SYSTEM>

```

INCLUDE
C:\SQLLIB\INCLUDE;C:\SQLLIB\LIB;c:\mssdkx64\Include\C:\MsSDKx64\
include\crt <SYSTEM>
LIB c:\mssdkx64\Lib;C:\SQLLIB\LIB;c:\mssdkx64\Lib\AMD64
<SYSTEM>
MsSDK c:\MsSDKx64 <SYSTEM>
NUMBER_OF_PROCESSORS 16 <SYSTEM>
OS Windows_NT <SYSTEM>
Path
c:\mssdkx64\Bin\Win64\x86\AMD64;c:\mssdkx64\Bin;c:\mssdkx64\Bin\WinN
T;C:\MsSDKx64\Include\crt;C:\Perl\bin;%SystemRoot%\system32;%SystemR
oot%;%SystemRoot%\System32\Wbem;c:\tools;c:\tools\util;c:\exe\tools;c:\vim
63;c:\bin;C:\SQLLIB\BIN;C:\SQLLIB\FUNCTION;C:\SQLLIB\SAMPLES\RE
PL <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 8,
GenuineIntel <SYSTEM>
PROCESSOR_LEVEL 15 <SYSTEM>
PROCESSOR_REVISION 0408 <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\TPCC
TMP %USERPROFILE%\Local Settings\Temp DB2SERV1\TPCC

```

[Print Jobs]

Document Size	Owner	Notify	Status	Time Submitted
Start Time	Until Time	Elapsed Time	Pages Printed	Job ID
Priority	Parameters	Driver	Print Processor	Host Print Queue
Data Type	Name			

[Network Connections]

Local Name	Remote Name	Type	Status	User Name
Z:	\\fsserv\ddrive	Disk	Current Connection	
DB2SERV1\TPCC				

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set	Max Working Set
system idle process	Not Available	Not Available	Not Available	0	0
system	Not Available	4	8	0	1413120
smss.exe	Not Available	516	11	204800	1413120
csrss.exe	Not Available	504	13	Not Available	Not Available

winlogon.exe	c:\windows\system32\winlogon.exe	964	13	204800	1413120	2/17/2006 4:10 PM	5.2.3790.1830	(srv03_sp1_rtm.050324-1447)	901.00 KB (922,624 bytes)	3/25/2005 7:00 AM
services.exe	c:\windows\system32\services.exe	240	9	204800	1413120	2/17/2006 4:10 PM	5.2.3790.1830	(srv03_sp1_rtm.050324-1447)	216.50 KB (221,696 bytes)	3/25/2005 7:00 AM
lsass.exe	c:\windows\system32\lsass.exe	244	9	204800	1413120	2/17/2006 4:10 PM	5.2.3790.1830	(srv03_sp1_rtm.050324-1447)	14.00 KB (14,336 bytes)	3/25/2005 7:00 AM
svchost.exe	c:\windows\system32\svchost.exe	624	8	204800	1413120	2/17/2006 4:10 PM	5.2.3790.1830	(srv03_sp1_rtm.050324-1447)	24.50 KB (25,088 bytes)	3/25/2005 7:00 AM
svchost.exe	Not Available	720	8	Not Available	Not Available	2/17/2006 4:10 PM	Not Available	Not Available	Not Available	Not Available
svchost.exe	Not Available	820	8	Not Available	Not Available	2/17/2006 4:10 PM	Not Available	Not Available	Not Available	Not Available
svchost.exe	c:\windows\system32\svchost.exe	856	8	204800	1413120	2/17/2006 4:10 PM	5.2.3790.1830	(srv03_sp1_rtm.050324-1447)	24.50 KB (25,088 bytes)	3/25/2005 7:00 AM
msdtc.exe	Not Available	1792	8	Not Available	Not Available	2/17/2006 4:11 PM	Not Available	Not Available	Not Available	Not Available
ibmhpasv.exe	c:\windows\system32\ibmhpasv.exe	1996	8	204800	1413120	2/17/2006 4:11 PM	5.2.1.0	17.50 KB (17,920 bytes)	8/30/2005 11:48 AM	
wrshdnt.exe	c:\wrshdnt\wrshdnt.exe	2020	8	204800	1413120	2/17/2006 4:11 PM	2.23.00	92.00 KB (94,208 bytes)	9/7/2005 9:19 AM	
svchost.exe	Not Available	1620	8	Not Available	Not Available	2/17/2006 4:11 PM	Not Available	Not Available	Not Available	Not Available
svchost.exe	c:\windows\system32\svchost.exe	1736	8	204800	1413120	2/17/2006 4:11 PM	5.2.3790.1830	(srv03_sp1_rtm.050324-1447)	24.50 KB (25,088 bytes)	3/25/2005 7:00 AM
svchost.exe	c:\windows\system32\svchost.exe	1464	8	204800	1413120	2/17/2006 4:11 PM	5.2.3790.1830	(srv03_sp1_rtm.050324-1447)	24.50 KB (25,088 bytes)	3/25/2005 7:00 AM
wmiprvse.exe	Not Available	1472	8	Not Available	Not Available	2/17/2006 4:11 PM	Not Available	Not Available	Not Available	Not Available
explorer.exe	c:\windows\explorer.exe	4496	8	204800	1413120	2/17/2006 4:24 PM	6.00.3790.1830	(srv03_sp1_rtm.050324-1447)	1.30 MB (1,364,480 bytes)	3/25/2005 7:00 AM
db2systray.exe	c:\sqlib\bin\db2systray.exe	4692	8	204800	1413120	2/17/2006 4:24 PM	8.1.11.973	69.50 KB (71,168 bytes)	1/22/2006 5:00 PM	
cmd.exe	c:\windows\system32\cmd.exe	4980	8	204800	1413120	2/17/2006 4:24 PM	5.2.3790.1830	(srv03_sp1_rtm.050324-1447)	538.50 KB (551,424 bytes)	3/25/2005 7:00 AM
db2bp.exe	c:\sqlib\bin\db2bp.exe	4924	8	204800	1413120	2/17/2006 4:46 PM	8.1.11.973	1.15 MB (1,209,344 bytes)	1/22/2006 4:59 PM	
csrss.exe	Not Available	3316	13	Not Available	Not Available	2/17/2006 8:19 PM	Not Available	Not Available	Not Available	Not Available
winlogon.exe	c:\windows\system32\winlogon.exe	4892	13	204800	1413120	2/17/2006 8:19 PM	5.2.3790.1830	(srv03_sp1_rtm.050324-1447)	901.00 KB (922,624 bytes)	3/25/2005 7:00 AM

rdpclip.exe:c:\windows\system32\rdpclip.exe	3996	8	204800				user32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.04 MB
1413120	2/17/2006 8:19 PM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)					(1,085,952 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
99.00 KB (101,376 bytes)	9/6/2005 2:11 PM						c:\windows\system32\user32.dll		
explorer.exe	c:\windows\explorer.exe	936	8				gdi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	592.00 KB
204800	1413120	2/17/2006 8:19 PM	6.00.3790.1830				(606,208 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
(srv03_sp1_rtm.050324-1447)	1.30 MB (1,364,480 bytes)	3/25/2005					c:\windows\system32\gdi32.dll		
7:00 AM							nddeapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	25.00 KB
db2systray.exe	c:\sqlib\bin\db2systray.exe	4020	8				(25,600 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
204800	1413120	2/17/2006 8:20 PM	8.1.11.973 69.50 KB (71,168 bytes)	1/22/2006 5:00 PM			c:\windows\system32\nddeapi.dll		
cmd.exe	c:\windows\system32\cmd.exe	1060	8	204800			profmap	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	36.00 KB
1413120	2/17/2006 8:30 PM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)					(36,864 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
538.50 KB (551,424 bytes)	3/25/2005 7:00 AM						c:\windows\system32\profmap.dll		
taskmgr.exe	c:\windows\system32\taskmgr.exe		3388				netapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	589.00 KB
13	204800	1413120	2/17/2006 8:31 PM	5.2.3790.1830			(603,136 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
(srv03_sp1_rtm.050324-1447)	231.00 KB (236,544 bytes)	3/25/2005					c:\windows\system32\netapi32.dll		
7:00 AM							userenv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.02 MB
cmd.exe	c:\windows\system32\cmd.exe	1708	8	204800			(1,069,056 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
1413120	2/17/2006 8:39 PM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)					c:\windows\system32\userenv.dll		
538.50 KB (551,424 bytes)	3/25/2005 7:00 AM						psapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	29.00 KB
db2syscs.exe	c:\sqlib\bin\db2syscs.exe	4512	8				(29,696 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
204800	1413120	2/17/2006 11:03 PM	8.1.11.973 97.50 KB (99,840 bytes)	1/22/2006 5:00 PM			c:\windows\system32\psapi.dll		
db2mh.exe:c:\sqlib\bin\db2mh.exe		3384	8	204800			regapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	108.50 KB
1413120	2/17/2006 11:03 PM	8.1.11.973 5.50 KB (5,632 bytes)	1/22/2006				(111,104 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
5:25 PM							c:\windows\system32\regapi.dll		
db2bp.exe	c:\sqlib\bin\db2bp.exe	1884	8	204800			secur32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	120.00 KB
1413120	2/17/2006 11:03 PM	8.1.11.973 1.15 MB (1,209,344 bytes)	1/22/2006				(122,880 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
1/22/2006 4:59 PM							c:\windows\system32\secur32.dll		
helpctr.exe	c:\windows\pchealth\helpctr\binaries\helpctr.exe		3104	8			setupapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.45 MB
204800	1413120	2/17/2006 11:13 PM	5.2.3790.1830				(1,523,200 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
(srv03_sp1_rtm.050324-1447)	1.30 MB (1,363,456 bytes)	9/6/2005					c:\windows\system32\setupapi.dll		
2:17 PM							version	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	28.00 KB
wmiprvse.exe	Not Available	2660	8	Not			(28,672 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
Available	Not Available	2/17/2006 11:13 PM	Not Available				c:\windows\system32\version.dll		
Not Available	Not Available						winsta	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	89.00 KB
helpsvc.exe	c:\windows\pchealth\helpctr\binaries\helpsvc.exe		3432	8	204800	1413120	(91,136 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.52 MB (1,591,296 bytes)	9/6/2005 2:17 PM					c:\windows\system32\winsta.dll		
							ws2_32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	176.50 KB
							(180,736 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
							c:\windows\system32\ws2_32.dll		
							ws2help	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	30.50 KB
							(31,232 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
							c:\windows\system32\ws2help.dll		
							msgina	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.14 MB
							(1,193,472 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
							c:\windows\system32\msgina.dll		
							shsvcs	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	193.50 KB
							(198,144 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
							c:\windows\system32\shsvcs.dll		
							shlwapi	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	606.50 KB
							(621,056 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
							c:\windows\system32\shlwapi.dll		
							sfc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	6.00 KB
							(6,144 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
							c:\windows\system32\sfc.dll		
							sfc_os	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	183.50 KB
							(187,904 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
							c:\windows\system32\sfc_os.dll		
							wintrust	5.131.3790.1830 (srv03_sp1_rtm.050324-1447)	297.50 KB
							(304,640 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
							c:\windows\system32\wintrust.dll		
							imagehlp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	57.50 KB
							(58,880 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
							c:\windows\system32\imagehlp.dll		
							ole32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	2.43 MB
							(2,543,616 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
							c:\windows\system32\ole32.dll		
							comctl32	6.0 (srv03_sp1_rtm.050324-1447)	1.51 MB (1,584,128 bytes)
								9/6/2005 5:50 AM	Microsoft Corporation

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
winlogon	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	922,624 bytes	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\winlogon.exe
ntdll	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1,257,472 bytes	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\ntdll.dll
kernel32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1,500,160 bytes	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\kernel32.dll
advapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1,051,136 bytes	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\advapi32.dll
rpcrt4	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1,714,176 bytes	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\rpcrt4.dll
crypt32	5.131.3790.1830 (srv03_sp1_rtm.050324-1447)	1,428,992 bytes	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\crypt32.dll
msasn1	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	156,160 bytes	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\msasn1.dll
msvcrt	7.0.3790.1830 (srv03_sp1_rtm.050324-1447)	520,192 bytes	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\msvcrt.dll

c:\windows\winsxs\amd64_microsoft.windows.common-controls_6595b64144cf1df_6.0.3790.1830_x-ww_aced72afcomctl32.dll			clbcatq	2001.12.4720.1830 (srv03_sp1_rtm.050324-1447)	865.00 KB (885,760 bytes)	9/6/2005 2:11 PM	Microsoft Corporation	
winscard 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	230.00 KB (235,520 bytes)	3/25/2005 7:00 AM	c:\windows\system32\clbcatq.dll	comres	2001.12.4720.1830 (srv03_sp1_rtm.050324-1447)	779.50 KB (798,208 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
c:\windows\system32\winscard.dll			c:\windows\system32\comres.dll	ntmarta	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	222.50 KB (227,840 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
wtsapi32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	29.00 KB (29,696 bytes)	3/25/2005 7:00 AM	c:\windows\system32\ntmarta.dll	xpsp2res	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	2.77 MB (2,899,456 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
c:\windows\system32\wtsapi32.dll			c:\windows\system32\comres.dll	authz	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	167.00 KB (171,008 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
winmm 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	303.50 KB (310,784 bytes)	3/25/2005 7:00 AM	c:\windows\system32\authz.dll	cryptnet	5.131.3790.1830 (srv03_sp1_rtm.050324-1447)	108.50 KB (111,104 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
c:\windows\system32\winmm.dll			c:\windows\system32\cryptnet.dll	sensapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	10.50 KB (10,752 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
shell32 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	10.01 MB (10,492,416 bytes)	3/25/2005 7:00 AM	c:\windows\system32\sensapi.dll	sclgntfy	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	27.00 KB (27,648 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
c:\windows\system32\shell32.dll			c:\windows\system32\sclgntfy.dll	drprov	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	24.00 KB (24,576 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
sxs 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.91 MB (2,003,968 bytes)	3/25/2005 7:00 AM	c:\windows\system32\drprov.dll	ntlanman	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	71.50 KB (73,216 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
c:\windows\system32\sxs.dll			c:\windows\system32\ntlanman.dll	netui0	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	130.00 KB (133,120 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
rsaenh 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	241.96 KB (247,768 bytes)	3/25/2005 7:00 AM	c:\windows\system32\netui0.dll	netui1	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	338.50 KB (346,624 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
c:\windows\system32\rsaenh.dll			c:\windows\system32\netui1.dll	davclnt	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	38.00 KB (38,912 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
wldap32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	390.00 KB (399,360 bytes)	3/25/2005 7:00 AM	c:\windows\system32\davclnt.dll	rasapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	410.00 KB (419,840 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
c:\windows\system32\wldap32.dll			c:\windows\system32\rasapi32.dll	rasman	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	95.50 KB (97,792 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
csddl 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	151.50 KB (155,136 bytes)	3/25/2005 7:00 AM	c:\windows\system32\rasman.dll	tapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	332.50 KB (340,480 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
c:\windows\system32\csddl.dll			c:\windows\system32\tapi32.dll	msv1_0	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	253.00 KB (259,072 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
dimsntfy 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	28.00 KB (28,672 bytes)	3/25/2005 7:00 AM	c:\windows\system32\msv1_0.dll	iphlpapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	177.00 KB (181,248 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
c:\windows\system32\dimsntfy.dll			c:\windows\system32\iphlpapi.dll	cscui	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	441.00 KB (451,584 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
wlnotify 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	148.00 KB (151,552 bytes)	3/25/2005 7:00 AM	c:\windows\system32\cscui.dll	wbemprox	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	38.00 KB (38,912 bytes)	9/6/2005 2:10 PM	Microsoft Corporation
c:\windows\system32\wlnotify.dll			c:\windows\system32\wbemprox.dll	wbemcomn	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	524.00 KB (536,576 bytes)	3/25/2005 7:00 AM	Microsoft Corporation
mpr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	115.00 KB (117,760 bytes)	3/25/2005 7:00 AM	c:\windows\system32\wbemcomn.dll	wbemsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	58.00 KB (59,392 bytes)	9/6/2005 2:10 PM	Microsoft Corporation
c:\windows\system32\mpr.dll			c:\windows\system32\wbem\wbemsvc.dll					
oleaut32 5.2.3790.1830	1.06 MB (1,116,160 bytes)	3/25/2005 7:00 AM						
c:\windows\system32\oleaut32.dll								
winspool 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	247.00 KB (252,928 bytes)	3/25/2005 7:00 AM						
c:\windows\system32\winspool.drv								
comctl32 5.82 (srv03_sp1_rtm.050324-1447)	934.50 KB (956,928 bytes)	9/6/2005 5:50 AM						
c:\windows\winsxs\amd64_microsoft.windows.common-controls_6595b64144cf1df_5.82.3790.1830_x-ww_4d792d2acomctl32.dll								
uxtheme 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	494.50 KB (506,368 bytes)	3/25/2005 7:00 AM						
c:\windows\system32\uxtheme.dll								
samlib 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	69.00 KB (70,656 bytes)	3/25/2005 7:00 AM						
c:\windows\system32\samlib.dll								
mprapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	154.50 KB (158,208 bytes)	3/25/2005 7:00 AM						
c:\windows\system32\mprapi.dll								
activeds 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	348.50 KB (356,864 bytes)	3/25/2005 7:00 AM						
c:\windows\system32\activeds.dll								
adslrpc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	240.50 KB (246,272 bytes)	3/25/2005 7:00 AM						
c:\windows\system32\adslrpc.dll								
credui 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	202.00 KB (206,848 bytes)	3/25/2005 7:00 AM						
c:\windows\system32\credui.dll								
atl 3.05.2284	96.50 KB (98,816 bytes)	3/25/2005 7:00 AM						
c:\windows\system32\atl.dll								
rtutils 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	66.00 KB (67,584 bytes)	3/25/2005 7:00 AM						
c:\windows\system32\rtutils.dll								

fastprox 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 866.50 KB (887,296 bytes) 9/6/2005 2:10 PM Microsoft Corporation c:\windows\system32\wbem\fastprox.dll	esent 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 2.26 MB (2,366,976 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\esent.dll
msvcp60 7.0.3790.1830 (srv03_sp1_rtm.050324-1447) 919.50 KB (941,568 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\msvcp60.dll	ntdsatq 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 51.00 KB (52,224 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\ntdsatq.dll
ntdsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 127.50 KB (130,560 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\ntdsapi.dll	mswsock 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 478.00 KB (489,472 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\mswsock.dll
dnsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 297.50 KB (304,640 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\dnsapi.dll	scecli 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 308.00 KB (315,392 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\scecli.dll
services 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 216.50 KB (221,696 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\services.exe	ws03res 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 794.00 KB (813,056 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\ws03res.dll
ncobjapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 80.00 KB (81,920 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\ncobjapi.dll	hnetcfg 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 561.00 KB (574,464 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\hnetcfg.dll
scesrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 594.50 KB (608,768 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\scesrv.dll	wshtcpip 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 29.00 KB (29,696 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\wshtcpip.dll
umpnpgmr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 205.00 KB (209,920 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\umpnpgmr.dll	pstorsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 36.00 KB (36,864 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\pstorsvc.dll
eventlog 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 127.00 KB (130,048 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\eventlog.dll	psbase 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 124.00 KB (126,976 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\psbase.dll
lsass 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB (14,336 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\lsass.exe	dssenh 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 226.96 KB (232,408 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\dssenh.dll
lsasrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.50 MB (1,568,256 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\lsasrv.dll	svchost 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.50 KB (25,088 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\svchost.exe
samsrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.01 MB (1,059,328 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\samsrv.dll	rpss 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 672.00 KB (688,128 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\rpss.dll
cryptdll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 47.00 KB (48,128 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\cryptdll.dll	schedsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 308.50 KB (315,904 bytes) 9/6/2005 2:15 PM Microsoft Corporation c:\windows\system32\schedsvc.dll
msprivs 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 47.50 KB (48,640 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\msprivs.dll	msidle 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 9.00 KB (9,216 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\msidle.dll
kerberos 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 698.00 KB (714,752 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\kerberos.dll	wkssvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 221.00 KB (226,304 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\wkssvc.dll
netlogon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 666.00 KB (681,984 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\netlogon.dll	aelupsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 31.50 KB (32,256 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\aelupsvc.dll
w32time 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 400.50 KB (410,112 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\w32time.dll	apphelp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 241.00 KB (246,784 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\apphelp.dll
schannel 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 248.00 KB (253,952 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\schannel.dll	dmserver 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 36.50 KB (37,376 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\dmserver.dll
wdigest 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 130.50 KB (133,632 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\wdigest.dll	es 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 357.00 KB (365,568 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\es.dll
rassfm 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 36.00 KB (36,864 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\rassfm.dll	srvsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 156.50 KB (160,256 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\srvsvc.dll
kdcsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 409.00 KB (418,816 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\kdcsvc.dll	seclogon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 27.50 KB (28,160 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\seclogon.dll
ntlsa 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 2.81 MB (2,948,096 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\ntlsa.dll	sens 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 63.50 KB (65,024 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\sens.dll

trkwks	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	177.50 KB	ncprov	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	73.00 KB
(181,760 bytes)	3/25/2005 7:00 AM Microsoft Corporation		(74,752 bytes)	9/6/2005 2:10 PM Microsoft Corporation	
c:\windows\system32\trkwks.dll			c:\windows\system32\wbem\ncprov.dll		
wmisvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	227.00 KB	rasdlg	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	859.50 KB
(232,448 bytes)	9/6/2005 2:10 PM Microsoft Corporation		(880,128 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\wbem\wmisvc.dll			c:\windows\system32\rasdlg.dll		
vssapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.26 MB	rasadhlp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	12.00 KB
(1,320,960 bytes)	3/25/2005 7:00 AM Microsoft Corporation		(12,288 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\vssapi.dll			c:\windows\system32\rasadhlp.dll		
comsvcs	2001.12.4720.1830 (srv03_sp1_rtm.050324-1447)	2.06 MB	rastapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	83.50 KB
(2,156,544 bytes)	9/6/2005 2:11 PM Microsoft Corporation		(85,504 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\comsvcs.dll			c:\windows\system32\rastapi.dll		
wiarpc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	57.00 KB	rasppp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	297.00 KB
(58,368 bytes)	3/25/2005 7:00 AM Microsoft Corporation		(304,128 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\wiarpc.dll			c:\windows\system32\rasppp.dll		
netman	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	457.00 KB	ntlsapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	11.00 KB
(467,968 bytes)	3/25/2005 7:00 AM Microsoft Corporation		(11,264 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\netman.dll			c:\windows\system32\ntlsapi.dll		
netshell	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	2.32 MB	raschap	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	141.00 KB
(2,437,120 bytes)	3/25/2005 7:00 AM Microsoft Corporation		(144,384 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\netshell.dll			c:\windows\system32\raschap.dll		
clusapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	127.00 KB	rastls	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	236.50 KB
(130,048 bytes)	3/25/2005 7:00 AM Microsoft Corporation		(242,176 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\clusapi.dll			c:\windows\system32\rastls.dll		
wininet	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	1.13 MB	cryptui	5.131.3790.1830 (srv03_sp1_rtm.050324-1447)	705.50 KB
(1,186,304 bytes)	3/25/2005 7:00 AM Microsoft Corporation		(722,432 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\wininet.dll			c:\windows\system32\cryptui.dll		
wzcsapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	49.00 KB	ipbootp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	47.50 KB
(50,176 bytes)	3/24/2005 12:35 PM Microsoft Corporation		(48,640 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\wzcsapi.dll			c:\windows\system32\ipbootp.dll		
wzcsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	492.00 KB	wbemcons	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	65.50 KB
(503,808 bytes)	3/24/2005 12:35 PM Microsoft Corporation		(67,072 bytes)	9/6/2005 2:10 PM Microsoft Corporation	
c:\windows\system32\wzcsvc.dll			c:\windows\system32\wbem\wbemcons.dll		
wmi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	5.50 KB	pchsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	76.00 KB
(5,632 bytes)	3/25/2005 7:00 AM Microsoft Corporation		(77,824 bytes)	9/6/2005 2:17 PM Microsoft Corporation	
c:\windows\system32\wmi.dll			c:\windows\pchealth\helpctr\binaries\pchsvc.dll		
dhcpcsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	219.00 KB	ibmhpasv	5.2.1.0	17.50 KB (17,920 bytes)
(224,256 bytes)	3/25/2005 7:00 AM Microsoft Corporation		IBM Corporation	8/30/2005 11:48 AM	
c:\windows\system32\dhcpcsvc.dll			c:\windows\system32\ibmhpasv.exe		
netcfgx	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.29 MB	wrshdnt	2.23.00	92.00 KB (94,208 bytes)
(1,354,240 bytes)	3/25/2005 7:00 AM Microsoft Corporation		Denicomp Systems	9/7/2005 9:19 AM	
c:\windows\system32\netcfgx.dll			c:\wrshdnt\wrshdnt.exe		
winpsec	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	52.50 KB	wow64	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	245.00 KB
(53,760 bytes)	3/25/2005 7:00 AM Microsoft Corporation		(250,880 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\winpsec.dll			c:\windows\system32\wow64.dll		
wbemcore	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.24 MB	wow64win	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	280.00 KB
(1,299,968 bytes)	9/6/2005 2:10 PM Microsoft Corporation		(286,720 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\wbem\wbemcore.dll			c:\windows\system32\wow64win.dll		
esscli	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	626.50 KB	wow64cpu	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	18.50 KB
(641,536 bytes)	9/6/2005 2:10 PM Microsoft Corporation		(18,944 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\wbem\esscli.dll			c:\windows\system32\wow64cpu.dll		
wmiutils	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	171.00 KB	termsrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	354.50 KB
(175,104 bytes)	9/6/2005 2:10 PM Microsoft Corporation		(363,008 bytes)	9/6/2005 2:11 PM Microsoft Corporation	
c:\windows\system32\wbem\wmiutils.dll			c:\windows\system32\termsrv.dll		
repdrvfs	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	353.50 KB	icaapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	27.50 KB
(361,984 bytes)	9/6/2005 2:10 PM Microsoft Corporation		(28,160 bytes)	9/6/2005 2:11 PM Microsoft Corporation	
c:\windows\system32\wbem\repdrvfs.dll			c:\windows\system32\icaapi.dll		
wmiprvsd	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	743.00 KB	mstlsapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	187.00 KB
(760,832 bytes)	9/6/2005 2:10 PM Microsoft Corporation		(191,488 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\wbem\wmiprvsd.dll			c:\windows\system32\mstlsapi.dll		
wbemess	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	532.50 KB	rdpwsx	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	170.13 KB
(545,280 bytes)	9/6/2005 2:10 PM Microsoft Corporation		(174,216 bytes)	9/6/2005 2:11 PM Microsoft Corporation	
c:\windows\system32\wbem\wbemess.dll			c:\windows\system32\rdpwsx.dll		
rasmans	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	279.50 KB	tapisrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	394.50 KB
(286,208 bytes)	3/25/2005 7:00 AM Microsoft Corporation		(403,968 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
c:\windows\system32\rasmans.dll			c:\windows\system32\tapisrv.dll		
			unimdm	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	302.50 KB
			(309,760 bytes)	3/25/2005 7:00 AM Microsoft Corporation	
			c:\windows\system32\unimdm.tsp		

uniplat 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 18.50 KB (18,944 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\uniplat.dll	db2osse64 8.1.11.973 1.89 MB (1,977,856 bytes) 1/22/2006 5:00 PM International Business Machines Corporation c:\sqlib\bin\db2osse64.dll
kmddsp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 56.50 KB (57,856 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\kmddsp.tsp	db2app64 8.1.11.973 6.28 MB (6,587,392 bytes) 1/22/2006 4:59 PM International Business Machines Corporation c:\sqlib\bin\db2app64.dll
ndptsp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 76.00 KB (77,824 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\ndptsp.tsp	db2sys64 8.1.11.973 1.46 MB (1,528,320 bytes) 1/22/2006 5:00 PM International Business Machines Corporation c:\sqlib\bin\db2sys64.dll
ipconf 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 22.50 KB (23,040 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\ipconf.tsp	db2wint64 8.1.11.973 30.00 KB (30,720 bytes) 1/22/2006 5:00 PM International Business Machines Corporation c:\sqlib\bin\db2wint64.dll
h323 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 424.00 KB (434,176 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\h323.tsp	db2sysp64 8.1.11.973 160.50 KB (164,352 bytes) 1/22/2006 5:00 PM International Business Machines Corporation c:\sqlib\bin\db2sysp64.dll
hidphone 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 48.00 KB (49,152 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\hidphone.tsp	db2g11n64 8.1.11.973 437.00 KB (447,488 bytes) 1/22/2006 4:59 PM International Business Machines Corporation c:\sqlib\bin\db2g11n64.dll
hid 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 33.00 KB (33,792 bytes) 3/24/2005 12:18 PM Microsoft Corporation c:\windows\system32\hid.dll	db2locale64 8.1.11.973 41.00 KB (41,984 bytes) 1/22/2006 4:59 PM International Business Machines Corporation c:\sqlib\bin\db2locale64.dll
explorer 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 1.30 MB (1,364,480 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\explorer.exe	db2sec64 8.1.11.973 23.00 KB (23,552 bytes) 1/22/2006 4:59 PM International Business Machines Corporation c:\sqlib\bin\db2sec64.dll
browseui 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 1.53 MB (1,601,536 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\browseui.dll	db2trcapi64 8.1.11.973 33.00 KB (33,792 bytes) 1/22/2006 4:59 PM International Business Machines Corporation c:\sqlib\bin\db2trcapi64.dll
shdocvw 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 2.30 MB (2,416,128 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\shdocvw.dll	db2install64 8.1.11.973 15.50 KB (15,872 bytes) 1/22/2006 4:59 PM International Business Machines Corporation c:\sqlib\bin\db2install64.dll
themeui 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 530.50 KB (543,232 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\themeui.dll	db2genreg64 8.1.11.973 161.50 KB (165,376 bytes) 1/22/2006 4:59 PM International Business Machines Corporation c:\sqlib\bin\db2genreg64.dll
msimg32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 6.50 KB (6,656 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\msimg32.dll	db2dascmn64 8.1.11.973 102.50 KB (104,960 bytes) 1/22/2006 4:59 PM International Business Machines Corporation c:\sqlib\bin\db2dascmn64.dll
actxprxy 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 220.50 KB (225,792 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\actxprxy.dll	db2osse_db264 8.1.11.973 65.50 KB (67,072 bytes) 1/22/2006 4:59 PM International Business Machines Corporation c:\sqlib\bin\db2osse_db264.dll
linkinfo 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 30.00 KB (30,720 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\linkinfo.dll	db2mmcr 8.1.11.973 6.00 KB (6,144 bytes) 1/22/2006 5:07 PM International Business Machines Corporation c:\sqlib\msg(en_us)\db2mmcr.dll
ntshrui 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 184.00 KB (188,416 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\ntshrui.dll	cmd 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 538.50 KB (551,424 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\cmd.exe
urlmon 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 1.02 MB (1,074,176 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\urlmon.dll	db2bp 8.1.11.973 1.15 MB (1,209,344 bytes) 1/22/2006 4:59 PM International Business Machines Corporation c:\sqlib\bin\db2bp.exe
webcheck 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 439.00 KB (449,536 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\webcheck.dll	db2clpfp 8.1.11.973 958.50 KB (981,504 bytes) 1/22/2006 4:59 PM International Business Machines Corporation c:\sqlib\bin\db2clpfp.dll
wsock32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.50 KB (25,088 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\wsock32.dll	db2dasapi64 8.1.11.973 425.00 KB (435,200 bytes) 1/22/2006 4:59 PM International Business Machines Corporation c:\sqlib\bin\db2dasapi64.dll
stobject 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 142.50 KB (145,920 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\stobject.dll	db2daskrb64 8.1.11.973 27.50 KB (28,160 bytes) 1/22/2006 4:59 PM International Business Machines Corporation c:\sqlib\bin\db2daskrb64.dll
batmeter 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 41.50 KB (42,496 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\batmeter.dll	IBMOSAuthclient64 8.1.11.973 27.00 KB (27,648 bytes) 1/22/2006 5:11 PM International Business Machines Corporation c:\sqlib\security\plugin\ibm\client\ibmosauthclient64.dll
powrprof 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 32.50 KB (33,280 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\powrprof.dll	rdpsnd 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 25.00 KB (25,600 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\rdpsnd.dll
db2systay 8.1.11.973 69.50 KB (71,168 bytes) 1/22/2006 5:00 PM International Business Machines Corporation c:\sqlib\bin\db2systay.exe	scredir 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 38.50 KB (39,424 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\scredir.dll
	msacm32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 31.00 KB (31,744 bytes) 3/25/2005 7:00 AM Microsoft Corporation c:\windows\system32\msacm32.drv

msacm32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	112.00 KB	(114,688 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\msacm32.dll
imaadp32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	24.00 KB	(24,576 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\imaadp32.acm
msadp32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	23.50 KB	(24,064 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\msadp32.acm
msg711	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	13.50 KB	(13,824 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\msg711.acm
msgsm32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	34.50 KB	(35,328 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\msgsm32.acm
tssoft32	1.01	13.50 KB	(13,824 bytes)	3/25/2005 7:00 AM	DSP GROUP, INC.	c:\windows\system32\tssoft32.acm
tsd32	1.03	24.50 KB	(25,088 bytes)	3/25/2005 7:00 AM	DSP GROUP, INC.	c:\windows\system32\tsd32.dll
rdpclip	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	99.00 KB	(101,376 bytes)	9/6/2005 2:11 PM	Microsoft Corporation	c:\windows\system32\rdpclip.exe
browseic	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	63.00 KB	(64,512 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\browseic.dll
duser	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	445.00 KB	(455,680 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\duser.dll
shdoclc	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	589.50 KB	(603,648 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\shdoclc.dll
mlang	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	686.00 KB	(702,464 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\mlang.dll
mprui	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	67.50 KB	(69,120 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\mprui.dll
netui2	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	542.00 KB	(555,008 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\netui2.dll
comdlg32	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	446.50 KB	(457,216 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\comdlg32.dll
netmsg	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	179.00 KB	(183,296 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\netmsg.dll
netplwiz	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	938.50 KB	(961,024 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\netplwiz.dll
twext	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	94.00 KB	(96,256 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\twext.dll
zipfldr	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	449.50 KB	(460,288 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\zipfldr.dll
shimgvw	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	618.50 KB	(633,344 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\shimgvw.dll
gdiplus	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	2.74 MB	(2,876,416 bytes)	9/6/2005 5:50 AM	Microsoft Corporation	c:\windows\winsxs\amd64_microsoft.windows.gdiplus_6595b64144ccf1df_1.0.3790.1830_x-ww_56cdf238.gdiplus.dll
mmshext	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	75.50 KB	(77,312 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\mmshext.dll
hhsetup	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	78.00 KB	(79,872 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\hhsetup.dll
taskmgr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	231.00 KB	(236,544 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\taskmgr.exe
util.dll	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	37.00 KB	(37,888 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\util.dll
db2syscs	8.1.11.973	97.50 KB	(99,840 bytes)	1/22/2006 5:00 PM	International Business Machines Corporation	c:\sqlib\bin\db2syscs.exe
db2engn	8.1.11.973	26.74 MB	(28,039,168 bytes)	1/22/2006 4:59 PM	International Business Machines Corporation	c:\sqlib\bin\db2engn.dll
db2suba	8.1.11.973	282.50 KB	(289,280 bytes)	1/22/2006 5:00 PM	International Business Machines Corporation	c:\sqlib\bin\db2suba.dll
db2dl	8.1.11.973	623.50 KB	(638,464 bytes)	1/22/2006 4:59 PM	International Business Machines Corporation	c:\sqlib\bin\db2dl.dll
db2dlnk	8.1.11.973	417.00 KB	(427,008 bytes)	1/22/2006 4:59 PM	International Business Machines Corporation	c:\sqlib\bin\db2dlnk.dll
db2qg	8.1.11.973	810.00 KB	(829,440 bytes)	1/22/2006 5:00 PM	International Business Machines Corporation	c:\sqlib\bin\db2qg.dll
db2dstf	8.1.11.973	809.00 KB	(828,416 bytes)	1/22/2006 4:59 PM	International Business Machines Corporation	c:\sqlib\bin\db2dstf.dll
icuc26db2engn	2, 6, 2, 0	657.50 KB	(673,280 bytes)	11/25/2005 10:17 PM	IBM Corporation and others	c:\sqlib\bin\icuc26db2engn.dll
db2icuglue64	8.1.11.973	14.50 KB	(14,848 bytes)	1/22/2006 4:59 PM	International Business Machines Corporation	c:\sqlib\bin\db2icuglue64.dll
icudt26ldb2engn	2, 6, 2, 0	7.82 MB	(8,201,728 bytes)	11/25/2005 10:17 PM	IBM Corporation and others	c:\sqlib\bin\icudt26ldb2engn.dll
db2sqlin	8.1.11.973	905.50 KB	(927,232 bytes)	1/22/2006 5:00 PM	International Business Machines Corporation	c:\sqlib\bin\db2sqlin.dll
db2licm	8.1.11.973	241.50 KB	(247,296 bytes)	1/22/2006 4:59 PM	International Business Machines Corporation	c:\sqlib\bin\db2licm.dll
shfolder	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	34.00 KB	(34,816 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\shfolder.dll
dbghep	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.22 MB	(1,274,368 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\dbghep.dll
IBMOSgroups64	8.1.11.973	22.50 KB	(23,040 bytes)	1/22/2006 5:11 PM	International Business Machines Corporation	c:\sqlib\security\plugin\ibm\group\ibmosgroups64.dll
IBMOSauthserver64	8.1.11.973	27.00 KB	(27,648 bytes)	1/22/2006 5:11 PM	International Business Machines Corporation	c:\sqlib\security\plugin\ibm\server\ibmosauthserver64.dll
db2sqnx	8.1.11.973	678.00 KB	(694,272 bytes)	1/22/2006 5:00 PM	International Business Machines Corporation	c:\sqlib\bin\db2sqnx.dll
db2tcp1	8.1.11.973	36.50 KB	(37,376 bytes)	1/22/2006 5:00 PM	International Business Machines Corporation	c:\sqlib\bin\db2tcp1.dll
winnr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	30.00 KB	(30,720 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\winnr.dll
db2mh	8.1.11.973	5.50 KB	(5,632 bytes)	1/22/2006 5:25 PM	International Business Machines Corporation	c:\sqlib\bin\db2mh.exe
helpctr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.30 MB	(1,363,456 bytes)	9/6/2005 2:17 PM	Microsoft Corporation	c:\windows\pchealth\helpctr\binaries\helpctr.exe
hcappres	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	7.50 KB	(7,680 bytes)	9/6/2005 2:17 PM	Microsoft Corporation	c:\windows\pchealth\helpctr\binaries\hcappres.dll
itss	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	208.00 KB	(212,992 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\itss.dll
msxml3	8.70.1104.0	2.04 MB	(2,141,184 bytes)	3/25/2005 7:00 AM	Microsoft Corporation	c:\windows\system32\msxml3.dll

pchshell 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 155.00 KB
 (158,720 bytes) 9/6/2005 2:17 PM Microsoft Corporation
 c:\windows\pchealth\helpctr\binaries\pchshell.dll
 mshtml 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 5.65 MB
 (5,928,448 bytes) 3/25/2005 7:00 AM Microsoft Corporation
 c:\windows\system32\mshtml.dll
 msls31 3.10.349.0 357.00 KB (365,568 bytes) 3/25/2005 7:00 AM
 Microsoft Corporation c:\windows\system32\msls31.dll
 msimtf 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 380.50 KB
 (389,632 bytes) 3/25/2005 7:00 AM Microsoft Corporation
 c:\windows\system32\msimtf.dll
 msctf 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 617.50 KB
 (632,320 bytes) 3/25/2005 7:00 AM Microsoft Corporation
 c:\windows\system32\msctf.dll
 jscript 5.6.0.8827 974.50 KB (997,888 bytes) 3/25/2005 7:00 AM
 Microsoft Corporation c:\windows\system32\jscript.dll
 imm32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 208.00 KB
 (212,992 bytes) 3/25/2005 7:00 AM Microsoft Corporation
 c:\windows\system32\imm32.dll
 mshtmlmled 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 905.50 KB
 (927,232 bytes) 3/25/2005 7:00 AM Microsoft Corporation
 c:\windows\system32\mshtmlmled.dll
 vbscript 5.6.0.8827 646.50 KB (662,016 bytes) 3/25/2005 7:00 AM
 Microsoft Corporation c:\windows\system32\vbscript.dll
 msinfo 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 636.00 KB
 (651,264 bytes) 9/6/2005 2:17 PM Microsoft Corporation
 c:\windows\pchealth\helpctr\binaries\msinfo.dll
 mfc42u 6.50.9146.0 1.39 MB (1,462,272 bytes) 3/25/2005
 7:00 AM Microsoft Corporation c:\windows\system32\mfc42u.dll
 riched32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 7.00 KB
 (7,168 bytes) 3/25/2005 7:00 AM Microsoft Corporation
 c:\windows\system32\riched32.dll
 riched20 5.31.23.1224 1.10 MB (1,157,120 bytes) 3/25/2005
 7:00 AM Microsoft Corporation c:\windows\system32\riched20.dll
 helpsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.52 MB
 (1,591,296 bytes) 9/6/2005 2:17 PM 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\dlhost.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	Running	Manual	Own Process
c:\sql\lib\bin\db2syscs.exe		Normal	LocalSystem	0
DB2 - DB2-0	DB2-0	Stopped	Manual	Own Process
c:\sql\lib\bin\db2syscs.exe		Normal	.\tpcc	0
DB2DAS - DB2DAS00	DB2DAS00	Stopped	Manual	
Own Process	"c:\sql\lib\bin\db2dasrm.exe"	Normal	.\tpcc	0
DB2 Governor	DB2GOVERNOR	Stopped	Manual	Own
Process	c:\sql\lib\bin\db2govds.exe	Normal	.\tpcc	0
DB2 JDBC Applet Server	DB2JDS	Stopped	Manual	Own
Process	c:\sql\lib\bin\db2jds.exe	Normal	LocalSystem	0
DB2 License Server	DB2LICD	Stopped	Manual	Own Process
c:\sql\lib\bin\db2licd.exe		Ignore	LocalSystem	0
DB2 Registry Reflector	DB2NTREGREFLECTOR	Stopped		
Manual Own Process	c:\sql\lib\bin\db2reg64.exe	Normal		
.\tpcc	0			
DB2 Security Server	DB2NTSECSEVER	Stopped	Manual	Own
Process	c:\sql\lib\bin\db2sec.exe	Normal	LocalSystem	0
DB2 Remote Command Server	DB2REMOCECMD	Stopped	Manual	Manual
Own Process	c:\sql\lib\bin\db2rcmd.exe	Ignore	.\tpcc	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
c:\windows\system32\svchost.exe -k networkservice		Normal	NT	
AUTHORITY\NetworkService	0			
Error Reporting Service	ERSvc	Stopped	Manual	Share
Process	c:\windows\system32\svchost.exe -k winerr	Ignore	LocalSystem	0
Event Log	Eventlog	Running	Auto	Share Process
c:\windows\system32\services.exe		Normal	LocalSystem	0
COM+ Event System	EventSystem	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Help and Support	helpsvc	Running	Manual	Share Process
c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0
Human Interface Device Access	HidServ	Stopped	Disabled	Share
Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
HTTP SSLHTTPFilter		Stopped	Manual	Share Process
c:\windows\system32\lsass.exe		Normal	LocalSystem	0
IAS Jet Database Access	IASJet	Stopped	Manual	Share
Process	c:\windows\syswow64\svchost.exe -k iasjet	Normal	LocalSystem	0
IBM Active PCI Alert Service	IBMHPS	Running	Auto	Own
Process	c:\windows\system32\ibmhpasv.exe	Normal	LocalSystem	0
IMAPI CD-Burning COM Service	ImapiService	Stopped		
Disabled Own Process	c:\windows\system32\imapi.exe	Normal	LocalSystem	0

Intersite Messaging	IsmServ	Stopped	Disabled	Own Process		Remote Procedure Call (RPC) Locator	RpcLocator	Stopped	
c:\windows\system32\ismerv.exe			Normal	LocalSystem	0	Manual	Own Process	c:\windows\system32\locator.exe	Normal
Kerberos Key Distribution Center	kdc	Stopped	Disabled	Share Process		NT AUTHORITY\NetworkService	0		
Share Process	c:\windows\system32\lsass.exe		Normal	LocalSystem	0	Remote Procedure Call (RPC)	RpcSs	Running	Auto
Server	lanmanserver	Running	Auto	Share Process		Process	c:\windows\system32\svchost.exe -k rpcss	Normal	NT
c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		AUTHORITY\NetworkService	0		
Workstation	lanmanworkstation	Running	Auto	Share Process		Resultant Set of Policy Provider	RSOPProv	Stopped	Manual
Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0	Process	c:\windows\system32\rsopprov.exe	Normal	Share
License Logging	LicenseService	Stopped	Disabled	Own Process		Special Administration Console Helper	sacsvr	Stopped	Manual
Process	c:\windows\system32\llssrv.exe		Normal	NT		Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem
AUTHORITY\NetworkService	0					Security Accounts Manager	SamSs	Running	Auto
TCP/IP NetBIOS Helper	LmHosts	Running	Auto	Share Process		Process	c:\windows\system32\lsass.exe	Normal	LocalSystem
Process	c:\windows\system32\svchost.exe -k localservice		Normal	LocalSystem	0	Smart Card	SCardSvr	Stopped	Manual
NT AUTHORITY\LocalService	0					c:\windows\system32\scardsvr.exe		Ignore	NT
Messenger Messenger	Stopped	Disabled	Share Process			AUTHORITY\LocalService	0		
c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		Task Scheduler	Schedule	Running	Auto
NetMeeting Remote Desktop Sharing	mnmsrv	Stopped	Disabled	Share Process		c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem
Own Process	c:\windows\system32\mnmsrv.exe		Normal	LocalSystem	0	Secondary Logon	seclogon	Running	Auto
Distributed Transaction Coordinator	MSDTC	Running	Auto	Share Process		c:\windows\system32\svchost.exe -k netsvcs		Ignore	LocalSystem
Own Process	c:\windows\system32\msdtc.exe		Normal	NT		System Event Notification	SENS	Running	Auto
AUTHORITY\NetworkService	0					Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem
Windows Installer	MSIServer	Stopped	Manual	Share Process		ServeRAID Manager Agent	ServeRAIDManagerAgent	Stopped	Manual
c:\windows\system32\msiexec.exe /v		Normal	LocalSystem	0		Manual	Own Process	"c:\program files (x86)\raidman\raidserv.exe"	
Network DDE	NetDDE	Stopped	Disabled	Share Process		Normal	LocalSystem	0	
c:\windows\system32\netdde.exe		Normal	LocalSystem	0		Windows Firewall/Internet Connection Sharing (ICS)	SharedAccess	Stopped	Disabled
Network DDE DSDM	NetDDEdsdm	Stopped	Disabled	Share Process		Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe
Process	c:\windows\system32\netdde.exe		Normal	LocalSystem	0	-k netsvcs	Normal	LocalSystem	0
Net Logon	Netlogon	Stopped	Manual	Share Process		Shell Hardware Detection	ShellHWDetection	Running	Auto
c:\windows\system32\lsass.exe		Normal	LocalSystem	0		Share Process	c:\windows\system32\svchost.exe -k netsvcs	Ignore	LocalSystem
Network Connections	Netman	Running	Manual	Share Process		Print Spooler	Spooler	Stopped	Manual
c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		c:\windows\system32\spoolsv.exe		Normal	LocalSystem
Network Location Awareness (NLA)	Nla	Running	Manual	Share Process		Windows Image Acquisition (WIA)	stisvc	Stopped	Disabled
Share Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0	Share Process	c:\windows\system32\svchost.exe -k imgsvc	Normal	NT AUTHORITY\LocalService
File Replication	NtFrs	Stopped	Manual	Own Process		DB2 - SVT-0	SVT-0	Stopped	Manual
c:\windows\system32\ntfrs.exe		Ignore	LocalSystem	0		c:\sql\bin\db2syscs.exe		Normal	.\tpcc
NT LM Security Support Provider	NtLmSsp	Stopped	Manual	Share Process		DB2 - SVT-1	SVT-1	Stopped	Manual
Share Process	c:\windows\system32\lsass.exe		Normal	LocalSystem	0	c:\sql\bin\db2syscs.exe		Normal	.\tpcc
Removable Storage	NtmsSvc	Stopped	Manual	Share Process		DB2 - SVT-2	SVT-2	Stopped	Manual
c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		c:\sql\bin\db2syscs.exe		Normal	.\tpcc
Plug and Play	PlugPlay	Running	Auto	Share Process		DB2 - SVT-3	SVT-3	Stopped	Manual
c:\windows\system32\services.exe		Normal	LocalSystem	0		c:\sql\bin\db2syscs.exe		Normal	.\tpcc
IPSEC Services	PolicyAgent	Stopped	Manual	Share Process		DB2 - SVT-4	SVT-4	Stopped	Manual
Process	c:\windows\system32\lsass.exe		Normal	LocalSystem	0	c:\sql\bin\db2syscs.exe		Normal	.\tpcc
Protected Storage	ProtectedStorage	Running	Auto	Share Process		DB2 - SVT-5	SVT-5	Stopped	Manual
Process	c:\windows\system32\lsass.exe		Normal	LocalSystem	0	c:\sql\bin\db2syscs.exe		Normal	.\tpcc
Remote Access Auto Connection Manager	RasAuto	Stopped	Manual	Share Process		DB2 - SVT-6	SVT-6	Stopped	Manual
Share Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0	c:\sql\bin\db2syscs.exe		Normal	.\tpcc
Remote Access Connection Manager	RasMan	Running	Manual	Share Process		DB2 - SVT-7	SVT-7	Stopped	Manual
Share Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0	c:\sql\bin\db2syscs.exe		Normal	.\tpcc
Remote Desktop Help Session Manager	RDSessMgr	Stopped	Manual	Own Process		Microsoft Software Shadow Copy Provider	swprv	Stopped	Manual
Manual	Own Process		c:\windows\system32\sessmgr.exe			Own Process	c:\windows\system32\svchost.exe -k swprv	Normal	LocalSystem
Normal	LocalSystem		0			Performance Logs and Alerts	SysmonLog	Stopped	Auto
Remote Shell Daemon	Remote Shell Daemon	Running	Manual	Share Process		Own Process	c:\windows\system32\smlogsvc.exe	Normal	NT Authority\NetworkService
Auto	Own Process		c:\wrshdnt\wrshdnt.exe			NT Authority\NetworkService	0		
LocalSystem	0					Telephony TapiSrv	Running	Manual	Share Process
Routing and Remote Access	RemoteAccess	Stopped	Disabled	Share Process		c:\windows\system32\svchost.exe -k tapisrv		Normal	LocalSystem
Share Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0	Terminal Services	TermService	Running	Manual
Remote Registry	RemoteRegistry	Running	Auto	Share Process		Process	c:\windows\system32\svchost.exe -k termvcs	Normal	LocalSystem
Process	c:\windows\system32\svchost.exe -k regsvc		Normal	NT		Themes	Themes	Stopped	Disabled
AUTHORITY\LocalService	0					c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem

Telnet	TlntSvr	Stopped	Disabled	Own Process		ActiveState	ActivePerl 5.8	All Users:ActiveState	ActivePerl 5.8
c:\windows\system32\	tlntsvr.exe	Normal	NT AUTHORITY\LocalService	0		All Users			
Distributed Link Tracking Server	TrkSvr	Stopped	Disabled	Share		Administrative Tools	All Users:Administrative Tools	All Users	
Process	c:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0	IBM DB2	All Users:IBM DB2	All Users	
Distributed Link Tracking Client	TrkWks	Running	Auto	Share		IBM DB2\	Command Line Tools	All Users:IBM DB2\	Command Line Tools
Process	c:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0	IBM DB2\	Development Tools	All Users:IBM DB2\	Development Tools
Terminal Services Session Directory	Tssdis	Stopped	Disabled			IBM DB2\	General Administration Tools	All Users:IBM DB2\	General
Own Process	c:\windows\system32\tssdis.exe	Normal	LocalSystem	0		Administration Tools	All Users		
Windows User Mode Driver Framework	UMWdf	Stopped	Manual			IBM DB2\	Information	All Users:IBM DB2\	Information
Own Process	c:\windows\system32\wdfmgr.exe	Normal	LocalSystem	0		IBM DB2\	Monitoring Tools	All Users:IBM DB2\	Monitoring Tools
NT AUTHORITY\LocalService	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		
WebClient	WebClient	Stopped	Disabled	Share Process		AUTHORITY\SYSTEM			
Process	c:\windows\system32\svchost.exe	-k localservice	Normal	NT AUTHORITY\LocalService	0	Accessories\Entertainment	NT		
WinHTTP Web Proxy Auto-Discovery Service	WinHttpAutoProxySvc	Stopped	Manual	Share Process		AUTHORITY\SYSTEM:Accessories\Entertainment	NT		
Process	c:\windows\system32\svchost.exe	-k localservice	Normal	NT AUTHORITY\LocalService	0	AUTHORITY\SYSTEM			
Windows Management Instrumentation	winmgmt	Running	Auto	Share Process		Startup	NT AUTHORITY\SYSTEM:Startup	NT	
Process	c:\windows\system32\svchost.exe	-k netsvcs	Ignore	LocalSystem	0	AUTHORITY\SYSTEM			
Portable Media Serial Number Service	WmdmPmSN	Stopped	Manual	Share Process		Accessories	DB2SERV1\TPCC:Accessories	DB2SERV1\TPCC	
Process	c:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0	Accessories\Accessibility	DB2SERV1\TPCC:Accessories\Accessibility		
Windows Management Instrumentation Driver Extensions	Wmi	Stopped	Manual	Share Process		Accessories\Entertainment	DB2SERV1\TPCC:Accessories\Entertainment	DB2SERV1\TPCC	
Process	c:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0	Microsoft Visual C++ Toolkit 2003	DB2SERV1\TPCC:Microsoft	Visual C++ Toolkit 2003	DB2SERV1\TPCC
WMI Performance Adapter	WmiApSrv	Stopped	Manual	Own Process		Startup	DB2SERV1\TPCC:Startup	DB2SERV1\TPCC	
Process	c:\windows\system32\wbem\wmiaprv.exe	Normal	LocalSystem	0		Winsock RSHD-NT	DB2SERV1\TPCC:Winsock RSHD-NT		
Automatic Updates	wuauerv	Stopped	Manual	Share Process		DB2SERV1\TPCC			
Process	c:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0	db2systray.exe	DB2 c:\sqlib\bin\	db2systray.exe	db2 All Users
Wireless Configuration	WZCSVC	Stopped	Manual	Share Process		HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run			
Process	c:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0				
Network Provisioning Service	xmlprov	Stopped	Manual	Share Process					
Process	c:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0				

[Program Groups]

Group Name	Name	User Name	Default User
Accessories	Default User:Accessories	Default User	
Accessories\Accessibility	Default User:Accessories\Accessibility	Default User	
Accessories\Entertainment	Default User:Accessories\Entertainment	Default User	
Startup	Default User:Startup	Default User	
Accessories	All Users:Accessories	All Users	
Accessories\Accessibility	All Users:Accessories\Accessibility	All Users	
Accessories\Communications	All Users:Accessories\Communications	All Users	
Accessories\Entertainment	All Users:Accessories\Entertainment	All Users	
Accessories\System Tools	All Users:Accessories\System Tools	All Users	

[Startup Programs]

Program	Command	User Name	Location	Startup
desktop	desktop.ini	NT AUTHORITY\SYSTEM		Startup
desktop	desktop.ini	DB2SERV1\TPCC		Startup
desktop	desktop.ini	.DEFAULT		Startup
desktop	desktop.ini	All Users	Common	Startup
db2systray.exe	DB2 c:\sqlib\bin\	db2systray.exe	db2	All Users
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				

[OLE Registration]

Object	Local Server
Sound (OLE2)	sndrec32.exe
Media Clip	mplay32.exe
Video Clip	mplay32.exe /avi
MIDI Sequence	mplay32.exe /mid
Sound	Not Available
Media Clip	Not Available
WordPad Document	"%programfiles%\windows nt\accessories\wordpad.exe"
Bitmap Image	mspaint.exe

[Windows Error Reporting]

Time	Type	Details
------	------	---------

[Internet Settings]

[Internet Explorer]

[Following are sub-categories of this main category]
 [Summary]

Item	Value
Version	6.0.3790.1830
Build	63790.1830
Application Path	C:\Program Files\Internet Explorer
Language	English (United States)
Active Printer	Not Available
Cipher Strength	128-bit
Content Advisor	Disabled
IEAK Install	No

[File Versions]

File	Version	Size	Date	Path	Company
actxprxy.dll	6.0.3790.1830	221 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
advpack.dll	6.0.3790.1830	146 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
asctrls.ocx	6.0.3790.1830	147 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
browsecl.dll	6.0.3790.1830	63 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
browseui.dll	6.0.3790.1830	1,564 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
cdfview.dll	6.0.3790.1830	216 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
comctl32.dll	5.82.3790.1830	935 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
dxtrans.dll	6.3.3790.1830	320 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
dxtmsft.dll	6.3.3790.1830	549 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
iecont.dll	<File Missing>	Not Available	Not Available		
iecontlc.dll	<File Missing>	Not Available	Not Available		
iedkcs32.dll	16.0.3790.1830	417 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
iepeers.dll	6.0.3790.1830	361 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
iesetup.dll	6.0.3790.1830	71 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
ieuinit.inf	Not Available	24 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Not Available
iexplore.exe	6.0.3790.1830	94 KB	3/25/2005 7:00:00 AM	C:\Program Files\Internet Explorer	Microsoft Corporation
imgutil.dll	6.0.3790.1830	61 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
inetctl.cpl	6.0.3790.1830	428 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
inetctl.dll	6.0.3790.1830	110 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
inseng.dll	6.0.3790.1830	147 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
mlang.dll	6.0.3790.1830	686 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
msencode.dll	<File Missing>	Not Available	Not Available		
mshta.exe	6.0.3790.1830	38 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
mshtml.dll	6.0.3790.1830	5,790 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation

mshtml.tlb	6.0.3790.1830	1,320 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
mshtml.dll	6.0.3790.1830	906 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
mshtmlr.dll	6.0.3790.1830	56 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
msident.dll	6.0.3790.1830	69 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
msidntld.dll	6.0.3790.1830	16 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
msieftp.dll	6.0.3790.1830	369 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
msrating.dll	6.0.3790.1830	240 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
mstime.dll	6.0.3790.1830	878 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
occache.dll	6.0.3790.1830	126 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
proctexe.ocx	<File Missing>	Not Available	Not Available		
sendmail.dll	6.0.3790.1830	64 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
shdoclc.dll	6.0.3790.1830	590 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
shdocvw.dll	6.0.3790.1830	2,360 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
shfolder.dll	6.0.3790.1830	34 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
shlwapi.dll	6.0.3790.1830	607 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
tdc.ocx	1.3.0.3130	91 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
url.dll	6.0.3790.1830	40 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
urlmon.dll	6.0.3790.1830	1,049 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
webcheck.dll	6.0.3790.1830	439 KB	3/25/2005 7:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
wininet.dll	6.0.3790.1830	1,159 KB	3/25/2005 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\TPCC\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

February 17, 2006 11:30:06 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.....7.10.53
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.....694300 MB
Free space.....0 MB
Number of logical drives.....1
Number of physical drives.....20

Arrays in spanned array 1

Array identifier.....B
Array size.....347150 MB
Stripe order (channel/device)...1/0 1/1 1/2 1/3 1/4 2/0 2/1 2/2 2/3 2/4
Number of physical drives.....10

Array identifier.....C
Array size.....347150 MB
Stripe order (channel/device)...2/5 2/6 2/8 2/9 2/10 1/5 1/6 1/8 1/9 1/10
Number of physical drives.....10

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.....3HX0PEAA
Firmware level.....B85B
Channel.....1
SCSI ID.....1
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.....3HX0QPDR
Firmware level.....B85B
Channel.....1
SCSI ID.....2
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.....3HX0QMT4
Firmware level.....B85B
Channel.....1
SCSI ID.....3
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.....3HX0QPBF
Firmware level.....B85B
Channel.....1
SCSI ID.....4
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

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.....B
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.....B
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.....B
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.....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.....3HX0MMA5
Firmware level.....B85B
Channel.....2
SCSI ID.....5
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.....3HX0MP10
Firmware level.....B85B
Channel.....2
SCSI ID.....6
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.....3HX0LK0Q

Firmware level.....B85B
Channel.....2
SCSI ID.....8
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.....3HX0MNAD
Firmware level.....B85B
Channel.....2
SCSI ID.....9
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.....3HX0Q4PP
Firmware level.....B85E
Channel.....2
SCSI ID.....10
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.....3HX0R239
Firmware level.....B85B
Channel.....1
SCSI ID.....5
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.....3HX0R1AF
Firmware level.....B85B
Channel.....1
SCSI ID.....6
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.....3HX0R1Q4
Firmware level.....B85B
Channel.....1
SCSI ID.....8

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.....3HX0R1E0
Firmware level.....B85B
Channel.....1
SCSI ID.....9
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.....3HX0MK6F
Firmware level.....B85B
Channel.....1
SCSI ID.....10
Size.....34715 MB
State.....Online
Array letter.....C
PFA error.....No

Logical drives in spanned array 1

Logical drive.....2
Spanned array number1
State.....Okay
RAID level.....1E0
Data space.....347150 MB
Parity space.....347150 MB
Date created.....02/13/2006
Write-cache mode.....Write through
Merge-group number.....207
Merge-group state.....Non-shared

Array A

Array identifier.....A
Array size.....69430 MB
Free space.....0 MB
Number of logical drives.....1
Stripe order (channel/device)...1/14 2/14
Number of physical drives.....2

Logical drives in array A

Logical drive.....1
Array letter.....A
State.....Okay
RAID level.....1
Data space.....34715 MB
Parity space.....34715 MB
Date created.....09/06/2005
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.....3HX0MMAX
Firmware level.....B85B
Channel.....1
SCSI ID.....14
Size.....34715 MB
State.....Online
Array letter.....A
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0MHNX
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.....11
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.....B
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.....B
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.....B
PFA error.....No

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.....B
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.....C
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.....C
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0R1Q4
Firmware level.....B85B
Channel.....1
SCSI ID.....8
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.....3HX0R1E0
Firmware level.....B85B
Channel.....1
SCSI ID.....9
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.....3HX0MK6F
Firmware level.....B85B
Channel.....1
SCSI ID.....10
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.....3HX0MMAX
Firmware level.....B85B
Channel.....1
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.....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.....1R043373198
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.....11
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.....3HX0QMSB
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.....B
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.....B
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.....B
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.....B
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.....C
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.....C
PFA error.....No

Type.....Hard disk drive
Vendor.....IBM-ESXS
Product or model number.....ST336753
FRU part number.....32P0736
Serial number.....3HX0LK0Q
Firmware level.....B85B
Channel.....2
SCSI ID.....8
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.....3HX0MNAD
Firmware level.....B85B
Channel.....2

SCSI ID.....9
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.....3HX0Q4PP
Firmware level.....B85E
Channel.....2
SCSI ID.....10
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.....3HX0MHNX
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

The configuration for Racks 2-4 was identical to Rack 1.

PROFILE FOR STORAGE SUBSYSTEM: Rack_1 (2/17/06 11:37:39 PM)

SUMMARY-----

Number of controllers: 2
Number of arrays: 10
Total number of logical drives (includes an access logical drive): 11 of 2048 used

Number of standard logical drives: 10
Number of access logical drives: 1
Number of drives: 140
Supported drive types: Fibre (140)
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: 06.12.16.00
NVS RAM version: N1742F900R912V06
Pending configuration
Staged firmware download supported?: Yes
Firmware version: None
NVS RAM version: None
Transferred on: None
NVS RAM configured for batteries?: Yes
Start cache flushing at (in percentage): 8
Stop cache flushing at (in percentage): 8
Cache block size (in KB): 16
Media scan frequency (in days): 30
Failover alert delay (in minutes): 5
Feature enable identifier: 30353134300030353131310041B429DB
Storage Subsystem worldwide name (ID):
600A0B800013C6EA0000000043E9FC06

CONTROLLERS-----

Number of controllers: 2

Controller in Slot A

Status: Online

Current configuration

Firmware version: 06.12.16.00
Appware version: 06.12.16.00
Bootware version: 06.10.06.00
NVS RAM version: N1742F900R912V06

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: 1T41105140
Date of manufacture: March 25, 2004
Cache/processor size (MB): 1024/128
Date/Time: Fri Feb 17 23:35:22 EST 2006
Associated Logical Drives (* = Preferred Owner):
Rack1_0*, Rack1_2*, Rack1_4*, Rack1_6*, Rack1_8*
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: 0/0xEF
Preferred ID: 0/0xEF
NL-Port ID: 0x0000EF
Maximum data rate: 2 Gbps
Current data rate: 2 Gbps
Data rate control: Switch
Link status: Up
Topology: Arbitrated Loop - Private
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: 06.12.16.00
Appware version: 06.12.16.00
Bootware version: 06.10.06.00
NVSRAM version: N1742F900R912V06

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: Fri Feb 17 23:35:17 EST 2006

Associated Logical Drives (* = Preferred Owner):

Rack1_1*, Rack1_3*, Rack1_5*, Rack1_7*, Rack1_9*

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: 2/0xE4

Preferred ID: 2/0xE4

NL-Port ID: 0x0000E4

Maximum data rate: 2 Gbps

Current data rate: 2 Gbps

Data rate control: Switch

Link status: Up

Topology: Arbitrated Loop - Private

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: 10

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 (467.619 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

Array 2 (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_1 (467.619 GB)

Associated drives (in piece order):

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 3 (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_2 (467.619 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
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 (467.619 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
Array 5 (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_4 (467.619 GB)
Associated drives (in piece order):
Drive at Enclosure 14, Slot 1
Drive at Enclosure 14, Slot 2
Drive at Enclosure 14, Slot 3
Drive at Enclosure 14, Slot 4
Drive at Enclosure 14, Slot 5
Drive at Enclosure 14, Slot 6
Drive at Enclosure 14, Slot 7
Drive at Enclosure 14, Slot 8
Drive at Enclosure 14, Slot 9
Drive at Enclosure 14, Slot 10
Drive at Enclosure 14, Slot 11
Drive at Enclosure 14, Slot 12
Drive at Enclosure 14, Slot 13
Drive at Enclosure 14, Slot 14
Array 6 (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_5 (467.619 GB)
Associated drives (in piece order):
Drive at Enclosure 20, Slot 1
Drive at Enclosure 20, Slot 2
Drive at Enclosure 20, Slot 3
Drive at Enclosure 20, Slot 4
Drive at Enclosure 20, Slot 5
Drive at Enclosure 20, Slot 6
Drive at Enclosure 20, Slot 7
Drive at Enclosure 20, Slot 8
Drive at Enclosure 20, Slot 9
Drive at Enclosure 20, Slot 10
Drive at Enclosure 20, Slot 11
Drive at Enclosure 20, Slot 12
Drive at Enclosure 20, Slot 13
Drive at Enclosure 20, Slot 14
Array 7 (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_6 (467.619 GB)
Associated drives (in piece order):
Drive at Enclosure 21, Slot 1
Drive at Enclosure 21, Slot 2
Drive at Enclosure 21, Slot 3
Drive at Enclosure 21, Slot 4
Drive at Enclosure 21, Slot 5
Drive at Enclosure 21, Slot 6
Drive at Enclosure 21, Slot 7
Drive at Enclosure 21, Slot 8
Drive at Enclosure 21, Slot 9
Drive at Enclosure 21, Slot 10
Drive at Enclosure 21, Slot 11
Drive at Enclosure 21, Slot 12
Drive at Enclosure 21, Slot 13
Drive at Enclosure 21, Slot 14
Array 8 (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_7 (467.619 GB)
Associated drives (in piece order):
Drive at Enclosure 22, Slot 1
Drive at Enclosure 22, Slot 2
Drive at Enclosure 22, Slot 3
Drive at Enclosure 22, Slot 4
Drive at Enclosure 22, Slot 5
Drive at Enclosure 22, Slot 6
Drive at Enclosure 22, Slot 7
Drive at Enclosure 22, Slot 8
Drive at Enclosure 22, Slot 9
Drive at Enclosure 22, Slot 10
Drive at Enclosure 22, Slot 11
Drive at Enclosure 22, Slot 12
Drive at Enclosure 22, Slot 13
Drive at Enclosure 22, Slot 14
Array 9 (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_8 (467.619 GB)
 Associated drives (in piece order):
 Drive at Enclosure 23, Slot 1
 Drive at Enclosure 23, Slot 2
 Drive at Enclosure 23, Slot 3
 Drive at Enclosure 23, Slot 4
 Drive at Enclosure 23, Slot 5
 Drive at Enclosure 23, Slot 6
 Drive at Enclosure 23, Slot 7
 Drive at Enclosure 23, Slot 8
 Drive at Enclosure 23, Slot 9
 Drive at Enclosure 23, Slot 10
 Drive at Enclosure 23, Slot 11
 Drive at Enclosure 23, Slot 12
 Drive at Enclosure 23, Slot 13
 Drive at Enclosure 23, Slot 14

Array 10 (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_9 (467.619 GB)

Associated drives (in piece order):
 Drive at Enclosure 24, Slot 1
 Drive at Enclosure 24, Slot 2
 Drive at Enclosure 24, Slot 3
 Drive at Enclosure 24, Slot 4
 Drive at Enclosure 24, Slot 5
 Drive at Enclosure 24, Slot 6
 Drive at Enclosure 24, Slot 7
 Drive at Enclosure 24, Slot 8
 Drive at Enclosure 24, Slot 9
 Drive at Enclosure 24, Slot 10
 Drive at Enclosure 24, Slot 11
 Drive at Enclosure 24, Slot 12
 Drive at Enclosure 24, Slot 13
 Drive at Enclosure 24, Slot 14

STANDARD LOGICAL DRIVES-----

SUMMARY

Number of standard logical drives: 10
 See other Logical Drives sub-tabs for premium feature information.

NAME	STATUS	CAPACITY	RAID LEVEL	ARRAY
Rack1_0	Optimal	467.619 GB	0	1
Rack1_1	Optimal	467.619 GB	0	2
Rack1_2	Optimal	467.619 GB	0	3
Rack1_3	Optimal	467.619 GB	0	4
Rack1_4	Optimal	467.619 GB	0	5
Rack1_5	Optimal	467.619 GB	0	6
Rack1_6	Optimal	467.619 GB	0	7
Rack1_7	Optimal	467.619 GB	0	8
Rack1_8	Optimal	467.619 GB	0	9
Rack1_9	Optimal	467.619 GB	0	10

DETAILS

Logical Drive name: Rack1_0
 Logical Drive ID: 60:0a:0b:80:00:13:c6:ea:00:00:00:0f:43:e7:18:4c
 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: 467.619 GB
 RAID level: 0
 Segment size: 64 KB
 Modification priority: High
 Associated array: 1
 Read cache: Enabled
 Write cache: Enabled
 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:00:00:00:11:43:e7:18:8e
 Subsystem ID (SSID): 1
 Status: Optimal
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Preferred owner: Controller in slot B
 Current owner: Controller in slot B
 Capacity: 467.619 GB
 RAID level: 0
 Segment size: 64 KB
 Modification priority: High
 Associated array: 2
 Read cache: Enabled
 Write cache: Enabled
 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:c6:ea:00:00:00:13:43:e7:18:c4
 Subsystem ID (SSID): 2
 Status: Optimal
 Drive type: Fibre Channel
 Enclosure loss protection: No
 Preferred owner: Controller in slot A
 Current owner: Controller in slot A
 Capacity: 467.619 GB
 RAID level: 0
 Segment size: 64 KB
 Modification priority: High
 Associated array: 3
 Read cache: Enabled
 Write cache: Enabled
 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:c6:ea:00:00:00:15:43:e7:19:12
 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: 467.619 GB

RAID level: 0
Segment size: 64 KB
Modification priority: High
Associated array: 4
Read cache: Enabled
Write cache: Enabled
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_4
Logical Drive ID: 60:0a:0b:80:00:13:c6:ea:00:00:00:17:43:e7:19:56
Subsystem ID (SSID): 4
Status: Optimal
Drive type: Fibre Channel
Enclosure loss protection: No
Preferred owner: Controller in slot A
Current owner: Controller in slot A
Capacity: 467.619 GB
RAID level: 0
Segment size: 64 KB
Modification priority: High
Associated array: 5
Read cache: Enabled
Write cache: Enabled
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_5
Logical Drive ID: 60:0a:0b:80:00:13:c5:fa:00:00:00:0e:43:e7:18:2f
Subsystem ID (SSID): 5
Status: Optimal
Drive type: Fibre Channel
Enclosure loss protection: No
Preferred owner: Controller in slot B
Current owner: Controller in slot B
Capacity: 467.619 GB
RAID level: 0
Segment size: 64 KB
Modification priority: High
Associated array: 6
Read cache: Enabled
Write cache: Enabled
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_6
Logical Drive ID: 60:0a:0b:80:00:13:c5:fa:00:00:00:10:43:e7:18:5f
Subsystem ID (SSID): 6
Status: Optimal
Drive type: Fibre Channel
Enclosure loss protection: No
Preferred owner: Controller in slot A
Current owner: Controller in slot A
Capacity: 467.619 GB
RAID level: 0
Segment size: 64 KB

Modification priority: High
Associated array: 7
Read cache: Enabled
Write cache: Enabled
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_7
Logical Drive ID: 60:0a:0b:80:00:13:c5:fa:00:00:00:12:43:e7:18:a5
Subsystem ID (SSID): 7
Status: Optimal
Drive type: Fibre Channel
Enclosure loss protection: No
Preferred owner: Controller in slot B
Current owner: Controller in slot B
Capacity: 467.619 GB
RAID level: 0
Segment size: 64 KB
Modification priority: High
Associated array: 8
Read cache: Enabled
Write cache: Enabled
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_8
Logical Drive ID: 60:0a:0b:80:00:13:c5:fa:00:00:00:14:43:e7:18:d7
Subsystem ID (SSID): 8
Status: Optimal
Drive type: Fibre Channel
Enclosure loss protection: No
Preferred owner: Controller in slot A
Current owner: Controller in slot A
Capacity: 467.619 GB
RAID level: 0
Segment size: 64 KB
Modification priority: High
Associated array: 9
Read cache: Enabled
Write cache: Enabled
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_9
Logical Drive ID: 60:0a:0b:80:00:13:c5:fa:00:00:00:16:43:e7:19:0b
Subsystem ID (SSID): 9
Status: Optimal
Drive type: Fibre Channel
Enclosure loss protection: No
Preferred owner: Controller in slot B
Current owner: Controller in slot B
Capacity: 467.619 GB
RAID level: 0
Segment size: 64 KB
Modification priority: High
Associated array: 10

Read cache: Enabled
 Write cache: Enabled
 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: 140
 Supported drive types: Fibre (140)

BASIC:

TRAY, SLOT	STATUS	CAPACITY	CURRENT DATA RATE
PRODUCT ID	FIRMWARE	VERSION	
10, 1	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
10, 2	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
10, 3	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
10, 4	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
10, 5	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
10, 6	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
10, 7	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
10, 8	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
10, 9	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
10, 10	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
10, 11	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
10, 12	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
10, 13	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
10, 14	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
11, 1	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
11, 2	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
11, 3	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
11, 4	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
11, 5	Optimal	33.902 GB 2 Gbps	ST336753FC F B954
11, 6	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
11, 7	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
11, 8	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B

11, 9	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
11, 10	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
11, 11	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
11, 12	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
11, 13	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
11, 14	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 1	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 2	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 3	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 4	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 5	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 6	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 7	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 8	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 9	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 10	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 11	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 12	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 13	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
12, 14	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
13, 1	Optimal	33.902 GB 2 Gbps	ST336753FC F B954
13, 2	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
13, 3	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
13, 4	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
13, 5	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
13, 6	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
13, 7	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
13, 8	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
13, 9	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
13, 10	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
13, 11	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
13, 12	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
13, 13	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B
13, 14	Optimal	33.902 GB 2 Gbps	ST336753FC F B95B

23, 13	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
23, 14	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 1	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 2	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 3	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 4	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 5	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 6	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 7	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 8	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 9	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 10	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 11	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 12	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 13	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B
24, 14	Optimal	33.902 GB	2 Gbps	ST336753FC	F	B95B

DRIVE CHANNELS:

TRAY, SLOT PREFERRED CHANNEL REDUNDANT CHANNEL

10, 1	3	1
10, 2	1	3
10, 3	3	1
10, 4	1	3
10, 5	3	1
10, 6	1	3
10, 7	3	1
10, 8	1	3
10, 9	3	1
10, 10	1	3
10, 11	3	1
10, 12	1	3
10, 13	3	1
10, 14	1	3
11, 1	3	1
11, 2	1	3
11, 3	3	1
11, 4	1	3
11, 5	3	1
11, 6	1	3
11, 7	3	1
11, 8	1	3
11, 9	3	1
11, 10	1	3
11, 11	3	1
11, 12	1	3
11, 13	3	1
11, 14	1	3
12, 1	3	1
12, 2	1	3
12, 3	3	1
12, 4	1	3

12, 5	3	1
12, 6	1	3
12, 7	3	1
12, 8	1	3
12, 9	3	1
12, 10	1	3
12, 11	3	1
12, 12	1	3
12, 13	3	1
12, 14	1	3
13, 1	3	1
13, 2	1	3
13, 3	3	1
13, 4	1	3
13, 5	3	1
13, 6	1	3
13, 7	3	1
13, 8	1	3
13, 9	3	1
13, 10	1	3
13, 11	3	1
13, 12	1	3
13, 13	3	1
13, 14	1	3
14, 1	3	1
14, 2	1	3
14, 3	3	1
14, 4	1	3
14, 5	3	1
14, 6	1	3
14, 7	3	1
14, 8	1	3
14, 9	3	1
14, 10	1	3
14, 11	3	1
14, 12	1	3
14, 13	3	1
14, 14	1	3
20, 1	4	2
20, 2	2	4
20, 3	4	2
20, 4	2	4
20, 5	4	2
20, 6	2	4
20, 7	4	2
20, 8	2	4
20, 9	4	2
20, 10	2	4
20, 11	4	2
20, 12	2	4
20, 13	4	2
20, 14	2	4
21, 1	4	2
21, 2	2	4
21, 3	4	2
21, 4	2	4
21, 5	4	2
21, 6	2	4
21, 7	4	2
21, 8	2	4
21, 9	4	2
21, 10	2	4
21, 11	4	2
21, 12	2	4
21, 13	4	2
21, 14	2	4
22, 1	4	2
22, 2	2	4

22, 3	4	2
22, 4	2	4
22, 5	4	2
22, 6	2	4
22, 7	4	2
22, 8	2	4
22, 9	4	2
22, 10	2	4
22, 11	4	2
22, 12	2	4
22, 13	4	2
22, 14	2	4
23, 1	4	2
23, 2	2	4
23, 3	4	2
23, 4	2	4
23, 5	4	2
23, 6	2	4
23, 7	4	2
23, 8	2	4
23, 9	4	2
23, 10	2	4
23, 11	4	2
23, 12	2	4
23, 13	4	2
23, 14	2	4
24, 1	4	2
24, 2	2	4
24, 3	4	2
24, 4	2	4
24, 5	4	2
24, 6	2	4
24, 7	4	2
24, 8	2	4
24, 9	4	2
24, 10	2	4
24, 11	4	2
24, 12	2	4
24, 13	4	2
24, 14	2	4

HOT SPARE COVERAGE:

The following arrays are not protected: 2, 6, 9, 3, 5, 10, 4, 8, 1, 7

Total hot spare drives: 0
 Standby: 0
 In use: 0

DETAILS

Drive at Enclosure 10, Slot 1
 Drive port: 1, Channel: 3, 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: B95B
 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: 1

Drive at Enclosure 10, Slot 2
 Drive port: 1, Channel: 1, ID: 1/0xE8
 Drive port: 2, Channel: 3, 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: B95B
 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: 1

Drive at Enclosure 10, Slot 3
 Drive port: 1, Channel: 3, 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: B95B
 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: 1

Drive at Enclosure 10, Slot 4
 Drive port: 1, Channel: 1, ID: 3/0xE2
 Drive port: 2, Channel: 3, 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: B95B
 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: 1

Drive at Enclosure 10, Slot 5
 Drive port: 1, Channel: 3, 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: B95B
 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: 1

Drive at Enclosure 10, Slot 6

Drive port: 1, Channel: 1, ID: 5/0xE0
Drive port: 2, Channel: 3, 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: B95B
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: 1

Drive at Enclosure 10, Slot 7

Drive port: 1, Channel: 3, 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: B95B
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: 1

Drive at Enclosure 10, Slot 8

Drive port: 1, Channel: 1, ID: 7/0xDA
Drive port: 2, Channel: 3, 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: B95B
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: 1

Drive at Enclosure 10, Slot 9

Drive port: 1, Channel: 3, 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: B95B
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: 1

Drive at Enclosure 10, Slot 10

Drive port: 1, Channel: 1, ID: 96/0x3A
Drive port: 2, Channel: 3, 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: B95B
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: 1

Drive at Enclosure 10, Slot 11

Drive port: 1, Channel: 3, 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: B95B
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: 1

Drive at Enclosure 10, Slot 12

Drive port: 1, Channel: 1, ID: 72/0x67
Drive port: 2, Channel: 3, 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: B95B
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: 1

Drive at Enclosure 10, Slot 13

Drive port: 1, Channel: 3, 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: B95B
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: 1

Drive at Enclosure 10, Slot 14

Drive port: 1, Channel: 1, ID: 104/0x2E
Drive port: 2, Channel: 3, 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: B95B
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: 1

Drive at Enclosure 11, Slot 1

Drive port: 1, Channel: 3, 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: B95B
Serial number: 3HX2DVH700007447LVPR
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: 2

Drive at Enclosure 11, Slot 2

Drive port: 1, Channel: 1, ID: 9/0xD6
Drive port: 2, Channel: 3, 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: B95B
Serial number: 3HX2NEJT000075029SZD
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: 2

Drive at Enclosure 11, Slot 3

Drive port: 1, Channel: 3, 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: B95B
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: 2

Drive at Enclosure 11, Slot 4

Drive port: 1, Channel: 1, ID: 11/0xD4
Drive port: 2, Channel: 3, 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: B95B
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: 2

Drive at Enclosure 11, Slot 5

Drive port: 1, Channel: 3, 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: 3HX29CZD00007436FR6R
Vendor: IBM-ESXS
Date of manufacture: May 10, 2005
World-wide name: 20:00:00:04:cf:f4:b3:3c
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned

Associated array: 2

Drive at Enclosure 11, Slot 6

Drive port: 1, Channel: 1, ID: 13/0xD2
Drive port: 2, Channel: 3, 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: B95B
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: 2

Drive at Enclosure 11, Slot 7

Drive port: 1, Channel: 3, 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: B95B
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: 2

Drive at Enclosure 11, Slot 8

Drive port: 1, Channel: 1, ID: 15/0xCE
Drive port: 2, Channel: 3, 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: B95B
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: 2

Drive at Enclosure 11, Slot 9

Drive port: 1, Channel: 3, 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: B95B
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: 2

Drive at Enclosure 11, Slot 10

Drive port: 1, Channel: 1, ID: 97/0x39
Drive port: 2, Channel: 3, 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: B95B
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: 2

Drive at Enclosure 11, Slot 11

Drive port: 1, Channel: 3, 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: B95B
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: 2

Drive at Enclosure 11, Slot 12

Drive port: 1, Channel: 1, ID: 73/0x66
Drive port: 2, Channel: 3, 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: B95B
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: 2

Drive at Enclosure 11, Slot 13

Drive port: 1, Channel: 3, 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: B95B
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: 2

Drive at Enclosure 11, Slot 14

Drive port: 1, Channel: 1, ID: 105/0x2D
Drive port: 2, Channel: 3, 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: B95B
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: 2

Drive at Enclosure 12, Slot 1

Drive port: 1, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 12, Slot 2

Drive port: 1, Channel: 1, ID: 17/0xCC
Drive port: 2, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 12, Slot 3

Drive port: 1, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 12, Slot 4

Drive port: 1, Channel: 1, ID: 19/0xCA
Drive port: 2, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 12, Slot 5

Drive port: 1, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 12, Slot 6

Drive port: 1, Channel: 1, ID: 21/0xC7
Drive port: 2, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 12, Slot 7
Drive port: 1, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 12, Slot 8
Drive port: 1, Channel: 1, ID: 23/0xC5
Drive port: 2, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 12, Slot 9
Drive port: 1, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 12, Slot 10
Drive port: 1, Channel: 1, ID: 98/0x36
Drive port: 2, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 12, Slot 11
Drive port: 1, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 12, Slot 12
Drive port: 1, Channel: 1, ID: 74/0x65
Drive port: 2, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 12, Slot 13
Drive port: 1, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 12, Slot 14

Drive port: 1, Channel: 1, ID: 106/0x2C
Drive port: 2, Channel: 3, 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: B95B
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: 3

Drive at Enclosure 13, Slot 1

Drive port: 1, Channel: 3, 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: 3HX2N4ZL00007502CWB6
Vendor: IBM-ESXS
Date of manufacture: July 20, 2004
World-wide name: 20:00:00:0c:50:45:ab:0e
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 2

Drive port: 1, Channel: 1, ID: 25/0xB3
Drive port: 2, Channel: 3, ID: 25/0xB3
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: B95B
Serial number: 3HX2NEV900007502CWZB
Vendor: IBM-ESXS
Date of manufacture: July 21, 2004
World-wide name: 20:00:00:0c:50:45:ac:fb
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 4

Drive at Enclosure 13, Slot 3

Drive port: 1, Channel: 3, 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: B95B
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: 3, 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: B95B
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: 3, 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: B95B
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: 3, 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: B95B

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: 3, 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: B95B
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: 3, 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: B95B
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: 3, 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: B95B
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: 3, 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: B95B
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: 3, 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: B95B
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: 3, 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: B95B
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: 3, 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: B95B
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: 3, 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: B95B
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

Drive at Enclosure 14, Slot 1

Drive port: 1, Channel: 3, ID: 32/0xB2
Drive port: 2, Channel: 1, ID: 32/0xB2
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 14, Slot 2

Drive port: 1, Channel: 1, ID: 33/0xB1
Drive port: 2, Channel: 3, ID: 33/0xB1
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 14, Slot 3

Drive port: 1, Channel: 3, ID: 34/0xAE
Drive port: 2, Channel: 1, ID: 34/0xAE
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 14, Slot 4

Drive port: 1, Channel: 1, ID: 35/0xAD
Drive port: 2, Channel: 3, ID: 35/0xAD
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 14, Slot 5

Drive port: 1, Channel: 3, ID: 36/0xAC
Drive port: 2, Channel: 1, ID: 36/0xAC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 14, Slot 6

Drive port: 1, Channel: 1, ID: 37/0xAB
Drive port: 2, Channel: 3, ID: 37/0xAB
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 14, Slot 7

Drive port: 1, Channel: 3, ID: 38/0xAA
Drive port: 2, Channel: 1, ID: 38/0xAA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 14, Slot 8

Drive port: 1, Channel: 1, ID: 39/0xA9
Drive port: 2, Channel: 3, ID: 39/0xA9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 14, Slot 9

Drive port: 1, Channel: 3, ID: 84/0x51
Drive port: 2, Channel: 1, ID: 84/0x51
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 14, Slot 10

Drive port: 1, Channel: 1, ID: 100/0x34
Drive port: 2, Channel: 3, ID: 100/0x34
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps

Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 14, Slot 11

Drive port: 1, Channel: 3, ID: 68/0x6C
Drive port: 2, Channel: 1, ID: 68/0x6C
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 14, Slot 12

Drive port: 1, Channel: 1, ID: 76/0x5C
Drive port: 2, Channel: 3, ID: 76/0x5C
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 14, Slot 13

Drive port: 1, Channel: 3, ID: 92/0x46
Drive port: 2, Channel: 1, ID: 92/0x46
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 14, Slot 14

Drive port: 1, Channel: 1, ID: 108/0x2A
Drive port: 2, Channel: 3, ID: 108/0x2A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 5

Drive at Enclosure 20, Slot 1

Drive port: 1, Channel: 4, ID: 0/0xEF
Drive port: 2, Channel: 2, 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: B95B
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: 6

Drive at Enclosure 20, Slot 2

Drive port: 1, Channel: 2, 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: B95B
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: 6

Drive at Enclosure 20, Slot 3

Drive port: 1, Channel: 4, ID: 2/0xE4
Drive port: 2, Channel: 2, 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: B95B
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: 6

Drive at Enclosure 20, Slot 4

Drive port: 1, Channel: 2, 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: B95B
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: 6

Drive at Enclosure 20, Slot 5

Drive port: 1, Channel: 4, ID: 4/0xE1
Drive port: 2, Channel: 2, 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: B95B
Serial number: 3HX24C2V00007433FR1S
Vendor: IBM-ESXS
Date of manufacture: February 27, 2004
World-wide name: 20:00:00:0c:50:b6:64:63
Drive type: Fibre Channel
Speed: 15015 RPM
Mode: Assigned
Associated array: 6

Drive at Enclosure 20, Slot 6

Drive port: 1, Channel: 2, 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: B95B
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: 6

Drive at Enclosure 20, Slot 7

Drive port: 1, Channel: 4, ID: 6/0xDC
Drive port: 2, Channel: 2, 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: B95B
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: 6

Drive at Enclosure 20, Slot 8
Drive port: 1, Channel: 2, 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: B95B
Serial number: 3HX2DED0000744780SZ
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: 6

Drive at Enclosure 20, Slot 9
Drive port: 1, Channel: 4, ID: 80/0x55
Drive port: 2, Channel: 2, 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: B95B
Serial number: 3HX2MKSK000075029SWZ
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: 6

Drive at Enclosure 20, Slot 10
Drive port: 1, Channel: 2, 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: B95B
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: 6

Drive at Enclosure 20, Slot 11
Drive port: 1, Channel: 4, ID: 64/0x72
Drive port: 2, Channel: 2, 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: B95B
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: 6

Drive at Enclosure 20, Slot 12
Drive port: 1, Channel: 2, 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: B95B
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: 6

Drive at Enclosure 20, Slot 13
Drive port: 1, Channel: 4, ID: 88/0x4B
Drive port: 2, Channel: 2, 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: B95B
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: 6

Drive at Enclosure 20, Slot 14
Drive port: 1, Channel: 2, 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: B95B
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: 6

Drive at Enclosure 21, Slot 1
Drive port: 1, Channel: 4, ID: 8/0xD9
Drive port: 2, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 21, Slot 2
Drive port: 1, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 21, Slot 3
Drive port: 1, Channel: 4, ID: 10/0xD5
Drive port: 2, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 21, Slot 4
Drive port: 1, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 21, Slot 5
Drive port: 1, Channel: 4, ID: 12/0xD3
Drive port: 2, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 21, Slot 6
Drive port: 1, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 21, Slot 7
Drive port: 1, Channel: 4, ID: 14/0xD1
Drive port: 2, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 21, Slot 8

Drive port: 1, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 21, Slot 9

Drive port: 1, Channel: 4, ID: 81/0x54
Drive port: 2, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 21, Slot 10

Drive port: 1, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 21, Slot 11

Drive port: 1, Channel: 4, ID: 65/0x71
Drive port: 2, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 21, Slot 12

Drive port: 1, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 21, Slot 13

Drive port: 1, Channel: 4, ID: 89/0x4A
Drive port: 2, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 21, Slot 14

Drive port: 1, Channel: 2, 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: B95B
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: 7

Drive at Enclosure 22, Slot 1

Drive port: 1, Channel: 4, ID: 16/0xCD
Drive port: 2, Channel: 2, 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: B95B
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: 8

Drive at Enclosure 22, Slot 2

Drive port: 1, Channel: 2, 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: B95B
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: 8

Drive at Enclosure 22, Slot 3

Drive port: 1, Channel: 4, ID: 18/0xCB
Drive port: 2, Channel: 2, 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: B95B
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: 8

Drive at Enclosure 22, Slot 4

Drive port: 1, Channel: 2, 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: B95B
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: 8

Drive at Enclosure 22, Slot 5

Drive port: 1, Channel: 4, ID: 20/0xC9
Drive port: 2, Channel: 2, 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: B95B
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: 8

Drive at Enclosure 22, Slot 6

Drive port: 1, Channel: 2, 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: B95B
Serial number: 3HX2E18000007447RBGZ
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: 8

Drive at Enclosure 22, Slot 7

Drive port: 1, Channel: 4, ID: 22/0xC6
Drive port: 2, Channel: 2, 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: B95B
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: 8

Drive at Enclosure 22, Slot 8

Drive port: 1, Channel: 2, 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: B95B
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: 8

Drive at Enclosure 22, Slot 9

Drive port: 1, Channel: 4, ID: 82/0x53
Drive port: 2, Channel: 2, 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: B95B
Serial number: 3HX2DNBQ00007447KUUQ
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: 8

Drive at Enclosure 22, Slot 10

Drive port: 1, Channel: 2, 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: B95B
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: 8

Drive at Enclosure 22, Slot 11

Drive port: 1, Channel: 4, ID: 66/0x6E
Drive port: 2, Channel: 2, 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: B95B
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: 8

Drive at Enclosure 22, Slot 12

Drive port: 1, Channel: 2, 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: B95B
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: 8

Drive at Enclosure 22, Slot 13

Drive port: 1, Channel: 4, ID: 90/0x49
Drive port: 2, Channel: 2, 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: B95B
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: 8

Drive at Enclosure 22, Slot 14

Drive port: 1, Channel: 2, 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: B95B
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: 8

Drive at Enclosure 23, Slot 1
Drive port: 1, Channel: 4, ID: 24/0xC3
Drive port: 2, Channel: 2, 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: B95B
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: 9

Drive at Enclosure 23, Slot 2
Drive port: 1, Channel: 2, 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: B95B
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: 9

Drive at Enclosure 23, Slot 3
Drive port: 1, Channel: 4, ID: 26/0xBA
Drive port: 2, Channel: 2, 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: B95B
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: 9

Drive at Enclosure 23, Slot 4
Drive port: 1, Channel: 2, 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: B95B

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: 9

Drive at Enclosure 23, Slot 5
Drive port: 1, Channel: 4, ID: 28/0xB6
Drive port: 2, Channel: 2, 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: B95B
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: 9

Drive at Enclosure 23, Slot 6
Drive port: 1, Channel: 2, 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: B95B
Serial number: 3HX2E9KG00007447QT58
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: 9

Drive at Enclosure 23, Slot 7
Drive port: 1, Channel: 4, ID: 30/0xB4
Drive port: 2, Channel: 2, 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: B95B
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: 9

Drive at Enclosure 23, Slot 8
Drive port: 1, Channel: 2, 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: B95B
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: 9

Drive at Enclosure 23, Slot 9

Drive port: 1, Channel: 4, ID: 83/0x52
Drive port: 2, Channel: 2, 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: B95B
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: 9

Drive at Enclosure 23, Slot 10

Drive port: 1, Channel: 2, 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: B95B
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: 9

Drive at Enclosure 23, Slot 11

Drive port: 1, Channel: 4, ID: 67/0x6D
Drive port: 2, Channel: 2, 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: B95B
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: 9

Drive at Enclosure 23, Slot 12

Drive port: 1, Channel: 2, 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: B95B
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: 9

Drive at Enclosure 23, Slot 13

Drive port: 1, Channel: 4, ID: 91/0x47
Drive port: 2, Channel: 2, 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: B95B
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: 9

Drive at Enclosure 23, Slot 14

Drive port: 1, Channel: 2, 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: B95B
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: 9

Drive at Enclosure 24, Slot 1

Drive port: 1, Channel: 4, ID: 32/0xB2
Drive port: 2, Channel: 2, ID: 32/0xB2
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 10

Drive at Enclosure 24, Slot 2
Drive port: 1, Channel: 2, ID: 33/0xB1
Drive port: 2, Channel: 4, ID: 33/0xB1
Drive path redundancy: OK
Status: Optimal

Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 10

Drive at Enclosure 24, Slot 3
Drive port: 1, Channel: 4, ID: 34/0xAE
Drive port: 2, Channel: 2, ID: 34/0xAE
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 10

Drive at Enclosure 24, Slot 4
Drive port: 1, Channel: 2, ID: 35/0xAD
Drive port: 2, Channel: 4, ID: 35/0xAD
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 10

Drive at Enclosure 24, Slot 5
Drive port: 1, Channel: 4, ID: 36/0xAC
Drive port: 2, Channel: 2, ID: 36/0xAC
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 10

Drive at Enclosure 24, Slot 6
Drive port: 1, Channel: 2, ID: 37/0xAB
Drive port: 2, Channel: 4, ID: 37/0xAB
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 10

Drive at Enclosure 24, Slot 7
Drive port: 1, Channel: 4, ID: 38/0xAA
Drive port: 2, Channel: 2, ID: 38/0xAA
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 10

Drive at Enclosure 24, Slot 8
Drive port: 1, Channel: 2, ID: 39/0xA9
Drive port: 2, Channel: 4, ID: 39/0xA9
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps

Product ID: ST336753FC F
Firmware version: B95B
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: 10

Drive at Enclosure 24, Slot 9
Drive port: 1, Channel: 4, ID: 84/0x51
Drive port: 2, Channel: 2, ID: 84/0x51
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 10

Drive at Enclosure 24, Slot 10
Drive port: 1, Channel: 2, ID: 100/0x34
Drive port: 2, Channel: 4, ID: 100/0x34
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 10

Drive at Enclosure 24, Slot 11
Drive port: 1, Channel: 4, ID: 68/0x6C
Drive port: 2, Channel: 2, ID: 68/0x6C
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 10

Drive at Enclosure 24, Slot 12
Drive port: 1, Channel: 2, ID: 76/0x5C
Drive port: 2, Channel: 4, ID: 76/0x5C
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 10

Drive at Enclosure 24, Slot 13
Drive port: 1, Channel: 4, ID: 92/0x46
Drive port: 2, Channel: 2, ID: 92/0x46
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 10

Drive at Enclosure 24, Slot 14
Drive port: 1, Channel: 2, ID: 108/0x2A
Drive port: 2, Channel: 4, ID: 108/0x2A
Drive path redundancy: OK
Status: Optimal
Raw capacity: 33.902 GB
Usable capacity: 33.402 GB
Current data rate: 2 Gbps
Product ID: ST336753FC F
Firmware version: B95B
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: 10

DRIVE CHANNEL INFORMATION - Channel 1

SUMMARY-----

Channel 1 status: Optimal
Controller A link status: Up
Controller B link status: Up

CUMULATIVE ERROR COUNTS-----

Controller A Cumulative Error Counts
Baseline time set: 2/8/06 9:17:24 AM
Sample period (days, hh:mm:ss): 9 days, 14:18:37

Controller detected errors: 0
 Drive detected errors: 0
 Timeout errors: 0
 Link down errors: 24
 Total I/O count: 334115112
 Controller B Cumulative Error Counts
 Baseline time set: 2/8/06 9:17:24 AM
 Sample period (days, hh:mm:ss): 9 days, 14:18:37
 Controller detected errors: 0
 Drive detected errors: 0
 Timeout errors: 0
 Link down errors: 25
 Total I/O count: 149975388

DRIVE CHANNEL INFORMATION - Channel 2

SUMMARY-----
 Channel 2 status: Optimal
 Controller A link status: Up
 Controller B link status: Up

CUMULATIVE ERROR COUNTS-----
 Controller A Cumulative Error Counts
 Baseline time set: 2/8/06 9:17:27 AM
 Sample period (days, hh:mm:ss): 9 days, 14:18:34
 Controller detected errors: 0
 Drive detected errors: 0
 Timeout errors: 0
 Link down errors: 23
 Total I/O count: 153776550
 Controller B Cumulative Error Counts
 Baseline time set: 2/8/06 9:17:27 AM
 Sample period (days, hh:mm:ss): 9 days, 14:18:34
 Controller detected errors: 0
 Drive detected errors: 0
 Timeout errors: 0
 Link down errors: 23
 Total I/O count: 329096013

DRIVE CHANNEL INFORMATION - Channel 3

SUMMARY-----
 Channel 3 status: Optimal
 Controller A link status: Up
 Controller B link status: Up

CUMULATIVE ERROR COUNTS-----
 Controller A Cumulative Error Counts
 Baseline time set: 2/8/06 9:17:27 AM
 Sample period (days, hh:mm:ss): 9 days, 14:18:34
 Controller detected errors: 0
 Drive detected errors: 0
 Timeout errors: 0
 Link down errors: 10
 Total I/O count: 335253365
 Controller B Cumulative Error Counts
 Baseline time set: 2/8/06 9:17:27 AM
 Sample period (days, hh:mm:ss): 9 days, 14:18:34
 Controller detected errors: 0
 Drive detected errors: 0
 Timeout errors: 0
 Link down errors: 10
 Total I/O count: 151477002

DRIVE CHANNEL INFORMATION - Channel 4

SUMMARY-----
 Channel 4 status: Optimal

Controller A link status: Up
 Controller B link status: Up

CUMULATIVE ERROR COUNTS-----
 Controller A Cumulative Error Counts
 Baseline time set: 2/8/06 9:17:27 AM
 Sample period (days, hh:mm:ss): 9 days, 14:18:34
 Controller detected errors: 0
 Drive detected errors: 0
 Timeout errors: 0
 Link down errors: 15
 Total I/O count: 154142051
 Controller B Cumulative Error Counts
 Baseline time set: 2/8/06 9:17:27 AM
 Sample period (days, hh:mm:ss): 9 days, 14:18:34
 Controller detected errors: 0
 Drive detected errors: 0
 Timeout errors: 0
 Link down errors: 15
 Total I/O count: 331319455

Client Configuration

Microsoft Windows 2000 Client System Information Report

Following is the system information report for Client 1. The system information report for clients 2-8 are identical except for the system name and network addresses.

System Information report written at: 02/17/2006 023:28:30 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	VCLIENT150
System Manufacturer	IBM
System Model	IBM eServer x226-[86482BU]-
System Type	X86-based PC
Processor	x86 Family 15 Model 4 Stepping 3 GenuineIntel ~3400 Mhz
Processor	x86 Family 15 Model 4 Stepping 3 GenuineIntel ~3400 Mhz
Processor	x86 Family 15 Model 4 Stepping 3 GenuineIntel ~3400 Mhz
Processor	x86 Family 15 Model 4 Stepping 3 GenuineIntel ~3400 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	VCLIENT150\Administrator
Time Zone	Eastern Daylight Time
Total Physical Memory	2,620,316 KB
Available Physical Memory	2,351,840 KB
Total Virtual Memory	7,176,064 KB
Available Virtual Memory	6,769,176 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 - 3596	Intel(R) E7525/E7520/E7320 PCI Express Root Port A1	OK
0x2000-0x4FFF OK	Intel(R) 6700PXH PCI Express-to-PCI Bridge A - 0329	OK
0x2000-0x4FFF	Adaptec AIC-7902B - Ultra320 SCSI	OK
0x3000-0x303F OK	Intel(R) PRO/1000 MT Dual Port Server Adapter	OK
0x3040-0x307F OK	Intel(R) PRO/1000 MT Dual Port Server Adapter #2	OK
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 OK	Intel(R) 6700PXH PCI Express-to-PCI Bridge B - 032A	OK
0x4000-0x4FFF OK	Intel(R) PRO/1000 MT Dual Port Server Adapter #5	OK
0x4040-0x407F OK	Intel(R) PRO/1000 MT Dual Port Server Adapter #6	OK
0x1400-0x141F OK	Intel(R) 82801EB USB Universal Host Controller - 24D2	OK
0x1420-0x143F OK	Intel(R) 82801EB USB Universal Host Controller - 24D4	OK
0x1440-0x145F OK	Intel(R) 82801EB USB Universal Host Controller - 24D7	OK
0x1460-0x147F OK	Intel(R) 82801EB USB Universal Host Controller - 24DE	OK
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
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
0xA000000-0xFEBFFFF	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
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
0xFEBFC00-0xFEBFFFFF	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	Manufacturer	Description	Status	File
Version	Size	Creation Date		
c:\winnt\system32\iac25_32.ax	Intel Corporation	Indeo@ audio software		
OK	C:\WINNT\system32\IAC25_32.AX	2.05.53	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\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\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\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
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\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

[Video Codecs]

Codec	Manufacturer	Description	Status	File
Version	Size	Creation Date		
c:\winnt\system32\ir50_32.dll	Intel Corporation	Indeo@ video 5.10		
OK	C:\WINNT\system32\IR50_32.DLL	R.5.10.15.2.55		737.50 KB (755,200 bytes) 12/7/1999 7:00:00 AM
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\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\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\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\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\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_00\4&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 65536
Resolution 1024 x 768 x 75 hertz
Bits/Pixel 16

[Infrared]

Item Value
No infrared devices

[Input]

[Following are sub-categories of this main category]

[Keyboard]

Item Value
Description Standard 101/102-Key or Microsoft Natural PS/2
Keyboard
Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID ACPI\PNP0303\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 10/21/2005 6:46:16 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 10/21/2005 6:46:16 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 10/21/2005 6:46:16 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_PTIMINIPOINT\0000
Last Reset 10/21/2005 6:46:16 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 10/21/2005 6:46:16 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 10/21/2005 6:46:16 AM

Index 5
 Service Name E1000
 IP Address 192.168.152.99
 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:35:7F:1C
 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 10/21/2005 6:46:16 AM

Index 6
 Service Name E1000
 IP Address 192.168.153.99
 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:35:7F:1D
 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 [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 10/21/2005 6:46:16 AM

Index 9
 Service Name E1000
 IP Address 192.168.150.99
 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:36:7E:CA
 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 10/21/2005 6:46:16 AM

Index 10
 Service Name E1000
 IP Address 192.168.151.99
 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:36:7E:CB
 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 10/21/2005 6:46:16 AM

Index 11
 Service Name b57w2k
 IP Address 192.168.122.150
 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:60:15:1B:37
 Service Name b57w2k
 IRQ Number 16
 Driver c:\winnt\system32\drivers\b57w2k.sys (192215, 7.80.0.0)

[Protocol]

Item Value
 Name MSAFD Tcpi [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 Tcpi [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-A9F0-4BCEDDEDBF8A3}]
 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_{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	28.12 GB (30,196,408,320 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 ST336607LC FN 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	False
abp480n5	abp480n5	Not Available	Kernel Driver	False	False
Disabled	Stopped	OK	Normal	False	False
acpi	Microsoft ACPI Driver		c:\winnt\system32\drivers\acpi.sys	Running	OK
Kernel Driver	True	Boot	Running	OK	Normal
False	True				
acpiec	ACPIEC	c:\winnt\system32\drivers\acpiec.sys	Kernel	False	False
Driver	False	Disabled	Stopped	OK	Normal
False					
adpu160m	adpu160m	Not Available	Kernel Driver	False	False
Disabled	Stopped	OK	Normal	False	False
adpu320	adpu320	c:\winnt\system32\drivers\adpu320.sys	Kernel	Running	OK
Driver	True	Boot	Running	OK	Normal
True					
afd	AFD Networking Support Environment		Kernel Driver	True	Auto
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	False
aic116x	aic116x	Not Available	Kernel Driver	False	False
Disabled	Stopped	OK	Normal	False	False
aic78u2	aic78u2	Not Available	Kernel Driver	False	False
Disabled	Stopped	OK	Normal	False	False
aic78xx	aic78xx	Not Available	Kernel Driver	False	False
Disabled	Stopped	OK	Normal	False	False
ami0nt	ami0nt	Not Available	Kernel Driver	False	False
Disabled	Stopped	OK	Normal	False	False
amsint	amsint	Not Available	Kernel Driver	False	False
Disabled	Stopped	OK	Normal	False	False
asc	asc	Not Available	Kernel Driver	False	False
Disabled	Stopped	OK	Normal	False	False
asc3350p	asc3350p	Not Available	Kernel Driver	False	False
Disabled	Stopped	OK	Normal	False	False
asc3550	asc3550	Not Available	Kernel Driver	False	False
Disabled	Stopped	OK	Normal	False	False
asynmac	RAS Asynchronous Media Driver		Kernel Driver	False	False
c:\winnt\system32\drivers\asynmac.sys			Kernel Driver	False	False
Manual	Stopped	OK	Normal	False	False
atapi	Standard IDE/ESDI Hard Disk Controller		Kernel Driver	True	True
c:\winnt\system32\drivers\atapi.sys			Kernel Driver	True	True
Boot	Running	OK	Normal	False	True
atdisk	Atdisk	Not Available	Kernel Driver	False	False
Disabled	Stopped	OK	Ignore	False	False
ati2mtag	ati2mtag	c:\winnt\system32\drivers\ati2mtag.sys	Kernel	Running	OK
Driver	True	Manual	Running	OK	Ignore
True					
atmarpc	ATM ARP Client Protocol		Kernel Driver	False	False
c:\winnt\system32\drivers\atmarpc.sys			Kernel Driver	False	False
Manual	Stopped	OK	Normal	False	False
audstub	Audio Stub Driver	c:\winnt\system32\drivers\audstub.sys	Kernel Driver	Running	OK
Kernel Driver	True	Manual	Running	OK	Normal
False	True				
b57w2k	Broadcom NetXtreme Gigabit Ethernet		Kernel Driver	True	True
c:\winnt\system32\drivers\b57w2k.sys			Kernel Driver	True	True
Manual	Running	OK	Normal	False	True
beep	Beep	c:\winnt\system32\drivers\beep.sys	Kernel	Running	OK
Driver	True	System	Running	OK	Normal
True					
buslogic	BusLogic	Not Available	Kernel Driver	False	False
Disabled	Stopped	OK	Normal	False	False
cd20xrt	cd20xrt	Not Available	Kernel Driver	False	False
Disabled	Stopped	OK	Normal	False	False
cdaudio	Cdaudio	c:\winnt\system32\drivers\cdaudio.sys	Kernel	Running	OK
Driver	False	System	Stopped	OK	Ignore
False					

cdfs	Cdfs	c:\winnt\system32\drivers\cdfs.sys	File System	i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver								
Driver	True	Disabled	Running	OK	Normal	False		c:\winnt\system32\drivers\i8042prt.sys	Kernel Driver	True			
True								System	Running	OK	Normal	False	True
cdrom	CD-ROM Driver	c:\winnt\system32\drivers\cdrom.sys		ibmfte	IBM 10/100 Ethernet PCI Adapter NT Driver			c:\winnt\system32\drivers\ibmfnt5.sys	Kernel Driver	False			
Kernel Driver	True	System	Running	OK	Normal	False		Manual	Stopped	OK	Normal	False	False
False	True							ini910u	ini910u	Not Available	Kernel Driver	False	False
changer	Changer	Not Available	Kernel Driver	False	False	False		Disabled	Stopped	OK	Normal	False	False
System	Stopped	OK	Ignore	False	False	False		intelide	IntelIde	Not Available	Kernel Driver	False	False
cpqarray	Cpqarray	Not Available	Kernel Driver	False	False	False		Disabled	Stopped	OK	Normal	False	False
Disabled	Stopped	OK	Normal	False	False	False		ipfilterdriver	IP Traffic Filter Driver				
cpqarry2	cpqarry2	Not Available	Kernel Driver	False	False	False		c:\winnt\system32\drivers\ipfltdrv.sys	Kernel Driver	False			
Disabled	Stopped	OK	Normal	False	False	False		Manual	Stopped	OK	Normal	False	False
cpqcalm	cpqcalm	Not Available	Kernel Driver	False	False	False		ipinip	IP in IP Tunnel Driver:c:\winnt\system32\drivers\ipinip.sys				
Disabled	Stopped	OK	Normal	False	False	False		Kernel Driver	False	Manual	Stopped	OK	Normal
cpqfws2e	cpqfws2e	Not Available	Kernel Driver	False	False	False		ipnat	IP Network Address Translator	c:\winnt\system32\drivers\ipnat.sys			
Disabled	Stopped	OK	Normal	False	False	False		Kernel Driver	False	Manual	Stopped	OK	Normal
dac960nt	dac960nt	Not Available	Kernel Driver	False	False	False		False	False				
Disabled	Stopped	OK	Normal	False	False	False		ipsec	IPSEC driver	c:\winnt\system32\drivers\ipsec.sys			
deckzpsx	deckzpsx	Not Available	Kernel Driver	False	False	False		Kernel Driver	True	Manual	Running	OK	Normal
Disabled	Stopped	OK	Normal	False	False	False		False	True				
dfsdriver	DfsDriver	c:\winnt\system32\drivers\dfs.sys	File System Driver					ipsraidn	ipsraidn	Not Available	Kernel Driver	False	False
True	Boot	Running	OK	Normal	False	True		Disabled	Stopped	OK	Normal	False	False
disk	Disk Driver	c:\winnt\system32\drivers\disk.sys		irenum	IR Enumerator Service			c:\winnt\system32\drivers\irenum.sys	Kernel Driver	False	False	False	False
Kernel Driver	True	Boot	Running	OK	Normal	False		Manual	Stopped	OK	Normal	False	False
False	True							isapnp	PnP ISA/EISA Bus Driver				
diskperf	Diskperf	c:\winnt\system32\drivers\diskperf.sys	Kernel Driver					c:\winnt\system32\drivers\isapnp.sys	Kernel Driver	True			
Driver	True	Boot	Running	OK	Normal	False		Boot	Running	OK	Critical	False	True
True								kbdclass	Keyboard Class Driver				
dmboot	dmboot	c:\winnt\system32\drivers\dmboot.sys	Kernel Driver					c:\winnt\system32\drivers\kbdclass.sys	Kernel Driver	True			
Driver	False	Disabled	Stopped	OK	Normal	False		System	Running	OK	Normal	False	True
False								ksecdd	KSecDD	c:\winnt\system32\drivers\ksecdd.sys	Kernel Driver	False	True
dmio	Logical Disk Manager Driver		Kernel Driver	True				Driver	True	Boot	Running	OK	Normal
c:\winnt\system32\drivers\dmio.sys			Kernel Driver	True				True					
Boot	Running	OK	Normal	False	True	True		lbrtdc	lbrtdc	Not Available	Kernel Driver	False	False
dmload	dmload	c:\winnt\system32\drivers\dmload.sys	Kernel Driver					System	Stopped	OK	Ignore	False	False
Driver	True	Boot	Running	OK	Normal	False		lp6nds35	lp6nds35	Not Available	Kernel Driver	False	False
True								Disabled	Stopped	OK	Normal	False	False
e1000	Intel(R) PRO/1000 Network Connection Driver		Kernel Driver	True				mnmdd	mnmdd	c:\winnt\system32\drivers\mnmdd.sys	Kernel Driver	False	Ignore
c:\winnt\system32\drivers\e1000nt5.sys			Kernel Driver	True				Driver	True	System	Running	OK	Ignore
Manual	Running	OK	Normal	False	True	True		modem	Modem	c:\winnt\system32\drivers\modem.sys	Kernel Driver	False	Ignore
True	Disabled	Running	OK	Normal	False	True		Driver	False	Manual	Stopped	OK	Ignore
efs	EFS	c:\winnt\system32\drivers\efs.sys	File System Driver					False					
True	Disabled	Running	OK	Normal	False	True		mouclass	Mouse Class Driver	c:\winnt\system32\drivers\mouclass.sys	Kernel Driver	True	System
fastfat	Fastfat	c:\winnt\system32\drivers\fastfat.sys	File System Driver					False	True				
Driver	True	Disabled	Running	OK	Normal	False		mountmgr	MountMgr	c:\winnt\system32\drivers\mountmgr.sys	Kernel Driver	True	Boot
True								Driver	True	Boot	Running	OK	Normal
fd16_700	Fd16_700	Not Available	Kernel Driver	False	False	False		True					
Disabled	Stopped	OK	Normal	False	False	False		mrraid35x	mrraid35x	Not Available	Kernel Driver	False	False
fdc	Floppy Disk Controller Driver	c:\winnt\system32\drivers\fdc.sys						Disabled	Stopped	OK	Normal	False	False
Kernel Driver	True	Manual	Running	OK	Normal	False		Driver	True	System	Running	OK	Normal
False	True							True					
fips	Fips	c:\winnt\system32\drivers\fips.sys	Kernel Driver					msfs	Msfs	c:\winnt\system32\drivers\msfs.sys	File System Driver	True	System
Driver	True	Auto	Running	OK	Normal	False		Driver	True	System	Running	OK	Normal
True								True					
fireport	fireport	Not Available	Kernel Driver	False	False	False		mksrv	Microsoft Streaming Service Proxy				
Disabled	Stopped	OK	Normal	False	False	False		c:\winnt\system32\drivers\mksrv.sys	Kernel Driver	False			
flashpnt	flashpnt	Not Available	Kernel Driver	False	False	False		Manual	Stopped	OK	Normal	False	False
Disabled	Stopped	OK	Normal	False	False	False		msplock	Microsoft Streaming Clock Proxy				
flpydisk	Floppy Disk Driver	c:\winnt\system32\drivers\flpydisk.sys						c:\winnt\system32\drivers\msplock.sys	Kernel Driver	False			
Kernel Driver	True	Manual	Running	OK	Normal	False		Manual	Stopped	OK	Normal	False	False
False	True							True					
ftdisk	Volume Manager Driver		Kernel Driver	True									
c:\winnt\system32\drivers\ftdisk.sys			Kernel Driver	True									
Boot	Running	OK	Normal	False	True	True							
gpc	Generic Packet Classifier		Kernel Driver	True									
c:\winnt\system32\drivers\msgpc.sys			Kernel Driver	True									
Manual	Running	OK	Normal	False	True	True							

srv	Srv	c:\winnt\system32\drivers\srvc.sys	File System Driver				
True	Manual	Running	OK	Normal	False	True	
swenum	Software Bus Driver	c:\winnt\system32\drivers\swenum.sys					
Kernel Driver	True	Manual	Running	OK	Normal		
False	True						
symc810	symc810	Not Available		Kernel Driver	False		
Disabled	Stopped	OK	Normal	False	False		
symc8xx	symc8xx	Not Available		Kernel Driver	False		
Disabled	Stopped	OK	Normal	False	False		
sym_hi	sym_hi	Not Available		Kernel Driver	False		
Disabled	Stopped	OK	Normal	False	False		
tcpip	TCP/IP Protocol Driver	c:\winnt\system32\drivers\tcpip.sys					
Kernel Driver	True	System	Running	OK	Normal		
False	True						
tdasync	TDASYNC	c:\winnt\system32\drivers\tdasync.sys					
Kernel Driver	False	Manual	Stopped	OK	Ignore		
False	False						
tdipx	TDIPX	c:\winnt\system32\drivers\tdipx.sys					
Driver	False	Manual	Stopped	OK	Ignore	False	
False							
tdnetb	TDNETB	c:\winnt\system32\drivers\tdnetb.sys					
Driver	False	Manual	Stopped	OK	Ignore	False	
False							
tdpipe	TDPIPE	c:\winnt\system32\drivers\tdpipe.sys					
Driver	False	Manual	Stopped	OK	Ignore	False	
False							
tdspix	TDSPX	c:\winnt\system32\drivers\tdspix.sys					
Driver	False	Manual	Stopped	OK	Ignore	False	
False							
tdtcp	TDTCP	c:\winnt\system32\drivers\tdtcp.sys					
Driver	True	Manual	Running	OK	Ignore	False	
True							
termdd	Terminal Device Driver	c:\winnt\system32\drivers\termdd.sys					
Auto	Running	OK	Normal	False	True		
tga	tga	Not Available		Kernel Driver	False		
System	Stopped	OK	Ignore	False	False		
udfs	Udfs	c:\winnt\system32\drivers\udfs.sys					
Driver	False	Disabled	Stopped	OK	Normal	False	
False							
uhcd	Microsoft USB Universal Host Controller Driver	c:\winnt\system32\drivers\uhcd.sys					
Manual	Running	OK	Normal	False	True		
ultra66	ultra66	Not Available		Kernel Driver	False		
Disabled	Stopped	OK	Normal	False	False		
update	Microcode Update Driver	c:\winnt\system32\drivers\update.sys					
Manual	Running	OK	Normal	False	True		
usbehci	Microsoft USB 2.0 Enhanced Host Controller Miniport Driver	c:\winnt\system32\drivers\usbehci.sys					
Manual	Running	OK	Normal	False	True		
usbhub	Microsoft USB Standard Hub Driver	c:\winnt\system32\drivers\usbhub.sys					
Manual	Running	OK	Normal	False	True		
usbhub20	USB 2.0 Root Hub Support	c:\winnt\system32\drivers\usbhub20.sys					
Manual	Running	OK	Normal	False	True		
vgasave	VgaSave	c:\winnt\system32\drivers\vga.sys					
Driver	True	System	Running	OK	Ignore	False	
True							
wanarp	Remote Access IP ARP Driver	c:\winnt\system32\drivers\wanarp.sys					
Manual	Running	OK	Normal	False	True		
wdica	WDICA	Not Available		Kernel Driver	False		
Manual	Stopped	OK	Ignore	False	False		
nal	Nal Service	??\c:\winnt\system32\drivers\iqvw32.sys					
Kernel Driver	True	Disabled	Running	OK	Normal		
False	True						

[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_cu.jar;C:\SQLLIB\bin;C:\SQLLIB\java\common.jar	<SYSTEM>
ComSpec	%SystemRoot%\system32\cmd.exe	<SYSTEM>
DB2INSTANCE	DB2	<SYSTEM>
DB2TEMPDIR	C:\SQLLIB\	<SYSTEM>
INCLUDE	C:\SQLLIB\INCLUDE;C:\SQLLIB\LIB;C:\Program Files\Microsoft Visual Studio\VC98\Include	<SYSTEM>
LIB	C:\SQLLIB\LIB	<SYSTEM>
NUMBER_OF_PROCESSORS	4	<SYSTEM>
OS	Windows_NT	<SYSTEM>
Os2LibPath	%SystemRoot%\system32\os2\dll;Path	<SYSTEM>
Path	C:\Perl\bin;%SystemRoot%\system32;%SystemRoot%\System32\Wbem;C:\SQLLIB\BIN;C:\SQLLIB\FUNCTION;c:\tools;c:\tools\util;C:\Program Files\Intel\DMIX;c:\Program Files\Microsoft Visual Studio\VC98\bin;c:\Program Files\Microsoft Visual Studio\Common\MSDev98\Bin	<SYSTEM>
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH	<SYSTEM>
PROCESSOR_ARCHITECTURE	x86	<SYSTEM>
PROCESSOR_IDENTIFIER	x86 Family 15 Model 4 Stepping 3, GenuineIntel	<SYSTEM>
PROCESSOR_LEVEL	15	<SYSTEM>
PROCESSOR_REVISION	0403	<SYSTEM>
TEMP	%SystemRoot%\TEMP	<SYSTEM>
TMP	%SystemRoot%\TEMP	<SYSTEM>
windir	%SystemRoot%	<SYSTEM>
TEMP	%USERPROFILE%\Local Settings\Temp	VCLIENT150\TPCC
TMP	%USERPROFILE%\Local Settings\Temp	VCLIENT150\TPCC
include	C:\Program Files\Microsoft Visual Studio\VC98\atl\include;C:\Program Files\Microsoft Visual Studio\VC98\mf\include;C:\Program Files\Microsoft Visual Studio\VC98\include	VCLIENT150\Administrator
lib	C:\Program Files\Microsoft Visual Studio\VC98\mf\lib;C:\Program Files\Microsoft Visual Studio\VC98\lib	VCLIENT150\Administrator
MSDevDir	C:\Program Files\Microsoft Visual Studio\Common\MSDev98	VCLIENT150\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	VCLIENT150\Administrator
TEMP	%USERPROFILE%\Local Settings\Temp	VCLIENT150\Administrator
TMP	%USERPROFILE%\Local Settings\Temp	VCLIENT150\Administrator

[Jobs]

[Following are sub-categories of this main category]

[Print]

Document Size	Owner	Notify	Status	Time Submitted
Start Time	Until Time	Elapsed Time	Pages Printed	Job ID
Priority	Parameters	Driver Name	Print Processor	Host Print
Queue	Data Type	Name		
Unknown	Unknown	Unknown	Unknown	Unknown
Unknown	Unknown	Unknown	Unknown	Unknown
Unknown	Unknown	Unknown	Unknown	Unknown

[Network Connections]

Local Name	Remote Name	Type	Status	User Name	
No network connections information					
[Running Tasks]					
Name	Path	Process ID	Priority	Min Working Set	Max
Working Set	Start Time	Version	Size	File Date	
system idle process	Not Available		0	0	Not Available
Available	Not Available		Not Available	Unknown	Unknown
Unknown					
system	Not Available	8	8	0	1413120
Not Available	Unknown	Unknown	Unknown	Unknown	Unknown
smss.exe	c:\winnt\system32\smss.exe		192	11	204800
1413120	10/21/2005 10:46:37 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	10/21/2005 10:46:40 AM		Unknown	Unknown	Unknown
Unknown					
winlogon.exe	c:\winnt\system32\winlogon.exe		212	13	
204800	1413120 10/21/2005 10:46:42 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 10/21/2005 10:46:43 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	10/21/2005 10:46:43 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 10/21/2005 10:46:44 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		500	8	
204800	1413120 10/21/2005 10:46:47 AM		5.00.2134.1		
7.77 KB (7,952 bytes)				12/7/1999 7:00:00 AM	
msdtc.exe	c:\winnt\system32\msdtc.exe		532	8	204800
1413120	10/21/2005 10:46:47 AM		1999.9.3421.3	6.77 KB	
(6,928 bytes)				11/3/2004 8:55:08 AM	
db2jds.exe	c:\sqlib\bin\db2jds.exe		704	8	204800
1413120	10/21/2005 10:46:49 AM		8.1.6.574	193.12 KB (197,752 bytes)	
(197,752 bytes)				6/17/2004 11:30:56 PM	
db2sec.exe	c:\sqlib\bin\db2sec.exe		720	8	204800
1413120	10/21/2005 10:46:51 AM		8.1.6.574	29.11 KB (29,808 bytes)	
(29,808 bytes)				6/17/2004 11:32:48 PM	
svchost.exe	c:\winnt\system32\svchost.exe		736	8	
204800	1413120 10/21/2005 10:46:51 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		768	9	204800
1413120	10/21/2005 10:46:52 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		852	8	204800
1413120	10/21/2005 10:46:53 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		980	8	204800
1413120	10/21/2005 10:47:08 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		1036	8	
204800	1413120 10/21/2005 10:47:09 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		1076	8	
204800	1413120 10/21/2005 10:47:09 AM		1.50.1085.0100		
192.10 KB (196,706 bytes)				11/3/2004 3:03:07 PM	
inetinfo.exe	c:\winnt\system32\inetinfo.exe		1092	8	
204800	1413120 10/21/2005 10:47:09 AM		5.00.0984	14.27 KB	
(14,608 bytes)				11/3/2004 3:03:22 PM	
dfssvc.exe	c:\winnt\system32\dfssvc.exe		1064	8	204800
1413120	10/21/2005 10:47:13 AM		5.00.2195.6664	88.77 KB	
(90,896 bytes)				11/3/2004 3:02:32 PM	

explorer.exe	c:\winnt\explorer.exe		1348	8	204800
1413120	10/21/2005 10:47:48 AM		5.00.3700.6690	237.77 KB	
(243,472 bytes)				11/3/2004 3:03:02 PM	
svchost.exe	c:\winnt\system32\svchost.exe		1468	8	
204800	1413120 10/21/2005 10:47:51 AM		5.00.2134.1		
7.77 KB (7,952 bytes)				12/7/1999 7:00:00 AM	
csrss.exe	Not Available		332	13	Not Available
Not Available	10/21/2005 11:12:21 AM		Unknown	Unknown	Unknown
Unknown					
winlogon.exe	c:\winnt\system32\winlogon.exe		760	13	
204800	1413120 10/21/2005 11:12:21 AM		5.00.2195.6714		
176.77 KB (181,008 bytes)				11/3/2004 3:03:01 PM	
rdpclip.exe	c:\winnt\system32\rdpclip.exe		1452	8	204800
1413120	10/21/2005 11:12:32 AM		5.00.2174.1	39.77 KB	
(40,720 bytes)				11/3/2004 8:55:10 AM	
explorer.exe	c:\winnt\explorer.exe		1332	8	204800
1413120	10/21/2005 11:12:36 AM		5.00.3700.6690	237.77 KB	
(243,472 bytes)				11/3/2004 3:03:02 PM	
cmd.exe	c:\winnt\system32\cmd.exe		1252	8	204800
1413120	10/21/2005 11:12:41 AM		5.00.2195.6656	230.77 KB	
(236,304 bytes)				11/3/2004 3:02:30 PM	
db2bp.exe	c:\sqlib\bin\db2bp.exe		1524	8	204800
1413120	10/21/2005 11:16:22 AM		8.1.6.574	821.11 KB (840,816 bytes)	
(840,816 bytes)				6/17/2004 11:28:24 PM	
cmd.exe	c:\winnt\system32\cmd.exe		1560	8	204800
1413120	10/21/2005 1:49:53 PM		5.00.2195.6656	230.77 KB	
(236,304 bytes)				11/3/2004 3:02:30 PM	
mmc.exe	c:\winnt\system32\mmc.exe		1536	8	204800
1413120	10/21/2005 3:57:24 PM		5.00.2195.6601	589.27 KB	
(603,408 bytes)				11/3/2004 3:02:41 PM	
mmc.exe	c:\winnt\system32\mmc.exe		988	8	204800
1413120	10/21/2005 4:00:51 PM		5.00.2195.6601	589.27 KB	
(603,408 bytes)				11/3/2004 3:02:41 PM	
mdm.exe	c:\winnt\system32\mdm.exe		1676	8	204800
1413120	10/21/2005 4:01:37 PM		6.00.8424	121.29 KB (124,200 bytes)	
(124,200 bytes)				11/3/2004 8:56:48 AM	
mmc.exe	c:\winnt\system32\mmc.exe		1232	8	204800
1413120	10/21/2005 4:03:05 PM		5.00.2195.6601	589.27 KB	
(603,408 bytes)				11/3/2004 3:02:41 PM	
rsvp.exe	c:\winnt\system32\rsvp.exe		1752	8	204800
1413120	10/21/2005 4:03:52 PM		5.00.2195.6663	172.77 KB	
(176,912 bytes)				11/3/2004 3:02:54 PM	

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
traffic.dll	5.00.2195.6613	30.77 KB (31,504 bytes)		Microsoft Corporation	11/3/2004
3:02:58 PM					
c:\winnt\system32\traffic.dll					
rsvp.exe	5.00.2195.6663	172.77 KB (176,912 bytes)		Microsoft Corporation	11/3/2004
3:02:54 PM					
c:\winnt\system32\rsvp.exe					
mdm.exe	6.00.8424	121.29 KB (124,200 bytes)		Microsoft Corporation	11/3/2004 8:56:48 AM
Microsoft Corporation					
btagtresenu.dll	9.2.4.5	20.00 KB (20,480 bytes)		Intel(R) Corporation	3/14/2005
2:20:07 PM					
files\intel\dmix\resource\btagtresenu.dll					
btagtsrv.dll	9.2.4.5	96.00 KB (98,304 bytes)		Intel(R) Corporation	3/14/2005
2:20:07 PM					
files\intel\dmix\btagtsrv.dll					
teamresenu.dll	9.2.4.4	172.00 KB (176,128 bytes)		Intel(R) Corporation	3/14/2005
2:20:07 PM					
files\intel\dmix\resource\teamresenu.dll					
teamsrv.dll	9.2.4.4	256.00 KB (262,144 bytes)		Intel(R) Corporation	3/14/2005 2:20:07 PM
Intel(R) Corporation					
c:\program files\intel\dmix\teamsrv.dll					
brandres.dll	9.2.4.7	12.00 KB (12,288 bytes)		Intel(R) Corporation	3/14/2005
2:20:07 PM					
files\intel\dmix\resource\brandres.dll					

etcoinst.dll 2.1.3.0 built by: WinDDK 54.50 KB (55,808 bytes) 11/16/2004 4:35:46 PM Intel Corporation c:\winnt\system32\etcoinst.dll	riched32.dll 5.00.2134.1 3.77 KB (3,856 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\riched32.dll
intelnic.dll 8.2.3.0 built by: WinDDK 19.00 KB (19,456 bytes) 10/29/2004 5:01:48 PM Intel(R) Corporation c:\winnt\system32\intelnic.dll	els.dll 5.00.2195.6610 154.27 KB (157,968 bytes) 11/3/2004 3:02:34 PM Microsoft Corporation c:\winnt\system32\els.dll
vlanresenu.dll 9.2.4.4 20.00 KB (20,480 bytes) 3/14/2005 2:20:07 PM Intel(R) Corporation c:\program files\intel\dmix\resource\vlanresenu.dll	ntmsmgr.dll 1.0,0,1 427.77 KB (438,032 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation and HighGround Systems, Inc. c:\winnt\system32\ntmsmgr.dll
vlanrv.dll 9.2.4.4 156.00 KB (159,744 bytes) 3/14/2005 2:20:07 PM Intel(R) Corporation c:\program files\intel\dmix\vlanrv.dll	mmfutil.dll 1.50.1085.0000 32.06 KB (32,829 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\mmfutil.dll
dmixresenu.dll 9.2.4.7 40.00 KB (40,960 bytes) 3/14/2005 2:20:07 PM Intel(R) Corporation c:\program files\intel\dmix\resource\dmixresenu.dll	logdrive.dll 1.50.1085.0000 200.06 KB (204,863 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\logdrive.dll
msvcpx60.dll 6.00.8972.0 392.05 KB (401,462 bytes) 6/17/1998 1:00:00 AM Microsoft Corporation c:\winnt\system32\msvcpx60.dll	dfrgres.dll 5.00.2150.1 27.50 KB (28,160 bytes) 12/7/1999 7:00:00 AM Executive Software International, Inc. c:\winnt\system32\dfrgres.dll
ncs2instutility.dll 9.2.4.0 124.00 KB (126,976 bytes) 3/14/2005 2:20:07 PM Intel(R) Corporation c:\winnt\system32\ncs2instutility.dll	dfrgsnap.dll 5.00.2195.6605 41.77 KB (42,768 bytes) 11/3/2004 3:02:32 PM Executive Software International, Inc. c:\winnt\system32\dfrgsnap.dll
accessor.dll 9.2.4.4 284.00 KB (290,816 bytes) 3/14/2005 2:20:07 PM Intel(R) Corporation c:\winnt\system32\accessor.dll	dmdskres.dll 2195.6605.297.3 119.50 KB (122,368 bytes) 11/3/2004 3:02:33 PM Microsoft Corp., VERITAS Software c:\winnt\system32\dmdskres.dll
ncs2dmix.dll 9.2.4.7 376.00 KB (385,024 bytes) 3/14/2005 2:20:07 PM Intel(R) Corporation c:\winnt\system32\ncs2dmix.dll	dmutil.dll 2195.6605.297.3 42.27 KB (43,280 bytes) 11/3/2004 3:02:33 PM VERITAS Software Corp. c:\winnt\system32\dmutil.dll
olepro32.dll 5.0.4522 160.27 KB (164,112 bytes) 11/3/2004 3:02:51 PM Microsoft Corporation c:\winnt\system32\olepro32.dll	ntmsapi.dll 5.00.1948.1 52.27 KB (53,520 bytes) 11/3/2004 3:02:49 PM Microsoft Corporation c:\winnt\system32\ntmsapi.dll
dmocx.dll 5.00.2134.1 23.27 KB (23,824 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\dmocx.dll	dmdskmgr.dll 2195.6605.297.3 159.77 KB (163,600 bytes) 11/3/2004 3:02:33 PM Microsoft Corp., VERITAS Software c:\winnt\system32\dmdskmgr.dll
wbemprox.dll 1.50.1085.0100 40.10 KB (41,061 bytes) 11/3/2004 3:03:07 PM Microsoft Corporation c:\winnt\system32\wbem\wbemprox.dll	mycomput.dll 5.00.2195.6601 107.77 KB (110,352 bytes) 11/3/2004 3:02:47 PM Microsoft Corporation c:\winnt\system32\mycomput.dll
rassapi.dll 5.00.2195.6604 14.27 KB (14,608 bytes) 11/3/2004 3:02:53 PM Microsoft Corporation c:\winnt\system32\rassapi.dll	mmcndmgr.dll 5.00.2195.6601 816.27 KB (835,856 bytes) 11/3/2004 3:02:41 PM Microsoft Corporation c:\winnt\system32\mmcndmgr.dll
adsnt.dll 5.00.2195.6658 196.77 KB (201,488 bytes) 11/3/2004 3:02:26 PM Microsoft Corporation c:\winnt\system32\adsnt.dll	mmc.exe 5.00.2195.6601 589.27 KB (603,408 bytes) 11/3/2004 3:02:41 PM Microsoft Corporation c:\winnt\system32\mmc.exe
dbghelp.dll 5.00.2195.6613 159.27 KB (163,088 bytes) 6/19/2003 1:05:04 PM Microsoft Corporation c:\winnt\system32\dbghelp.dll	db2tcp.dll 8.1.6.574 64.06 KB (65,598 bytes) 6/17/2004 8:31:06 PM International Business Machines Corporation c:\sql\lib\bin\db2tcp.dll
localsec.dll 5.00.2195.6623 240.27 KB (246,032 bytes) 11/3/2004 3:02:40 PM Microsoft Corporation c:\winnt\system32\localsec.dll	db2daskrb.dll 8.1.6.574 36.07 KB (36,935 bytes) 6/17/2004 8:30:10 PM International Business Machines Corporation c:\sql\lib\bin\db2daskrb.dll
devmgr.dll 5.00.2195.6619 216.77 KB (221,968 bytes) 11/3/2004 3:02:32 PM Microsoft Corporation c:\winnt\system32\devmgr.dll	db2osse_db2.dll 8.1.6.574 60.08 KB (61,517 bytes) 6/17/2004 8:31:10 PM International Business Machines Corporation c:\sql\lib\bin\db2osse_db2.dll
filemgmt.dll 5.00.2195.6601 287.77 KB (294,672 bytes) 11/3/2004 3:02:35 PM Microsoft Corporation c:\winnt\system32\filemgmt.dll	db2genreg.dll 8.1.6.574 152.07 KB (155,722 bytes) 6/17/2004 8:28:20 PM International Business Machines Corporation c:\sql\lib\bin\db2genreg.dll
pdh.dll 5.00.2195.6660 148.27 KB (151,824 bytes) 11/3/2004 3:02:51 PM Microsoft Corporation c:\winnt\system32\pdh.dll	db2dascmn.dll 8.1.6.574 80.07 KB (81,991 bytes) 6/17/2004 8:30:02 PM International Business Machines Corporation c:\sql\lib\bin\db2dascmn.dll
smlogcfg.dll 5.00.2195.6612 278.77 KB (285,456 bytes) 11/3/2004 3:02:56 PM Microsoft Corporation c:\winnt\system32\smlogcfg.dll	db2dasapi.dll 8.1.6.574 312.07 KB (319,559 bytes) 6/17/2004 8:30:02 PM International Business Machines Corporation c:\sql\lib\bin\db2dasapi.dll
cabinet.dll 5.00.2147.1 54.77 KB (56,080 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\cabinet.dll	db2bp.exe 8.1.6.574 821.11 KB (840,816 bytes) 6/17/2004 11:28:24 PM International Business Machines Corporation c:\sql\lib\bin\db2bp.exe
msinfo32.dll 5.00.2195.6601 312.27 KB (319,760 bytes) 11/3/2004 3:03:09 PM Microsoft Corporation c:\program files\common files\microsoft shared\msinfo\msinfo32.dll	cmd.exe 5.00.2195.6656 230.77 KB (236,304 bytes) 11/3/2004 3:02:30 PM Microsoft Corporation c:\winnt\system32\cmd.exe
riched20.dll 5.30.23.1215 421.77 KB (431,888 bytes) 11/3/2004 3:02:53 PM Microsoft Corporation c:\winnt\system32\riched20.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

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
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
usp10.dll 1.0325.2195.6692	308.27 KB (315,664 bytes)	11/3/2004	3:02:59 PM	Microsoft Corporation	c:\winnt\system32\usp10.dll
thumbvw.dll 5.00.3502.6601	183.27 KB (187,664 bytes)	11/3/2004	3:02:58 PM	Microsoft Corporation	c:\winnt\system32\thumbvw.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
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
mshtml.dll 5.00.3700.6699	229.77 KB (235,280 bytes)	11/3/2004	3:02:44 PM	Microsoft Corporation	c:\winnt\system32\mshtml.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
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
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
jscrip.dll 5.1.0.8513	476.06 KB (487,481 bytes)	11/3/2004	3:02:40 PM	Microsoft Corporation	c:\winnt\system32\jscrip.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
wininet.dll 5.00.3700.6713	455.77 KB (466,704 bytes)	11/3/2004	3:03:01 PM	Microsoft Corporation	c:\winnt\system32\wininet.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
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
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
stobject.dll 5.00.2195.6601	79.27 KB (81,168 bytes)	11/3/2004	3:02:57 PM	Microsoft Corporation	c:\winnt\system32\stobject.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
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
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
ilsdbx.dll 5.00.0984	56.27 KB (57,616 bytes)	3/15/2005	5:39:16 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	c:\winnt\system32\msrd3x40.dll

odbc32.dll	3.520.7713.0	92.00 KB (94,208 bytes)	11/3/2004 5:27:34 PM	Microsoft Corporation	c:\winnt\system32\odbc32.dll
comsvcs.dll	2000.2.3504.0	1.38 MB (1,448,208 bytes)	11/3/2004 3:02:30 PM	Microsoft Corporation	c:\winnt\system32\comsvcs.dll
mtxdm.dll	2000.2.3504.0	22.77 KB (23,312 bytes)	11/3/2004 3:02:47 PM	Microsoft Corporation	c:\winnt\system32\mtxdm.dll
odbcji32.dll	4.0.6200.0	52.27 KB (53,520 bytes)	11/3/2004 3:02:50 PM	Microsoft Corporation	c:\winnt\system32\odbcji32.dll
odbcjt32.dll	4.0.6200.0	264.27 KB (270,608 bytes)	11/3/2004 3:02:50 PM	Microsoft Corporation	c:\winnt\system32\odbcjt32.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	c:\program files\common files\system\ole db\msdasqlr.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	c:\program files\common files\system\ole db\msdatl3.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	c:\program files\common files\system\ole db\msdasql.dll
ldapdbx.dll	5.00.0984	73.27 KB (75,024 bytes)	5:39:16 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\ldapdbx.dll
dscomobx.dll	5.00.0984	188.77 KB (193,296 bytes)	5:39:16 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\dscomobx.dll
httpext.dll	5.00.0984	240.77 KB (246,544 bytes)	11/3/2004 3:03:21 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\httpext.dll
rpcproxy.dll	5.00.2195.6701	16.27 KB (16,656 bytes)	11/3/2004 3:04:06 PM	Microsoft Corporation	c:\winnt\system32\rpcproxy.dll
fpexedll.dll	4.0.2.7523	20.06 KB (20,541 bytes)	3:03:13 PM	Microsoft Corporation	c:\program files\common files\microsoft shared\web server extensions\40\bin\fpexedll.dll
md5filt.dll	5.00.0984	32.77 KB (33,552 bytes)	11/3/2004 3:03:23 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\md5filt.dll
gzip.dll	5.00.0984	30.27 KB (30,992 bytes)	11/3/2004 3:03:21 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\gzip.dll
compfilt.dll	5.00.0984	22.77 KB (23,312 bytes)	3:03:21 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\compfilt.dll
odbcint.dll	3.520.7713.0	88.00 KB (90,112 bytes)	5:27:33 PM	Microsoft Corporation	c:\winnt\system32\odbcint.dll
odbc32.dll	3.520.7713.0	196.00 KB (200,704 bytes)	5:27:34 PM	Microsoft Corporation	c:\winnt\system32\odbc32.dll
ldapacx.dll	5.00.0984	8.27 KB (8,464 bytes)	3:15:2005 5:39:16 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\ldapacx.dll
storedbx.dll	5.00.0984	251.27 KB (257,296 bytes)	3:03:26 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\storedbx.dll
ladminx.dll	5.00.0984	61.27 KB (62,736 bytes)	3:04:05 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\ladminx.dll
sspifilt.dll	5.00.0984	42.77 KB (43,792 bytes)	11/3/2004 3:03:23 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\sspifilt.dll
iscomlog.dll	5.00.0984	24.27 KB (24,848 bytes)	3:03:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\iscomlog.dll
lonsint.dll	5.00.0984	11.77 KB (12,048 bytes)	11/3/2004 3:03:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\lonsint.dll
inetsloc.dll	5.00.0984	20.27 KB (20,752 bytes)	11/3/2004 3:02:38 PM	Microsoft Corporation	c:\winnt\system32\inetsloc.dll
w3svc.dll	5.00.0984	338.27 KB (346,384 bytes)	11/3/2004 3:03:23 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\w3svc.dll
staxmem.dll	5.00.0984	8.27 KB (8,464 bytes)	11/3/2004 3:02:57 PM	Microsoft Corporation	c:\winnt\system32\staxmem.dll
extrace.dll	5.00.0984	13.77 KB (14,096 bytes)	8:55:28 AM	Microsoft Corporation	c:\winnt\system32\extrace.dll
iisfecnv.dll	5.00.0984	7.27 KB (7,440 bytes)	11/3/2004 8:55:27 AM	Microsoft Corporation	c:\winnt\system32\inetsrv\iisfecnv.dll
isatq.dll	5.00.0984	61.27 KB (62,736 bytes)	11/3/2004 3:03:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\isatq.dll
infocomm.dll	5.00.0984	242.27 KB (248,080 bytes)	3:03:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\infocomm.dll
ldapsvcx.dll	5.00.0984	126.77 KB (129,808 bytes)	3:03:33 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\ldapsvcx.dll
security.dll	5.00.2154.1	5.77 KB (5,904 bytes)	12/7/1999 7:00:00 AM	Microsoft Corporation	c:\winnt\system32\security.dll
svcxext.dll	5.00.0984	39.77 KB (40,720 bytes)	11/3/2004 3:03:23 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\svcxext.dll
admexs.dll	5.00.0984	27.77 KB (28,432 bytes)	11/3/2004 3:03:20 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\admexs.dll
wamreg.dll	5.00.0984	45.77 KB (46,864 bytes)	11/3/2004 3:03:24 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\wamreg.dll
metadata.dll	5.00.0984	68.77 KB (70,416 bytes)	3:03:23 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\metadata.dll
iismap.dll	5.00.0984	56.27 KB (57,616 bytes)	11/3/2004 3:02:37 PM	Microsoft Corporation	c:\winnt\system32\iismap.dll
nsepm.dll	5.00.0984	43.27 KB (44,304 bytes)	11/3/2004 3:03:23 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\nsepm.dll
admwprox.dll	5.00.0984	31.77 KB (32,528 bytes)	8:55:27 AM	Microsoft Corporation	c:\winnt\system32\admwprox.dll
coadmin.dll	5.00.0984	39.77 KB (40,720 bytes)	3:03:21 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\coadmin.dll
iisadmin.dll	5.00.0984	15.77 KB (16,144 bytes)	3:03:21 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\iisadmin.dll
rpref.dll	5.00.0984	4.27 KB (4,368 bytes)	11/3/2004 3:03:23 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\rpref.dll
iisrtl.dll	5.00.0984	121.27 KB (124,176 bytes)	11/3/2004 3:02:37 PM	Microsoft Corporation	c:\winnt\system32\iisrtl.dll
inetinfo.exe	5.00.0984	14.27 KB (14,608 bytes)	3:03:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\inetinfo.exe
netui1.dll	5.00.2134.1	210.27 KB (215,312 bytes)	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\netui1.dll
netui0.dll	5.00.2195.6601	70.27 KB (71,952 bytes)	3:02:48 PM	Microsoft Corporation	c:\winnt\system32\netui0.dll
ntlanman.dll	5.00.2195.6601	35.27 KB (36,112 bytes)	12/7/1999 7:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntlanman.dll
wshnetbs.dll	5.00.2134.1	7.77 KB (7,952 bytes)	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\wshnetbs.dll
ntmarta.dll	5.00.2195.6666	100.27 KB (102,672 bytes)	3:02:49 PM	Microsoft Corporation	c:\winnt\system32\ntmarta.dll
perfos.dll	5.00.2155.1	21.27 KB (21,776 bytes)	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\perfos.dll

psapi.dll	5.00.2134.1	28.27 KB (28,944 bytes)	12/7/1999	ntmsdba.dll	5.00.2195.6655	169.27 KB (173,328 bytes)	
7:00:00 AM	Microsoft Corporation			11/3/2004 3:02:49 PM	Microsoft Corporation		
c:\winnt\system32\psapi.dll				c:\winnt\system32\ntmsdba.dll			
provthrd.dll	1.50.1085.0000	68.07 KB (69,708 bytes)		rastapi.dll	5.00.2195.6604	52.77 KB (54,032 bytes)	12/7/1999
11/3/2004 1:58:37 PM	Microsoft Corporation			7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\wbem\provthrd.dll				c:\winnt\system32\rastapi.dll			
ntevt.dll	1.50.1085.0072	192.06 KB (196,671 bytes)	11/3/2004	rasdlg.dll	5.00.2195.6625	516.77 KB (529,168 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\rasdlg.dll			
framedyn.dll	1.50.1085.0076	164.07 KB (168,009 bytes)		netcfgx.dll	5.00.2195.6604	534.77 KB (547,600 bytes)	11/3/2004
11/3/2004 3:03:06 PM	Microsoft Corporation			3:02:47 PM	Microsoft Corporation		
c:\winnt\system32\wbem\framedyn.dll				c:\winnt\system32\netcfgx.dll			
cimwin32.dll	1.50.1085.0103	1.04 MB (1,089,637 bytes)		rasmans.dll	5.00.2195.6696	149.77 KB (153,360 bytes)	
11/3/2004 3:03:05 PM	Microsoft Corporation			11/3/2004 3:02:52 PM	Microsoft Corporation		
c:\winnt\system32\wbem\cimwin32.dll				c:\winnt\system32\rasmans.dll			
wbemsvc.dll	1.50.1085.0007	40.07 KB (41,036 bytes)		wmi.dll	5.00.2191.1	6.27 KB (6,416 bytes)	12/7/1999 7:00:00 AM
11/3/2004 3:03:07 PM	Microsoft Corporation			Microsoft Corporation	c:\winnt\system32\wmi.dll		
c:\winnt\system32\wbem\wbemsvc.dll				netshell.dll	5.00.2195.6604	466.27 KB (477,456 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\netshell.dll			
c:\winnt\system32\wbem\wbemess.dll				netman.dll	5.00.2195.6660	93.27 KB (95,504 bytes)	11/3/2004
fastprox.dll	1.50.1085.0100	152.10 KB (155,749 bytes)		3:02:48 PM	Microsoft Corporation		
11/3/2004 3:03:06 PM	Microsoft Corporation			c:\winnt\system32\netman.dll			
c:\winnt\system32\wbem\fastprox.dll				sens.dll	5.00.2195.6627	37.27 KB (38,160 bytes)	11/3/2004
wbemcore.dll	1.50.1085.0100	632.09 KB (647,257 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\wbemcore.dll				iashlpr.dll	5.00.2184.1	33.27 KB (34,064 bytes)	12/7/1999
wbemcomm.dll	1.50.1085.0100	692.09 KB (708,696 bytes)		7:00:00 AM	Microsoft Corporation		
11/3/2004 3:03:06 PM	Microsoft Corporation			c:\winnt\system32\iashlpr.dll			
c:\winnt\system32\wbem\wbemcomm.dll				iasacct.dll	5.00.2195.6603	28.27 KB (28,944 bytes)	11/3/2004
winmgmt.exe	1.50.1085.0100	192.10 KB (196,706 bytes)		3:02:37 PM	Microsoft Corporation		
11/3/2004 3:03:07 PM	Microsoft Corporation			c:\winnt\system32\iasacct.dll			
c:\winnt\system32\wbem\winmgmt.exe				iasuser.dll	5.00.2195.6622	19.77 KB (20,240 bytes)	12/7/1999
simptcp.dll	5.00.2134.1	19.27 KB (19,728 bytes)	3/15/2005	7:00:00 AM	Microsoft Corporation		
5:39:17 PM	Microsoft Corporation			c:\winnt\system32\iasuser.dll			
c:\winnt\system32\simptcp.dll				iasnap.dll	5.00.2195.6601	58.77 KB (60,176 bytes)	11/3/2004
tcpvcs.exe	5.00.2134.1	24.77 KB (25,360 bytes)		3:02:37 PM	Microsoft Corporation		
12/7/1999 7:00:00 AM	Microsoft Corporation			c:\winnt\system32\iasnap.dll			
c:\winnt\system32\tcpvcs.exe				iaspipe.dll	5.00.2134.1	41.77 KB (42,768 bytes)	12/7/1999
msidle.dll	5.00.2920.0000	6.27 KB (6,416 bytes)	12/7/1999 7:00:00 AM	7:00:00 AM	Microsoft Corporation		
Microsoft Corporation	c:\winnt\system32\msidle.dll			c:\winnt\system32\iaspipe.dll			
mstask.exe	4.71.2195.6704	116.77 KB (119,568 bytes)	11/3/2004	expshr.dll	6.0.9589	372.03 KB (380,957 bytes)	11/3/2004 3:02:35 PM
3:02:46 PM	Microsoft Corporation			Microsoft Corporation	c:\winnt\system32\expshr.dll		
c:\winnt\system32\mstask.exe				vbajet32.dll	6.1.9431	30.03 KB (30,749 bytes)	11/3/2004
regsvc.exe	5.00.2195.6701	66.77 KB (68,368 bytes)	11/3/2004	3:02:59 PM	Microsoft Corporation		
3:02:53 PM	Microsoft Corporation			c:\winnt\system32\vbajet32.dll			
c:\winnt\system32\regsvc.exe				msjtes40.dll	4.00.7328.0	236.27 KB (241,936 bytes)	
llsrpc.dll	5.00.2195.6601	47.77 KB (48,912 bytes)	12/7/1999	11/3/2004 3:02:45 PM	Microsoft Corporation		
7:00:00 AM	Microsoft Corporation			c:\winnt\system32\msjtes40.dll			
c:\winnt\system32\llsrpc.dll				oledb32r.dll	2.70.7713.0	built by: Lab06_N(dagbuild)	64.00 KB (65,536 bytes)
llssrv.exe	5.00.2195.6697	81.77 KB (83,728 bytes)	6/19/2003	11/3/2004 5:27:34 PM	Microsoft Corporation		
1:05:04 PM	Microsoft Corporation			c:\program files\common files\system\ole db\oledb32r.dll			
c:\winnt\system32\llssrv.exe				comdlg32.dll	5.00.3700.6693	235.77 KB (241,424 bytes)	
ipbootp.dll	5.00.2168.1	33.77 KB (34,576 bytes)	12/7/1999	12/7/1999 7:00:00 AM	Microsoft Corporation		
7:00:00 AM	Microsoft Corporation			c:\winnt\system32\comdlg32.dll			
c:\winnt\system32\ipbootp.dll				msdart.dll	2.70.7713.0	built by: Lab06_N(dagbuild)	124.00 KB (126,976 bytes)
cryptui.dll	5.131.2195.6628	433.27 KB (443,664 bytes)	11/3/2004	11/3/2004 5:27:34 PM	Microsoft Corporation		
3:02:31 PM	Microsoft Corporation			c:\winnt\system32\msdart.dll			
c:\winnt\system32\cryptui.dll				oledb32.dll	2.70.7713.0	built by: Lab06_N(dagbuild)	404.00 KB (413,696 bytes)
rastls.dll	5.00.2195.6680	98.27 KB (100,624 bytes)	11/3/2004	11/3/2004 5:27:34 PM	Microsoft Corporation		
3:02:53 PM	Microsoft Corporation			c:\program files\common files\system\ole db\oledb32.dll			
c:\winnt\system32\rastls.dll				msjint40.dll	4.00.6508.0	148.27 KB (151,824 bytes)	
raschap.dll	5.00.2195.6663	59.27 KB (60,688 bytes)	11/3/2004	11/3/2004 3:02:45 PM	Microsoft Corporation		
3:02:52 PM	Microsoft Corporation			c:\winnt\system32\msjint40.dll			
c:\winnt\system32\raschap.dll				msjter40.dll	4.00.6508.0	52.27 KB (53,520 bytes)	
rasppp.dll	5.00.2195.6626	194.27 KB (198,928 bytes)	11/3/2004	11/3/2004 3:02:45 PM	Microsoft Corporation		
3:02:53 PM	Microsoft Corporation			c:\winnt\system32\msjter40.dll			
c:\winnt\system32\rasppp.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			
msjetoledb40.dll	4.00.6807.0	340.27 KB (348,432 bytes)	
11/3/2004 3:02:45 PM Microsoft Corporation			
c:\winnt\system32\msjetoledb40.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			
iasam.dll	5.00.2195.6601	98.27 KB (100,624 bytes)	11/3/2004
3:02:37 PM Microsoft Corporation			
c:\winnt\system32\iasam.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			
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			
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			
iassdo.dll	5.00.2195.6601	263.27 KB (269,584 bytes)	11/3/2004
3:02:37 PM Microsoft Corporation			
c:\winnt\system32\iassdo.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			
db2sec.exe	8.1.6.574	29.11 KB (29,808 bytes)	6/17/2004 11:32:48 PM
International Business Machines Corporation c:\sqlib\bin\db2sec.exe			
db2cli.dll	8.1.6.574	2.77 MB (2,908,222 bytes)	6/17/2004 8:30:24 PM
International Business Machines Corporation c:\sqlib\bin\db2cli.dll			
db2abind.dll	8.1.6.574	244.06 KB (249,920 bytes)	6/17/2004
8:30:18 PM International Business Machines Corporation			
c:\sqlib\bin\db2abind.dll			
db2util.dll	8.1.6.574	1.18 MB (1,237,055 bytes)	6/17/2004 8:31:08 PM
International Business Machines Corporation c:\sqlib\bin\db2util.dll			
db2install.dll	8.1.6.574	28.06 KB (28,738 bytes)	6/17/2004
8:28:20 PM International Business Machines Corporation			
c:\sqlib\bin\db2install.dll			
db2trcapi.dll	8.1.6.574	36.07 KB (36,938 bytes)	6/17/2004
8:28:20 PM International Business Machines Corporation			
c:\sqlib\bin\db2trcapi.dll			
db2locale.dll	8.1.6.574	48.06 KB (49,217 bytes)	6/17/2004
8:28:20 PM International Business Machines Corporation			
c:\sqlib\bin\db2locale.dll			
db2osse.dll	8.1.6.574	312.07 KB (319,561 bytes)	6/17/2004
8:31:10 PM International Business Machines Corporation			
c:\sqlib\bin\db2osse.dll			
db2g11n.dll	8.1.6.574	364.06 KB (372,799 bytes)	6/17/2004
8:28:20 PM International Business Machines Corporation			
c:\sqlib\bin\db2g11n.dll			
db2syp.dll	8.1.6.574	88.06 KB (90,176 bytes)	6/17/2004
8:31:06 PM International Business Machines Corporation			
c:\sqlib\bin\db2syp.dll			
db2wint.dll	8.1.6.574	48.06 KB (49,215 bytes)	6/17/2004
8:31:08 PM International Business Machines Corporation			
c:\sqlib\bin\db2wint.dll			
db2sys.dll	8.1.6.574	2.38 MB (2,490,430 bytes)	6/17/2004 8:31:04 PM
International Business Machines Corporation c:\sqlib\bin\db2sys.dll			
db2app.dll	8.1.6.574	2.51 MB (2,629,694 bytes)	6/17/2004 8:30:20 PM
International Business Machines Corporation c:\sqlib\bin\db2app.dll			
db2jds.exe	8.1.6.574	193.12 KB (197,752 bytes)	6/17/2004 11:30:56 PM
International Business Machines Corporation c:\sqlib\bin\db2jds.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			
msvcps50.dll	5.00.7051	552.50 KB (565,760 bytes)	12/7/1999
7:00:00 AM Microsoft Corporation			
c:\winnt\system32\msvcps50.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			
rasadhlp.dll	5.00.2168.1	7.27 KB (7,440 bytes)	12/7/1999
7:00:00 AM Microsoft Corporation			
c:\winnt\system32\rasadhlp.dll			
winrrn.dll	5.00.2160.1	18.77 KB (19,216 bytes)	12/7/1999
7:00:00 AM Microsoft Corporation			
c:\winnt\system32\winrrn.dll			
rpss.dll	5.00.2195.6702	233.77 KB (239,376 bytes)	11/3/2004
3:02:53 PM Microsoft Corporation			
c:\winnt\system32\rpss.dll			
svchost.exe	5.00.2134.1	7.77 KB (7,952 bytes)	12/7/1999
7:00:00 AM Microsoft Corporation			
c:\winnt\system32\svchost.exe			
rdpwsx.dll	5.00.2195.6697	97.90 KB (100,248 bytes)	11/3/2004
3:02:53 PM Microsoft Corporation			
c:\winnt\system32\rdpwsx.dll			
mstlsapi.dll	5.00.2195.6659	25.77 KB (26,384 bytes)	
11/3/2004 3:02:46 PM Microsoft Corporation			
c:\winnt\system32\mstlsapi.dll			
icaapi.dll	5.00.2195.6654	122.77 KB (125,712 bytes)	11/3/2004
3:02:37 PM Microsoft Corporation			
c:\winnt\system32\icaapi.dll			
regapi.dll	5.00.2195.6602	35.27 KB (36,112 bytes)	11/3/2004
3:02:53 PM Microsoft Corporation			
c:\winnt\system32\regapi.dll			
termsrv.exe	5.00.2195.6696	139.27 KB (142,608 bytes)	
11/3/2004 3:02:58 PM Microsoft Corporation			
c:\winnt\system32\termsrv.exe			
dssenh.dll	5.00.2195.6612	143.77 KB (147,216 bytes)	11/3/2004
3:03:16 PM Microsoft Corporation			
c:\winnt\system32\dssenh.dll			

wshhcpip.dll	5.00.2195.6601	17.27 KB (17,680 bytes)	
11/3/2004 3:03:01 PM Microsoft Corporation			
c:\winnt\system32\wshhcpip.dll			
msafd.dll	5.00.2195.6602	106.27 KB (108,816 bytes)	11/3/2004
3:02:42 PM Microsoft Corporation			
c:\winnt\system32\msafd.dll			
oakley.dll	5.00.2195.6662	435.77 KB (446,224 bytes)	11/3/2004
3:02:49 PM Microsoft Corporation			
c:\winnt\system32\oakley.dll			
mfc42u.dll	6.00.9586.0	988.05 KB (1,011,764 bytes)	11/3/2004
3:02:41 PM Microsoft Corporation			
c:\winnt\system32\mfc42u.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			
scecli.dll	5.00.2195.6704	111.77 KB (114,448 bytes)	11/3/2004
3:02:54 PM Microsoft Corporation			
c:\winnt\system32\scecli.dll			
esent.dll	6.1.3940.31	1.08 MB (1,135,376 bytes)	11/3/2004
3:02:34 PM Microsoft Corporation			
c:\winnt\system32\esent.dll			
mswsock.dll	5.00.2195.6603	62.77 KB (64,272 bytes)	
11/3/2004 3:02:47 PM Microsoft Corporation			
c:\winnt\system32\mswsock.dll			
ntdsatq.dll	5.00.2195.6620	31.27 KB (32,016 bytes)	11/3/2004
3:02:49 PM Microsoft Corporation			
c:\winnt\system32\ntdsatq.dll			
ntdsa.dll	5.00.2195.6697	1016.27 KB (1,040,656 bytes)	11/3/2004
3:02:48 PM Microsoft Corporation			
c:\winnt\system32\ntdsa.dll			
kdcsvc.dll	5.00.2195.6627	144.77 KB (148,240 bytes)	11/3/2004
3:02:40 PM Microsoft Corporation			
c:\winnt\system32\kdcsvc.dll			
sfmapi.dll	5.00.2134.1	38.77 KB (39,696 bytes)	12/7/1999
7:00:00 AM Microsoft Corporation			
c:\winnt\system32\sfmapi.dll			
rassfm.dll	5.00.2195.6604	21.27 KB (21,776 bytes)	11/3/2004
3:02:53 PM Microsoft Corporation			
c:\winnt\system32\rassfm.dll			
rsabase.dll	5.00.2195.6619	129.27 KB (132,368 bytes)	6/19/2003
1:05:04 PM Microsoft Corporation			
c:\winnt\system32\rsabase.dll			
schannel.dll	5.00.2195.6705	144.27 KB (147,728 bytes)	
12/7/1999 7:00:00 AM Microsoft Corporation			
c:\winnt\system32\schannel.dll			
netlogon.dll	5.00.2195.6695	363.27 KB (371,984 bytes)	
11/3/2004 3:02:48 PM Microsoft Corporation			
c:\winnt\system32\netlogon.dll			
kerberos.dll	5.00.2195.6666	207.77 KB (212,752 bytes)	
11/3/2004 3:02:40 PM Microsoft Corporation			
c:\winnt\system32\kerberos.dll			
msprivs.dll	5.00.2195.6695	46.00 KB (47,104 bytes)	11/3/2004
3:02:45 PM Microsoft Corporation			
c:\winnt\system32\msprivs.dll			
samsrv.dll	5.00.2195.6697	380.77 KB (389,904 bytes)	12/7/1999
7:00:00 AM Microsoft Corporation			
c:\winnt\system32\samsrv.dll			
lsasrv.dll	5.00.2195.6695	506.77 KB (518,928 bytes)	12/7/1999
7:00:00 AM Microsoft Corporation			
c:\winnt\system32\lsasrv.dll			
lsass.exe	5.00.2195.6695	32.77 KB (33,552 bytes)	12/7/1999
7:00:00 AM Microsoft Corporation			
c:\winnt\system32\lsass.exe			
ntsapi.dll	5.00.2195.6601	6.77 KB (6,928 bytes)	12/7/1999 7:00:00 AM
Microsoft Corporation			
c:\winnt\system32\ntsapi.dll			
rnr20.dll	5.00.2195.6603	35.77 KB (36,624 bytes)	11/3/2004
3:02:53 PM Microsoft Corporation			
c:\winnt\system32\rnr20.dll			
wmicore.dll	5.00.2195.6611	72.77 KB (74,512 bytes)	
11/3/2004 3:03:01 PM Microsoft Corporation			
c:\winnt\system32\wmicore.dll			
alrsvc.dll	5.00.2134.1	17.77 KB (18,192 bytes)	12/7/1999
7:00:00 AM Microsoft Corporation			
c:\winnt\system32\alrsvc.dll			
trkwks.dll	5.00.2195.6623	88.27 KB (90,384 bytes)	11/3/2004
3:02:58 PM Microsoft Corporation			
c:\winnt\system32\trkwks.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			
srsvcs.dll	5.00.2195.6697	81.77 KB (83,728 bytes)	12/7/1999
7:00:00 AM Microsoft Corporation			
c:\winnt\system32\srsvcs.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			
adslsdp.dll	5.00.2195.6701	130.77 KB (133,904 bytes)	11/3/2004
3:02:26 PM Microsoft Corporation			
c:\winnt\system32\adslsdp.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			
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			
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			
cscui.dll	5.00.2195.6705	237.27 KB (242,960 bytes)	11/3/2004
3:02:31 PM	Microsoft Corporation		
c:\winnt\system32\cscui.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			
cscdll.dll	5.00.2195.6713	98.77 KB (101,136 bytes)	11/3/2004
3:02:31 PM	Microsoft Corporation		
c:\winnt\system32\cscdll.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			
mecat32.dll	5.131.2134.1	7.77 KB (7,952 bytes)	12/7/1999
7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\mec32.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
rpert4.dll 5.00.2195.6701    443.77 KB (454,416 bytes)    11/3/2004
3:02:53 PM    Microsoft Corporation
c:\winnt\system32\rpert4.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
sfccfiles.dll 5.00.2195.6717    948.27 KB (971,024 bytes)    11/3/2004
3:02:55 PM    Microsoft Corporation
c:\winnt\system32\sfccfiles.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
Background Intelligent Transfer Service	BITS	Stopped	Manual	Share Process
Share Process	c:\winnt\system32\svchost.exe -k bitsgroup	Normal	LocalSystem	0
Computer Browser	Browser	Stopped	Manual	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	Running	Auto	Own
Process "c:\sql\lib\bin\db2jds.exe"		Normal	LocalSystem	0
DB2 Security Server	DB2NTSECSERVER	Running	Auto	Own
Process "c:\sql\lib\bin\db2sec.exe"		Normal	LocalSystem	0
Distributed File System	Dfs	Running	Auto	Own
Process c:\winnt\system32\dfsrv.exe		Normal	LocalSystem	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
Normal	Share Process	c:\winnt\system32\dmadmin.exe /com	Normal	LocalSystem
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
c:\winnt\system32\ismserv.exe		Normal	LocalSystem	0
Kerberos Key Distribution Center	kdc	Stopped	Disabled	Share Process
Share Process	c:\winnt\system32\lsass.exe	Normal	LocalSystem	0
Server	lanmanserver	Running	Auto	Share Process
c:\winnt\system32\services.exe		Normal	LocalSystem	0
Workstation	lanmanworkstation	Running	Auto	Share
Process c:\winnt\system32\services.exe		Normal	LocalSystem	0
Site Server ILS Service	LDAPSVCX	Running	Auto	Share Process
Share Process	c:\winnt\system32\inetrv\inetinfo.exe	Normal	LocalSystem	0
License Logging Service	LicenseService	Running	Auto	Own Process
c:\winnt\system32\llsrrv.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
c:\winnt\system32\services.exe		Normal	LocalSystem	0
NetMeeting Remote Desktop Sharing	mnmsrvc	Stopped	Manual	Own Process
Share Process	c:\winnt\system32\mnmsrvc.exe	Normal	LocalSystem	0
Distributed Transaction Coordinator	MSDTC	Running	Auto	Own Process
Share Process	c:\winnt\system32\msdtc.exe	Normal	LocalSystem	0
Windows Installer	MSIServer	Stopped	Manual	Share Process
c:\winnt\system32\msiexec.exe /v		Normal	LocalSystem	0
Network DDE	NetDDE	Stopped	Manual	Share Process
c:\winnt\system32\netdde.exe		Normal	LocalSystem	0
Network DDE DSDM	NetDDEdsdm	Stopped	Manual	Share
Process c:\winnt\system32\netdde.exe		Normal	LocalSystem	0
Net Logon	Netlogon	Stopped	Manual	Share Process
c:\winnt\system32\lsass.exe		Normal	LocalSystem	0
Network Connections	Netman	Running	Manual	Share Process
c:\winnt\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0
File Replication	NtFrs	Stopped	Manual	Own Process
c:\winnt\system32\ntfrs.exe		Ignore	LocalSystem	0
NT LM Security Support Provider	NtLmSsp	Stopped	Manual	Share Process
Share Process	c:\winnt\system32\lsass.exe	Normal	LocalSystem	0
Removable Storage	NtmsSvc	Running	Auto	Share Process
c:\winnt\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0
Plug and Play	PlugPlay	Running	Auto	Share Process
c:\winnt\system32\services.exe		Normal	LocalSystem	0
IPSEC Policy Agent	PolicyAgent	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
Share Process	c:\winnt\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0

Remote Access Connection Manager	RasMan	Running	Manual	Wireless Configuration	WZCSVC	Stopped	Manual	Share
Share Process	c:\winnt\system32\svchost.exe -k netsvcs		Normal	Process	c:\winnt\system32\svchost.exe -k netsvcs		Normal	
LocalSystem	0			LocalSystem	0			
Routing and Remote Access	RemoteAccess	Stopped	Disabled	[Program Groups]				
Share Process	c:\winnt\system32\svchost.exe -k netsvcs		Normal	Group Name	Name	User Name		
LocalSystem	0			Accessories	Default User:Accessories	Default User		
Remote Registry Service	RemoteRegistry	Running	Auto	Accessories\Accessibility		Default User:Accessories\Accessibility		
Own Process	c:\winnt\system32\regsvc.exe	Normal		Default User				
LocalSystem	0			Accessories\Entertainment		Default User:Accessories\Entertainment		
Remote Procedure Call (RPC) Locator	RpcLocator		Stopped	Default User				
Manual Own Process	c:\winnt\system32\locator.exe		Normal	Accessories\System Tools		Default User:Accessories\System Tools		
LocalSystem	0			Default User				
Remote Procedure Call (RPC)	RpcSs	Running	Auto	Startup	Default User:Startup	Default User		
Process	c:\winnt\system32\svchost -k rpcss	Normal		Accessories	All Users:Accessories	All Users		
LocalSystem	0			Accessories\Communications		All Users:Accessories\Communications		
QoS Admission Control (RSVP)	RSVP	Running	Auto	0				
Process	c:\winnt\system32\rsvp.exe -s	Normal	LocalSystem	Accessories\Entertainment		All Users:Accessories\Entertainment		
Security Accounts Manager	SamSs	Running	Auto	0				
Process	c:\winnt\system32\lsass.exe	Normal	LocalSystem	Accessories\Microsoft Script Debugger		All Users:Accessories\Microsoft		
Smart Card Helper	SCardDrv	Stopped	Manual	Script Debugger		All Users		
c:\winnt\system32\scardsvr.exe	Ignore		LocalSystem	0				
Smart Card	SCardSvr	Stopped	Manual	Accessories\System Tools		All Users:Accessories\System Tools		
c:\winnt\system32\scardsvr.exe	Ignore		LocalSystem	0				
Task Scheduler	Schedule	Running	Auto	All Users				
c:\winnt\system32\mstask.exe	Normal		LocalSystem	0				
RunAs Service	seclogon	Running	Auto	ActiveState	ActivePerl 5.8	All Users:ActiveState	ActivePerl 5.8	
c:\winnt\system32\services.exe	Ignore		LocalSystem	0				
System Event Notification	SENS	Running	Auto	All Users				
Process	c:\winnt\system32\svchost.exe -k netsvcs	Normal		Administrative Tools	All Users:Administrative Tools	All Users		
LocalSystem	0			IBM DB2	All Users:IBM DB2	All Users		
Internet Connection Sharing	SharedAccess	Stopped	Manual	IBM DB2\Command Line Tools		All Users:IBM DB2\Command Line Tools		
Share Process	c:\winnt\system32\svchost.exe -k netsvcs		Normal	All Users				
LocalSystem	0			IBM DB2\Development Tools		All Users:IBM DB2\Development Tools		
Simple TCP/IP Services	SimpTcp	Running	Auto	IBM DB2\General Administration Tools		All Users:IBM DB2\General		
Process	c:\winnt\system32\tcpvcs.exe	Normal	LocalSystem	Administration Tools		All Users		
0				IBM DB2\Information		All Users:IBM DB2\Information	All Users	
Print Spooler	Spooler	Stopped	Manual	IBM DB2\Monitoring Tools		All Users:IBM DB2\Monitoring Tools		
c:\winnt\system32\spoolsv.exe	Normal		LocalSystem	0				
Performance Logs and Alerts	SysmonLog	Stopped	Manual	IBM DB2\Set-up Tools		All Users:IBM DB2\Set-up Tools		
Own Process	c:\winnt\system32\smlogsvc.exe	Normal		All Users				
LocalSystem	0			Microsoft Visual C++ 6.0		All Users:Microsoft Visual C++ 6.0		
Telephony TapiSrv	Running	Manual	Share Process	All Users				
c:\winnt\system32\svchost.exe -k tapisrv			Normal	0				
Terminal Services	TermService	Running	Auto	Microsoft Visual C++ 6.0\Microsoft Visual C++ 6.0 Tools		All		
Process	c:\winnt\system32\termsrv.exe	Normal	LocalSystem	Users:Microsoft Visual C++ 6.0\Microsoft Visual C++ 6.0 Tools		All Users		
Telnet	TlntSvr	Stopped	Manual	0				
c:\winnt\system32\tlntsvr.exe	Normal		LocalSystem	0				
Distributed Link Tracking Server	TrkSvr	Stopped	Manual	Startup	All Users:Startup	All Users		
Process	c:\winnt\system32\services.exe	Normal	LocalSystem	0				
Distributed Link Tracking Client	TrkWks	Running	Auto	Accessories	VCLIENT150\TPCC:Accessories			
Process	c:\winnt\system32\services.exe	Normal	LocalSystem	0				
Uninterruptible Power Supply	UPS	Stopped	Manual	VCLIENT150\TPCC				
Process	c:\winnt\system32\ups.exe	Normal	LocalSystem	0				
Utility Manager	UtilMan	Stopped	Manual	Accessories\Accessibility		VCLIENT150\TPCC		
c:\winnt\system32\utilman.exe	Normal		LocalSystem	0				
Windows Time	W32Time	Stopped	Manual	Accessories\Entertainment		VCLIENT150\TPCC:Accessories\Entertainment		
c:\winnt\system32\services.exe	Normal		LocalSystem	0				
World Wide Web Publishing Service	W3SVC	Running	Auto	Accessories\System Tools		VCLIENT150\TPCC:Accessories\System		
Share Process	c:\winnt\system32\inetrv\inetinfo.exe		Normal	Tools	VCLIENT150\TPCC			
LocalSystem	0			Administrative Tools	VCLIENT150\TPCC:Administrative Tools			
Windows Management Instrumentation	WinMgmt	Running	Auto	VCLIENT150\TPCC				
Own Process	c:\winnt\system32\wbem\winmgmt.exe	Ignore		Startup	VCLIENT150\TPCC:Startup	VCLIENT150\TPCC		
LocalSystem	0			Accessories	VCLIENT150\TPCC:Administrator:Accessories			
Windows Management Instrumentation Driver Extensions			Wmi	VCLIENT150\Administrator				
Running Manual Share Process	c:\winnt\system32\services.exe			Accessories\Accessibility		VCLIENT150\TPCC:Administrator:Accessories\Accessibility		
Normal LocalSystem	0			VCLIENT150\Administrator				
Automatic Updates	wuauerv	Stopped	Manual	Accessories\Entertainment		VCLIENT150\TPCC:Administrator:Accessories\Entertainment		
c:\winnt\system32\svchost.exe -k wugroup	Normal		LocalSystem	0				
0				VCLIENT150\Administrator				
				Accessories\System Tools		VCLIENT150\TPCC:Administrator:Accessories\System Tools		
				VCLIENT150\Administrator				
				Administrative Tools	VCLIENT150\TPCC:Administrator:Administrative Tools			
				VCLIENT150\Administrator				

Startup VCLIENT150\Administrator:Startup
VCLIENT150\Administrator

[Startup Programs]

Program	Command	User Name	Location	Startup
synctime	synctime.bat		VCLIENT150\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	5.00.3700.1000
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 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
advpack.dll	5.0.3502.6601	87 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
browsecl.dll	5.0.3700.6661	35 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
browseui.dll	5.0.3700.6661	789 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
ckcnv.exe	5.0.2189.1	9 KB	12/7/1999 8:00:00 AM	C:\WINNT\system32	Microsoft Corporation
comctl32.dll	5.81.3502.6601	538 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
crypt32.dll	5.131.2195.6661	468 KB	6/19/2003 1:05:04 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 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
ieexplore.exe	5.0.2920.0	59 KB	12/7/1999 8:00:00 AM	C:\Program Files\Internet Explorer	Microsoft Corporation

imagehlp.dll	5.0.2195.6613	126 KB	6/19/2003 1:05:04 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 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
jobexec.dll	5.0.0.1	47 KB	12/7/1999 8:00:00 AM	C:\WINNT\system32	Microsoft Corporation
jscrip.dll	5.1.0.8513	476 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
jsproxy.dll	5.0.2920.0	13 KB	12/7/1999 8:00:00 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 1:05:04 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 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
occache.dll	5.0.3502.6601	86 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
ole32.dll	5.0.2195.6692	973 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
oleaut32.dll	2.40.4522.0	612 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
olepro32.dll	5.0.4522.0	160 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
rsabase.dll	5.0.2195.6619	129 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
rsaenh.dll	5.0.2195.6611	132 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
rsapi32.dll	<File Missing>	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 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
shdoc401.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available
shdocvw.dll	5.0.3700.6668	1082 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
shell32.dll	5.0.3700.6705	2328 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
shlwapi.dll	5.0.3502.6601	283 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
url.dll	5.0.3502.6601	82 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
urlmon.dll	5.0.3700.6705	443 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
vbscript.dll	5.1.0.7426	428 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
webcheck.dll	5.0.3502.6601	252 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
win.com	5.0.2134.1	24 KB	12/7/1999 8:00:00 AM	C:\WINNT\system32	Microsoft Corporation
wininet.dll	5.0.3700.6713	456 KB	6/19/2003 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
winsock.dll	3.10.0.103	3 KB	12/7/1999 8:00:00 AM	C:\WINNT\system32	Microsoft Corporation
wintrust.dll	5.131.2195.6624	162 KB	6/19/2003 1:05:04 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 1:05:04 PM	C:\WINNT\system32	Microsoft Corporation
wsock32n.dll	<File Missing>	Not Available	Not 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	28797 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

COM+ Settings

tpccCom.tpcc_com.1:

Activation:

- Enable Object Pooling selected
- Minimum Pool Size: 63
- Maximum Pool Size: 63
- Creating Timeout: 60,000
- Enable Just in Time Activation

Concurrency:

- Concurrency Required

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:00007D00
"dbUserName"="TPCC"
"dbPassword"="tpcc"
"dbInterfacePath"="C:\\inetpub\\wwwroot\\tpcc\\db2glue.dll"
"dIvyThreads"=dword:00000011
```

```
[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,00,00,44,00,41,00,50,00,53,00,56,00,43,00,58,00,00,00,00,00
"PoolThreadLimit"=dword:000000be
"ThreadTimeout"=dword:00015180
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infectrs.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):52,00,50,00,43,00,53,00,53,00,00,00,50,00,72,00,6
f,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,00,00,00,14,00,00,00,30,00,00,00,0
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,00,00,00,20,02,00,00,00,1c,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\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,00,0
0,00,\
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\
ASP\LanguageEngines]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\
ASP\LanguageEngines\PerlScript]
"Write"="$Response->write();"
"WriteBlock"="$Response->writeblock();"
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\
ASP\Parameters]
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\WINNT\System32\inetsrv"
"CertMapList"="C:\WINNT\System32\inetsrv\iisrcmap.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\WINNT\System32\LogFiles"
"AcceptExOutstanding"=dword:00000028
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\
ASP\ADCLaunch]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\
ASP\ADCLaunch\AdvancedDataFactory]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\
ASP\ADCLaunch\RDSServer.DataFactory]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\
Script Map]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\
Virtual Roots]
"/"="c:\inetpub\wwwroot,,207"
"/Scripts"="c:\inetpub\scripts,,1"
"/IISHelp"="c:\winnt\help\iishelp,,1"
"/IISAdmin"="C:\WINNT\System32\inetsrv\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\
Performance]
"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
"WbemAdapFileTime"=hex:00,a0,38,ed,84,36,c3,01
"WbemAdapFileSize"=dword:00001d10
"WbemAdapStatus"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\
Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,0
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,00,00,00,20,02,00,00,00,1c,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\Enum]
"0"="Root\LEGACY_W3SVC\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

RTE Parameters

IBM BenchMaster benchmark profile. DO NOT CHANGE THE SPACING IN THIS FILE!

64 ** Number of slaves (all must be defined directly below)

SEGMENT	MACHINE	LOG DIRECTORY	ODBC
WEBSERVER	DB SERVER	STARTWH	ENDWH
#USERS	SP PREFIX	GROUP1	GROUP2
v11	rte10	c:\rtelogs	tpcc
340	3400	n/a	client10
v12	rte10	c:\rtelogs	tpcc
341	680	3400	n/a
v13	rte10	c:\rtelogs	tpcc
681	1020	3400	n/a
v14	rte10	c:\rtelogs	tpcc
1021	1360	3400	n/a
v15	rte10	c:\rtelogs	tpcc
1361	1700	3400	n/a
v16	rte10	c:\rtelogs	tpcc
1701	2040	3400	n/a
v17	rte10	c:\rtelogs	tpcc
2041	2380	3400	n/a
v18	rte10	c:\rtelogs	tpcc
2381	2720	3400	n/a
v21	vrte160	c:\rtelogs	tpcc
db2serv2_tpcc	2721	3070	3500 n/a client160
n/a			
v22	vrte160	c:\rtelogs	tpcc
db2serv2_tpcc	3071	3420	3500 n/a client160
n/a			
v23	vrte160	c:\rtelogs	tpcc
db2serv2_tpcc	3421	3770	3500 n/a client160
n/a			
v24	vrte160	c:\rtelogs	tpcc
db2serv2_tpcc	3771	4120	3500 n/a client160
n/a			
v25	vrte160	c:\rtelogs	tpcc
db2serv2_tpcc	4121	4475	3550 n/a client160
n/a			
v26	vrte160	c:\rtelogs	tpcc
db2serv2_tpcc	4476	4830	3550 n/a client160
n/a			
v27	vrte160	c:\rtelogs	tpcc
db2serv2_tpcc	4831	5185	3550 n/a client160
n/a			
v28	vrte160	c:\rtelogs	tpcc
db2serv2_tpcc	5186	5540	3550 n/a client160
n/a			
v31	rte20	c:\rtelogs	tpcc
5541	5855	3150	n/a
v32	rte20	c:\rtelogs	tpcc
5856	6170	3150	n/a
v33	rte20	c:\rtelogs	tpcc
6171	6485	3150	n/a
v34	rte20	c:\rtelogs	tpcc
6486	6800	3150	n/a

v35	rte20	c:\rtelogs	tpcc	vclient20A	db2serv1_tpcc
6801	7120	3200	n/a	client20	n/a
v36	rte20	c:\rtelogs	tpcc	vclient20B	db2serv1_tpcc
7121	7440	3200	n/a	client20	n/a
v37	rte20	c:\rtelogs	tpcc	vclient20C	db2serv1_tpcc
7441	7760	3200	n/a	client20	n/a
v38	rte20	c:\rtelogs	tpcc	vclient20D	db2serv1_tpcc
7761	8080	3200	n/a	client20	n/a
v41	vrte150	c:\rtelogs	tpcc	vclient150A	
db2serv2_tpcc	8081	8430	3500	n/a	client150
n/a					
v42	vrte150	c:\rtelogs	tpcc	vclient150B	
db2serv2_tpcc	8431	8780	3500	n/a	client150
n/a					
v43	vrte150	c:\rtelogs	tpcc	vclient150C	
db2serv2_tpcc	8781	9130	3500	n/a	client150
n/a					
v44	vrte150	c:\rtelogs	tpcc	vclient150D	
db2serv2_tpcc	9131	9480	3500	n/a	client150
n/a					
v45	vrte150	c:\rtelogs	tpcc	vclient150A	
db2serv2_tpcc	9481	9835	3550	n/a	client150
n/a					
v46	vrte150	c:\rtelogs	tpcc	vclient150B	
db2serv2_tpcc	9836	10190	3550	n/a	client150
n/a					
v47	vrte150	c:\rtelogs	tpcc	vclient150C	
db2serv2_tpcc	10191	10545	3550	n/a	client150
n/a					
v48	vrte150	c:\rtelogs	tpcc	vclient150D	
db2serv2_tpcc	10546	10900	3550	n/a	client150
n/a					
v51	vrte140	c:\rtelogs	tpcc	vclient140A	
db2serv2_tpcc	10901	11235	3350	n/a	client140
n/a					
v52	vrte140	c:\rtelogs	tpcc	vclient140B	
db2serv2_tpcc	11236	11570	3350	n/a	client140
n/a					
v53	vrte140	c:\rtelogs	tpcc	vclient140C	
db2serv2_tpcc	11571	11905	3350	n/a	client140
n/a					
v54	vrte140	c:\rtelogs	tpcc	vclient140D	
db2serv2_tpcc	11906	12240	3350	n/a	client140
n/a					
v55	vrte140	c:\rtelogs	tpcc	vclient140A	
db2serv2_tpcc	12241	12580	3400	n/a	client140
n/a					
v56	vrte140	c:\rtelogs	tpcc	vclient140B	
db2serv2_tpcc	12581	12920	3400	n/a	client140
n/a					
v57	vrte140	c:\rtelogs	tpcc	vclient140C	
db2serv2_tpcc	12921	13260	3400	n/a	client140
n/a					
v58	vrte140	c:\rtelogs	tpcc	vclient140D	
db2serv2_tpcc	13261	13600	3400	n/a	client140
n/a					
v61	rte30	c:\rtelogs	tpcc	vclient30A	db2serv1_tpcc
13601	13945	3450	n/a	client30	n/a
v62	rte30	c:\rtelogs	tpcc	vclient30B	db2serv1_tpcc
13946	14290	3450	n/a	client30	n/a
v63	rte30	c:\rtelogs	tpcc	vclient30C	db2serv1_tpcc
14291	14635	3450	n/a	client30	n/a
v64	rte30	c:\rtelogs	tpcc	vclient30D	db2serv1_tpcc
14636	14980	3450	n/a	client30	n/a
v65	rte30	c:\rtelogs	tpcc	vclient30A	db2serv1_tpcc
14981	15330	3500	n/a	client30	n/a
v66	rte30	c:\rtelogs	tpcc	vclient30B	db2serv1_tpcc
15331	15680	3500	n/a	client30	n/a

```

v67      rte30      c:\rtelogs tpcc      vclient30C db2serv1_tpcc
15681    16030          3500    n/a      client30    n/a
v68      rte30      c:\rtelogs tpcc      vclient30D db2serv1_tpcc
16031    16380          3500    n/a      client30    n/a
v71      rte40      c:\rtelogs tpcc      vclient40A db2serv1_tpcc
16381    16705          3250    n/a      client40    n/a
v72      rte40      c:\rtelogs tpcc      vclient40B db2serv1_tpcc
16706    17030          3250    n/a      client40    n/a
v73      rte40      c:\rtelogs tpcc      vclient40C db2serv1_tpcc
17031    17355          3250    n/a      client40    n/a
v74      rte40      c:\rtelogs tpcc      vclient40D db2serv1_tpcc
17356    17680          3250    n/a      client40    n/a
v75      rte40      c:\rtelogs tpcc      vclient40A db2serv1_tpcc
17681    18005          3250    n/a      client40    n/a
v76      rte40      c:\rtelogs tpcc      vclient40B db2serv1_tpcc
18006    18330          3250    n/a      client40    n/a
v77      rte40      c:\rtelogs tpcc      vclient40C db2serv1_tpcc
18331    18655          3250    n/a      client40    n/a
v78      rte40      c:\rtelogs tpcc      vclient40D db2serv1_tpcc
18656    18980          3250    n/a      client40    n/a
v81      vrte130    c:\rtelogs tpcc      vclient130A
db2serv2_tpcc      18981    19325    3450    n/a      client130
n/a
v82      vrte130    c:\rtelogs tpcc      vclient130B
db2serv2_tpcc      19326    19670    3450    n/a      client130
n/a
v83      vrte130    c:\rtelogs tpcc      vclient130C
db2serv2_tpcc      19671    20015    3450    n/a      client130
n/a
v84      vrte130    c:\rtelogs tpcc      vclient130D
db2serv2_tpcc      20016    20360    3450    n/a      client130
n/a
v85      vrte130    c:\rtelogs tpcc      vclient130A
db2serv2_tpcc      20361    20710    3500    n/a      client130
n/a
v86      vrte130    c:\rtelogs tpcc      vclient130B
db2serv2_tpcc      20711    21060    3500    n/a      client130
n/a
v87      vrte130    c:\rtelogs tpcc      vclient130C
db2serv2_tpcc      21061    21410    3500    n/a      client130
n/a
v88      vrte130    c:\rtelogs tpcc      vclient130D
db2serv2_tpcc      21411    21760    3500    n/a      client130
n/a

1000    ** Connect rate - rate users log in to the database (users per minute)

340     ** 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)****

21760   ** Total number of warehouses

0       *** Run type (0 = 3-tier, 1 = 2-tier)****

0       ** 2-tier run options (BITFIELD: 1=use tpcc_neworder_new;
2=fetch transaction results)

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.

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	NEWORDERPAYMENT	DELI	STCKLVLORDSTAT
0	44950	43020	4010
0	12030	12030	5030
0	0	0	0
0	18000	3000	2000
0	0	0	0
0	5000	5000	20000
0	100	100	100
0	100	100	100

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	21,760
Measured TpmC	273,520

Table	Rows	Table	Index	5% Space	Total Space
Warehouse	21,760		7	0	0
District	217,600		34	0	2
Item	100,000		10	0	1
Stock	2,176,000,000	708,520		0	35,426
Customer	652,800,000	510,200	31,760	0	27,098
New-Order	195,840,000	15,088		0	754
Orders	652,800,000	24,007	18,304	0	0
Order-Line	6,528,000,000	428,730		0	0
History	652,800,000	40,320		0	0
Additional Overhead		364,463			364,463

Free Space	65,816				
Dynamic Space	493,057				
Static Space	1,711,667				
Daily Growth	99,162				
Daily Spread	0				
				<u>30 Minute log Computations</u>	
				Log Written (KB)	19,533,704
				New-Order Txns	8,149,399
				Log Written per New-Order (KB)	2.40

Data Storage Requirement

60 Days (MB)	7,661,414
60 Days (GB)	7,482

Log Storage Requirement

8 Hours (GB)	300.12
--------------	--------

Disk Sizing

Disk Type	Formatted		SUT		Priced	
	Capacity (GB)	# of Disks	Capacity (GB)	# of Disks	Capacity (GB)	# of Disks
DB FastT 36GB	33.40	560	18,704	560	18,704	560
LOG SCSI 36GB (RAID10)	33.90	20	339	20	339	20
OS SCSI 36GB (RAID1)	33.90	2	34	2	34	2

Total Capacity						19,077
-----------------------	--	--	--	--	--	--------

Appendix E: Third-Party Quotations



Protect Your Data - Grow Your Business

To:
Attention:
Phone:
Fax:
Email:

From: Alan Powers
Phone: (248)223-1020 x344
Fax: (248)223-1026
Email: apowers@compsat.com

QUOTE # : 4W460022306
DATE: February 23, 2006

IBM x460 Configuration

Part No.	Description	Qty	List Price		Compsat Discounted Price	
			(per unit) US Dollar	(quantity x unit price) US Dollar	(per unit) US Dollar	(quantity x unit price) US Dollar
x460 SERVER						
8872-6RU	xSeries 460 with 2 x Intel Xeon Processor 7040 3.00GHz/2x2MB L2 Cache	1	\$25,999.00	\$25,999.00	\$21,319.18	\$21,319.18
8874-1RU	MXE-460 (0 processors, 0 memory, 2 Memory Cards)	1	\$6,999.00	\$6,999.00	\$5,739.18	\$5,739.18
13M7409	Active Memory 4-Slot Memory Expansion Card	4	\$499.00	\$1,996.00	\$439.12	\$1,756.48
96P2688	3 YR onsite repair 24x7x4 hour (x460 and MXE-x460)	2	\$3,390.00	\$6,780.00	\$2,983.20	\$5,966.40
30R5145	8GB (2x4GB) PC2-3200 CL3 2RX4 ECC DDR2 SDRAM RDIMM	16	\$5,299.00	\$84,784.00	\$4,663.12	\$74,609.92
13M7414	Scalability Cable 2.3M	2	\$299.00	\$598.00	\$263.12	\$526.24
25R8942	Intel Xeon Processor 7040 3.00GHz/2x2MB L2 Cache	2	\$5,699.00	\$11,398.00	\$5,015.12	\$10,030.24
32P0033	IBM ServeRAID-6M Ultra320 SCSI Adapter	1	\$879.00	\$879.00	\$773.52	\$773.52
633147N	E54 15" Colour Monitor (Stealth Grey)/MPRII	1	\$139.00	\$139.00	\$122.32	\$122.32
30L9183	3 YR onsite exch. 24x7x4 hour (E54 Monitor)	1	\$90.00	\$90.00	\$79.20	\$79.20
73P2620	ThinkPlus Enhanced Performance USB Keyboard	1	\$39.00	\$39.00	\$34.32	\$34.32
90P0742	ScrollPoint 800 DPI Optical Mouse - USB & PS/2	1	\$25.00	\$25.00	\$22.00	\$22.00
EXP710(s)						
24P0960	IBM TotalStorage DS4000 Host Bus Adapter	8	\$1,485.00	\$11,880.00	\$1,306.80	\$10,454.40
174290U	IBM TotalStorage DS4500 Midrange Disk Subsystem	4	\$49,900.00	\$199,600.00	\$40,918.00	\$163,672.00
96P2062	3 YR onsite repair 24x7x4 hour (DS4500)	4	\$1,087.00	\$4,348.00	\$956.56	\$3,826.24
19K1269	IBM DS4000 Mini Hub	8	\$899.00	\$7,192.00	\$791.12	\$6,328.96
22R0483	IBM Short Wave SFP Module (Pack of 4)	42	\$550.00	\$23,100.00	\$484.00	\$20,328.00
19K1247	IBM 1m LC-LC Fibre Channel Cable	72	\$79.00	\$5,688.00	\$69.52	\$5,005.44
19K1248	IBM 5m LC-LC Fibre Channel Cable	16	\$129.00	\$2,064.00	\$113.52	\$1,816.32
1740710	IBM TotalStorage DS4000 EXP710 Storage Exp. Unit	40	\$6,000.00	\$240,000.00	\$4,920.00	\$196,800.00
41L2768	3 YR onsite repair 24x7x4 hour (EXP710)	40	\$760.00	\$30,400.00	\$668.80	\$26,752.00
06P5772	2Gbps FC 36.4GB 15K Hot-Swap HDD	560	\$1,115.00	\$624,400.00	\$981.20	\$549,472.00
EXP400(s)						
17331RU	IBM EXP400 Storage Expansion Enclosure	2	\$3,099.00	\$6,198.00	\$2,541.18	\$5,082.36
41L2768	3 YR onsite repair 24x7x4 hour (EXP400)	2	\$760.00	\$1,520.00	\$668.80	\$1,337.60
21301TX	IBM UPS 750TLV	1	\$299.00	\$299.00	\$263.12	\$263.12
03K9310	2M SCSI Cable	2	\$75.00	\$150.00	\$66.00	\$132.00
90P1380	36.4GB 15K Ultra320 SCSI Hot-Swap HDD	22	\$279.00	\$6,138.00	\$245.52	\$5,401.44
93074SX	IBM S2 42U Standard Rack	5	\$1,489.00	\$7,445.00	\$1,310.32	\$6,551.60
41L2760	3 YR onsite exch. 24x7x4 hour (Rack)	5	\$300.00	\$1,500.00	\$264.00	\$1,320.00
x226 SERVER(s)						
8648-6AU	x226 with 3.4GHz/2MB Xeon DP, 512MB (2x256MB) Memory	8	\$1,499.00	\$11,992.00	\$1,229.18	\$9,833.44
13N0666	3.4GHz/2MB Xeon DP Processor Upgrade	8	\$879.00	\$7,032.00	\$773.52	\$6,188.16
73P3522	1GB (2x512MB) PC-3200 DDR2 ECC SDRAM RDIMM	8	\$399.00	\$6,384.00	\$351.12	\$5,617.92
90P1380	36.4GB 15K Ultra320 SCSI Drive	16	\$279.00	\$4,464.00	\$245.52	\$3,928.32
73P5101	PRO/1000 GT Dual-Port Server Adapter by Intel	16	\$249.00	\$3,984.00	\$219.12	\$3,505.92
633147N	E54 15" (13.8" Viewable) Color Monitor	8	\$139.00	\$1,112.00	\$122.32	\$978.56
96P2250	ServicePac for 3-Year 24x7x4 Support (x226)	8	\$586.00	\$4,688.00	\$515.68	\$4,125.44
30L9183	ServicePac for 3-Year 24x7x4 Support (Monitor)	8	\$90.00	\$720.00	\$79.20	\$633.60
TOTAL =				\$1,349,792.00		\$1,158,369.68

14.18%

25330 Telegraph Road / Suite 200 Raleigh Officentre / Southfield, Michigan 48034
Phone: 248-223-1020 / Fax: 248-223-1026 / www.compsat.com

NOTE:

- This quote may include Compsat Technology consulting and configuration charges.
- Mfg. pricing is out of our control and could change without notice.
- Pricing good for 30 Days from date quoted.



March 1, 2006

IBM Corporation
Ms. Celia Schreiber
xSeries Performance

Dear Celia:

The table shown below lists the U.S. pricing for DB2 Universal Database Enterprise Server Edition product that has been used in the TPC-C Benchmark.

All prices shown are in U.S. Dollars.

DB2 Enterprise Server Edition (ESE)	Qty	Reference Price per unit	Total Reference price
SW License & 12 Months Maintenance	4	23,902	95,608
SW Maintenance Renewal - 1 year	8	1,138	9,104
		Sub-total reference price for DB2 ESE:	104,712
		TOTAL REFERENCE PRICE:	104,712

Any and all prices herein are suggested prices only and are subject to change at IBM's sole discretion. Products listed herein are subject to withdrawal or modification by IBM at any time at IBM's sole discretion.

Sincerely,

Richard Hughes
IBM Sales & Distribution, Software Sales
Americas Sales Executive DB2 and Informix
212-493-2065
rhughes@us.ibm.com



> My Shopping Cart

MY SHOPPING CART

MY SAVED ITEMS

[Returns](#) | [Privacy](#) | [Security](#)

Clear Cart Move Cart to Wish List Save Cart Print Cart Email Cart

Switches

Qty.	Product Description	Unit Price	Savings	Total Price
1	NETGEAR GS108 10/100/1000Mbps Copper Gigabit Switch - Retail Model #: GS108 Item #: N82E16833122111 <i>** This item is warranted through the product manufacturer only. ?</i> \$10.00 Mail-in Rebate Select An Optional Extended Warranty Plan <input type="checkbox"/>	\$75.99		\$75.99
<p> Remove Save Move To Wish List</p>				

Network - Cables

Qty.	Product Description	Unit Price	Savings	Total Price
1	AMC CC5E-B14B 14 FT Cat 5E Blue Cat 5E Blue Cable - OEM Model #: CC5E-B14B Item #: N82E16812105305	\$3.29		\$3.29
<p> Remove Save Move To Wish List</p>				

Subtotal: \$79.28

Shipping: \$0.00

Shipping

Zip Code: UPS Guaranteed 3 Day Service

*Enter your Zip Code and select a shipping option to determine your shipping cost.

Redeem Gift Certificates

Claim Code:
Security Code:

Gift Certificates: \$0.00

Apply Promo Code

Promo Code:

Total(before tax): \$79.28

If you're experiencing problems with your shopping cart, please click here and try again.

Note: Your shopping cart will be emptied.

IMPORTANT SHIPPING INFORMATION

- All orders require 1-2 business days of processing time prior to shipping.
- FedEx Express Saver - Delivery within 3 business days.
- FedEx 2nd Day - Delivery within 2 business days.
- FedEx Standard Overnight - Delivery in 1 business day.
- UPS Guaranteed 3 Day Service - Delivery within 3 business days.