Sequent NUMACenter TM
Using DYNIX/ptx 4.4.4 and
Oracle8 Enterprise EditionTM

TPC Benchmark™C
Full Disclosure Report

First Edition
18-Dec-1998

Prepared by
Basker Shanmugam, Horace Tong, Tommy Tse

First Printing, *18-Dec-1998*

Sequent Computer Systems, Inc. believes that the information included in this document is accurate as of the publication date. The information in this document is subject to change without notice. Furthermore, Sequent Computer Corporation is not responsible for any errors contained within this document.

The pricing information given in this FDR is accurate as of the publication date, 10-Nov-1998 and is generally available, at the availability date.

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

All performance data contained in this report were obtained in a rigorously controlled environment. Actual performance experienced by a particular customer may vary due to differences in system layout and configuration, hardware and/or software revision levels, and background system activity. The content of this document is for informational purposes only.

NUMACenter and NUMA-Q are trademarks of Sequent Computer Systems, Inc.

Microsoft Windows NT is a registered trademark of Microsoft Corporation.

Oracle is a register trademark of Oracle Corporation.

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

Intel and Pentium are registered trademarks of Intel Corporation.

Other product names mentioned in this document may be trademarks and/or registered trademarks of their respective companies.

Abstract

1.1 Overview

This report documents the methodology and results of the TPC Benchmark™ C test conducted on a Sequent Computer Systems, Inc. NUMACentre 2000 NE300. The operating system used for the benchmark was DYNIX/ptx® 4.4.4 for the database server and Microsoft® Windows NT Server 4.0 on the clients. The database was the Oracle® 8.0.4 Enterprise Edition. Tuxedo® 6.3 Core Functional Services (CFS) provided the database connection queues from the client. All tests were done in compliance with Revision 3.4 of the Transaction Processing Council's TPC Benchmark™ C Standard Specification. Two standard TPC Benchmark™ C metrics, transactions per second (tpmC) and price per tpmC ($/tpmC) are reported and referred to in this document. The results from the tests are summarized below.

Hardware	Software	Total System Cost	tpmC	$/tpmC	Availability Date
Sequent Computer Systems, Inc. NUMACentre	DYNIX/ptx 4.4.4, Oracle 8.0.4 Enterprise Edition, Tuxedo 6.3 CFS	$12,363,684	93,900.85	$131.67/tpmC	HW: 15-Jun-1999. SW: Currently available.

1.2 Auditor

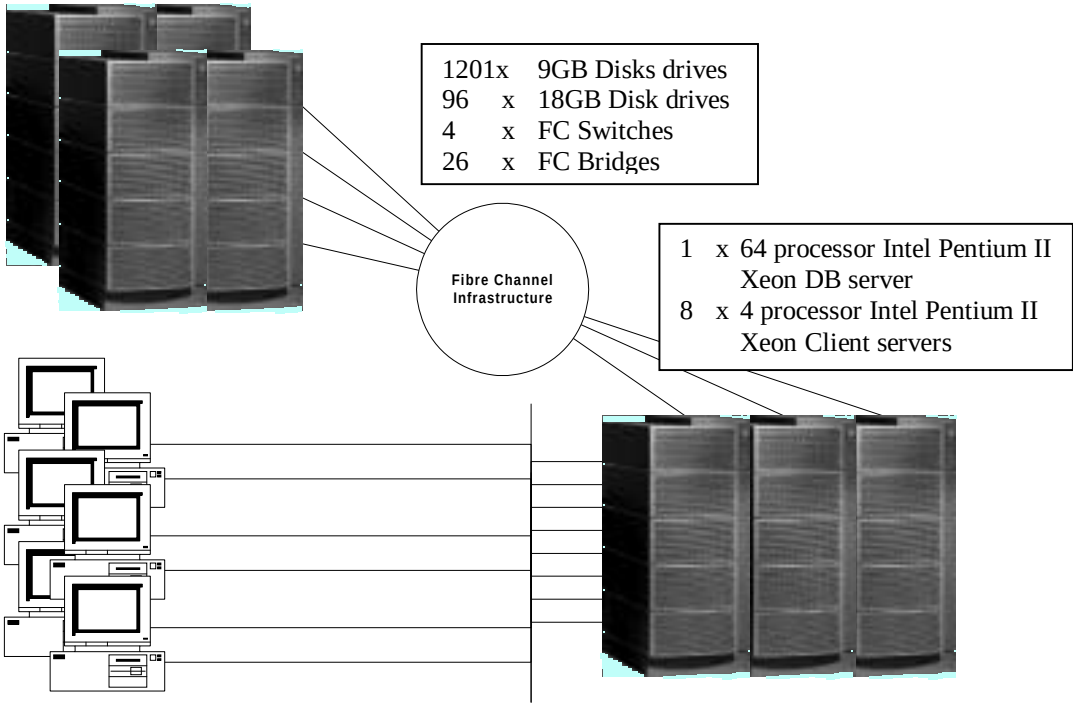
The results of the benchmark and test methodology used to produce the results were audited by Tom Sawyer of Performance Metrics, Inc. and have fully met the TPC-C rev 3.4 specifications.

Additional copies of this Full Disclosure Report can be obtained from Sequent Computer Systems, Inc. at the following address:

Sequent Computer Systems, Inc.
15450 SW Koll Parkway
Beaverton, OR 97006-6063
Attention: Martin O'Sullivan

SEQUENT / ORACLE	NUMACenter 2000 (C/S)		TPC-C Rev 3.4 Report Date 18-Dec-1998
Total System Cost	**TPC-C Throughput**	**Price/Performance**	**Availability Date**
$12,363,684	**93,900.85**	**$131.67/tpmC**	15-Jun-1999

Processors	Database Manager	OS	Other Software	Number of Users
64 x Intel Pentium II Xeon Processors 405MHz	Oracle 8.0.4 Enterprise Edition	DYNIX/ptx 4.4.4	Tuxedo 6.3 CFS Microsoft Internet Information Server Microsoft Visual C++	80,000



System Component	Server		Each Client	
Processors Cache	64	Intel Pentium II Xeon® 405MHz 2MB	4	Intel Pentium II Xeon® @ 400MHz 512 KB
Memory	1	64GB	1	1GB
Disk Controllers	32 32	Fiber controllers SCSI controllers	 1	 Adaptec On-Board
Disk Drives	1201 96	9 GB (8.473 GB useable) 18GB disks (17.1 useable)	1	9GB
Total Storage		12582GB		9GB
Other	15 1 1	PCI 1-port Ethernet NIC PCI 4-port Ethernet NIC Tape Drive	1 4	CD-ROM PCI 4-port Ethernet NIC

SEQUENT / ORACLE	NUMACenter 2000	TPC-C Rev 3.4 Report Date 18-Dec-1998

Description	Part Number	Brand[1]	Unit Price	Qty	Extended Price	5 yr. Maint. Price
Console						
NUMA-Q Console the ethernet	CON-0000	1	6,500	1	6,500	
Server Hardware						
Enterprise (large) NUMA-Q cabinet	CABE-0000-00	1	6,500	4	26,000	
Boot Pbay	EXP-PBAY-04	1	3,900	1	3,900	
CDROM	CDR-0004-01	1	2,080	1	2,080	
SCSI Pbay cable	CBL-0300-RL	1	91	32	2,912	
PCI SCSI Adapter	IOC-0120-51	1	715	32	22,880	
Xeon Quad w/ IQ-LINK 405MHz 2MB	QUAD-0301	1	159,250	16	2,548,000	
IQ Plus-IQ Link interconnect cable	CBL-PLUS-03	1	455	16	7,280	
8G Addon Memory for Xeon Quad	MEM-8100-00	1	127,400	6	764,400	
Single port ethernet adapter	COM-0041-50	1	374	15	5,610	
Four port ethernet adapter	COM-0040-50	1	975	1	975	
Rev 4 Fibre Channel PCI Host adapter	IOC-0210-54	1	2612	32	83,584	
Server Hardware total					**3,474,121**	
Storage Subsystem						
48 x 9GB disk Storage unit - Includes 1 FC Bridge, 4 PBAYs, Cables, Cabinet	DSK-0948-40	1	84,500	25	2,112,500	
48 x 18GB disk Storage unit - Includes 1 FC Bridge, 4 PBAYs, Cables, Cabinet	DSK-1848-40	1	123,500	2	247,000	
9gb Add-on disk drive	DSK-0009-40	1	1,690	1	1,690	
Add-on PBAY	EXP-PBAY-00	1	3,900	7	27,300	
Fibre Channel Switch	FCS-0016-02	1	32,435	4	129,740	
Storage Subsystem total					**2,518,230**	
Server Hardware total					**5,992,351**	**1,198,470**
Server Software						
DYNIX/ptx V4.4.2 CD-ROM Media (NAO)	PTX-C440-00	1	0	1	0	
Sequent Volume Manager for Base Quad	SVM-444-001	1	3,750	1	3,750	
Sequent Volume Manager for Additional Quads	SVM-444-ADD	1	3,750	15	56,250	
Server software subtotal					**60,000**	**12,000**
Oracle8 DataBase licenses w/ Parallel Server and Partitioning Options	ORA8.0.4	2	2,069,841	1	2,069,841	2,069,841
Server Software total					**2,129,841**	**2,081,841**
Client Hardware						
Xeon 4x 400MHz 512K cache 1G mem	NTM-N300-00	1	40,860	8	326,880	
4-port Ether	COM-0040-50	1	975	32	31,200	
Client Hardware Subtotal					**358,080**	**71,616**
Client Software						
MS Windows NT Server 4.0 w/5 CAL	OS-NT4-00	1	809	8	6,472	1,940
Tuxedo 6.3 Core Functionality Services for NT		3	12,000	8	96,000	72,000
Visual C++ 32 bit Edition		Microsoft	499	1	499	150
Client Software subtotal					**102,971**	**74,090**
User Connectivity						
COMPEX MicroHub/8 8+1 A11Port 10BaseT Hub (including 10% spare)	DEH2924	Compex	33	11,000	363,000	
User Connectivity subtotal					**363,000**	
Total cost of hardware and software					**8,941,955**	**3,421,729**
	Five-Year Cost of Ownership:				**$12,363,684**	
	TPM-C rating				**93,900.85**	
	$ / tpmC:				**131.67**	

Audited by Tom Sawyer, Performance Metrics

Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.

[1] Pricing sources – 1=Sequent, 2=Oracle, 3=BEA

Numerical Quantities Summary for the Sequent NUMACentre 2000 NE300

MQTh, computed Maximum Qualified Throughput						**93,900.85tpm-C**
% throughput difference, reported & reproducibility runs						<1%
Response Times (in seconds)			Average		90th	Max
- Neworder			0.90		1.600	133.32
- Payment			0.71		1.300	89.23
- Order Status			0.76		1.400	6.15
- Delivery (interactive portion)			0.70		1.100	6.11
- Delivery (deferred portion)			1.01		1.493	228.20
- Stock-Level			1.15		2.100	6.94
- Menu			0.20		0.440	10.29
Response time delay added for emulated components						Menu 0.1 Resp 0.1
Transaction Mix, in percent of total transactions						
- New-Order						44.78%
- Payment						43.07%
- Order-Status						4.04%
- Delivery						4.04%
- Stock-Level						4.05%
Keying/Think Times (in seconds),	Min		Average		Max	
- New-Order	18.88	0.00	18.89	12.02	19.22	120.39
- Payment	3.88	0.00	3.89	12.02	4.17	120.44
- Order-Status	2.88	0.00	2.89	10.07	3.13	100.86
- Delivery	2.88	0.00	2.89	5.03	3.08	50.37
- Stock-Level	2.88	0.00	2.89	5.03	3.09	50.36
Test Duration						
- Ramp-up time						46 minutes
- Measurement interval						120 minutes
- Number of checkpoints						4
- Checkpoint interval						30 minutes
- Number of transactions (all types) completed in measurement interval						25,161,760

Table of Contents

2 Introduction

2.1 Document Structure

The contents of this report are determined by the TPC Benchmark C Standard Specification Revision 3.4, written and approved by the Transaction Processing Performance Council (TPC). The format of this report is based on this specification. Most sections of this report begin with the specification requirements printed in italic type, immediately followed by the detail in plain type of how Sequent Computer Systems, Inc. complied with the specification. Where extensive listings are required (such as listing of code), a note is included which references an appendix containing the listing.

2.2 Benchmark Overview

TPC Benchmark™ C (TPC-C) is an 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 such environments, which are characterized by:

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

Although these specifications express implementation in terms of a relational data model with conventional locking scheme, the database may be implemented using any commercially available database management system (DBMS), database server, file system, or other data repository that provides a functionally equivalent implementation. The terms "table", "row", and "column" are used in this document only as examples of logical data structures.

TPC-C uses terminology and metrics that are similar to other benchmarks, originated by the TPC or others. Such similarity in terminology does not in any way imply that TPC-C results are comparable to other benchmarks. The only benchmark results comparable to TPC-C are other TPC-C results conformant with the same revision.

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.

2.3 System Overview

The system used in this TPC test is the basis of Sequent's NUMACenter architecture that integrates both Unix and Windows NT. The NT configuration comprised of 8 independent instances of version 4.0 of the operating system, each running on a 4 processor node. These 8 instances hosted the TPC application code. The 64 processor single instance of Dynix/ptx 4.4.4 hosted the Oracle 8.0.4 database software.

Twelve remote terminal emulators (RTE) emulated 80,000 users executing the standard TPC-C workload. The RTE's were connected to the Client partitions via 10BaseT hubs.

The I/O subsystem comprised 4 Fiber Channel switches connected to 26 Bridges, supporting 192 single-initiated and 8 dual-initiated SCSI channels. Each SCSI channel has 6 disks. In addition, 32 SCSI channels were configured to support the database log requirement.

3 General Items

3.1 Test Sponsor

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

Sequent Computer Systems, Inc. and Oracle Corporation were joint sponsors of this TPC Benchmark™ C.

3.2 *Application Code and Definition Statements*

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

The application consists of a Remote Terminal Emulator (RTE) program emulating a set of users entering TPC-C transactions through web browsers, and communicating with Client nodes within the NUMACenter system complex running the Microsoft Internet Information Server (IIS) web server. The Client nodes use the BEA Tuxedo™ transaction monitor to communicate with the database server node.

The Remote Terminal Emulator program is a custom, multithreaded, C program split into an RTE master program and an RTE slave program. A single RTE master program runs on one NUMA-Q 2000 four processor, Pentium Pro Processor system running NT 4.0 and drives 96 RTE slave processes (one per network segment), distributed across 12 other NUMA-Q 2000 four processor systems also running NT 4.0. The RTE master consists of a main program that creates and controls one thread for each RTE slave process. These threads communicate through named pipes to the RTE slave processes. In turn, each RTE slave process consists of a main program which creates and controls one thread for each user it will emulate. Each thread opens up a persistent socket to the web server running in one of the Client partitions, then issues transactions according to the specified parameters. Transaction data is stored in a data structure which is saved to a file at the end of the run, or when selected by the test operator. The RTE parameters – login rate, ramp up rate, run time, screen update interval – as well as the transaction parameters – mix, key time, think time, etc. – are specified in two files. All data from the run in progress, including average tpmC over the last 1, 5, 10 and 30 minutes are shown on a single console.

On each Client partition, IIS loads a custom Microsoft Internet Information Server Application Programming Interface dynamic link library (ISAPI DLL) program that communicates with the emulated web browsers through the HTTP protocol and with the database server through the Tuxedo transaction monitor and the Oracle OCI interface. The ISAPI DLL, tpcc.dll, supplies fill-in screens to the user for each transaction, then parses the data in each request, makes a Tuxedo call and hands the data to the appropriate Tuxedo server. Tuxedo manages the request in its queue and then makes an OCI call to the database server running Oracle. The resulting data is passed back to the tpcc.dll where it is formatted into HTML and sent back to the user's browser. The C program tpcc.dll features efficient, robust input handling and user screen management.

The connections between the Client application and the Database server are implemented with a multithreaded version of Tuxedo 6.3 Core Functionality Services for NT. A single executable is implemented to handle all TPC-C transaction types, and multiple instances of this transaction server are started by Tuxedo prior to the start of the simulation. Deferred Delivery transactions are handled by asynchronous Tuxedo calls in which control is returned immediately to the ISAPI DLL while Tuxedo completes the Delivery database stored procedure. The other 4 transactions are handled by synchronous Tuxedo calls which don't return to tpcc.dll until the database access is complete.

The web Client and Tuxedo server code is listed in Appendix A.

3.3 Parameter Settings

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

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

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

Appendix C contains all the Oracle 8.0.4, Windows NT Server 4.0, Dynix Ptx 4.4.4, Internet Information Service and Tuxedo parameters used in this benchmark.

3.4 *Configuration Diagrams*

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

Figure 1 and 2 show the measured and priced full configurations. The system under test (SUT) in the measured system was identical to the priced one. The only differences are the use of RTEs.

Figure 1: Measured Configuration

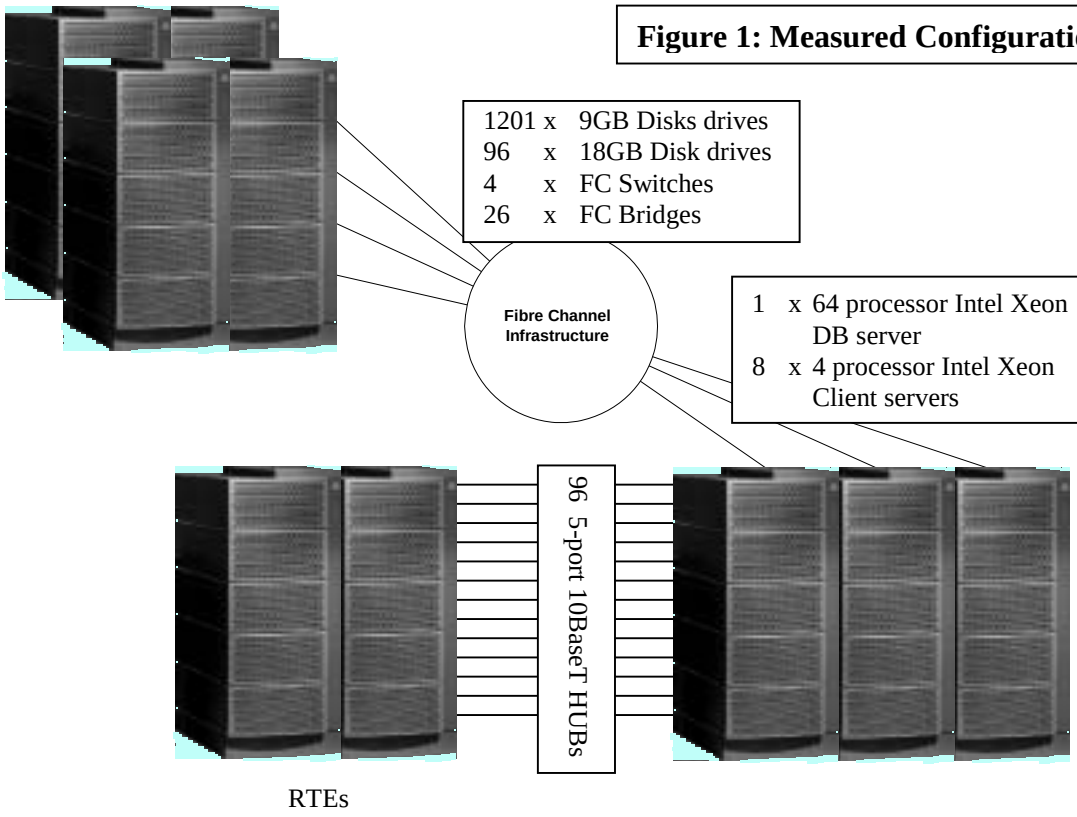
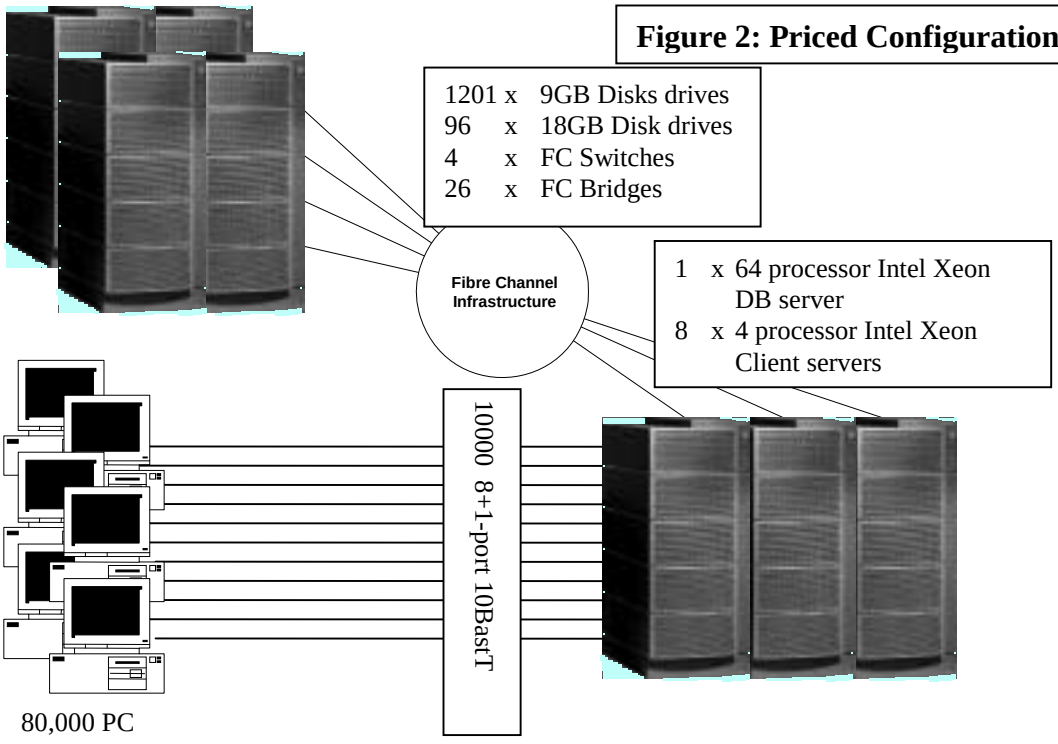


Figure 2: Priced Configuration



4 Clause 1 -- Logical Database Design Related Items

4.1 Table Definitions

Listings must be provided for all table definition statements and all other statements used to set-up the database. (8.1.2.1)

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

4.2 *Physical Organization of the Database*

The physical organization of tables and indices, within the database, must be disclosed. (8.1.2.2)

The measured configuration used 1200 disk drives for data and 96 disk drives for log. The organization is shown in Table 5, database build scripts are shown in Appendix B.

4.3 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. (8.1.2.3)

Insert and delete functionality was fully operational during the benchmark.

4.4 *Horizontal and Vertical Partitioning*

While there are a 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. (8.1.2.4)

Horizontal partitioning was used on all tables and indexes other than the CUSTOMER and STOCK tables and their associated indexes. Horizontal partitioning was implemented using functionality provided by Oracle8.

4.5 Replication

Replication of tables, if used, must be disclosed (see Clause 1.4.6). (8.1.2.5)

Replication was not used in this benchmark.

4.6 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). (8.1.2.6)

No additional attributes were used in this benchmark.

5 Clause 2 -- Transaction and Terminal Profiles Related Items

5.1 Random Number Generation

The method of verification for the random number generation must be described. (8.1.3.1)

Random numbers for transaction distributions, etc., were generated inside the RTE using a Lehmer random number generator which returns a pseudo-random real number uniformly distributed between 0.0 and 1.0. The method is described in "Random Number Generators: Good Ones Are Hard to Find", by Steve Park and Keith Miller in **Communications of the ACM,** October, 1988. Sample code is available from ftp://cs.wm.edu/pub/rngs.tar. During the RTE audit the auditor verified generated random numbers conformed to required distributions.

5.2 Screen Layout

The actual layouts of the terminal input/output screens must be disclosed. (8.1.3.2)

The screen layouts are based on those in Clauses 2.4.3, 2.5.3, 2.6.3, 2.7.3, and 2.8.3 of the TPC-C Standard Specification. There are some very minor differences based on the fact that this is a web client implementation.

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

The terminal features were verified by allowing the auditor to manually execute each of the five transaction types, using Microsoft Internet Explorer version 3.0.

5.4 Intelligent Terminals

Any usage of presentation managers or intelligent terminals must be explained. (8.1.3.4)
***Comment 1**: The intent of this clause is to describe any special manipulations performed by a local terminal or workstation to off-load work from the SUT. This includes, but is not limited to: screen presentations, message bundling, and local storage of TPC-C rows.*
***Comment 2**: This disclosure also requires that all data manipulation functions performed by the local terminal to provide navigational aids for transaction(s) must also be described. Within this disclosure, the purpose of such additional function(s) must be explained.*

Application code involved in the manipulation of data was run on the client. Screen manipulation commands in the form of HTML were downloaded to the web browser which handled input and output presentation graphics. A listing of this code is included in Appendix A. Microsoft Internet Information Service assisted in the processing and presentation of this data.

5.5 Transaction Profiles

The percentage of home and remote order-lines in the New-Order transactions must be disclosed. (8.1.3.5)
The percentage of New-Order transactions that were rolled back as a result of an unused item number must be disclosed. (8.1.3.6)
The number of items per orders entered by New-Order transactions must be disclosed. (8.1.3.7)
The percentage of home and remote Payment transactions must be disclosed. (8.1.3.8)

The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the database must be disclosed. (8.1.3.9)
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. (8.1.3.10)

Table 1: Transaction Statistics

Transaction	Function	Value
New Order	Home Warehouse Items	99.00%
	Remote Warehouse Items	1.00%
	Rolled Back Transactions	0.99%
	Average Lines Per Order	10.00
Payment	Home Warehouse	85.01%
	Remote Warehouse	14.99%
	Non-Primary Key Access	60.03%
Order Status	Non-Primary Key Access	60.02%
Delivery	Skipped Transactions	0

5.6 Transaction Mix

The mix (i.e., percentages) of transaction types seen by the SUT must be disclosed. (8.1.3.11)

Table 2: Transaction Mix

Transaction	Percentage
New Order	44.78%
Payment	43.07%
Order Status	4.04%
Delivery	4.04%
Stock Level	4.05%

5.7 Deferred Delivery Mechanism

The queuing mechanism used to defer the execution of the Delivery transaction must be disclosed. (8.1.3.12)

The client application submits delivery transactions to asynchronous Tuxedo queues running on the client machines. There were multiple delivery servers with single execution threads running on each client machine. These delivery servers were responsible for processing deliveries queued to Tuxedo and submitting them to the database server.

The source code is listed in Appendix A.

6 Clause 3 -- Transaction and System Properties Related Items

6.1 ACID Tests

The results of the ACID tests 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. (8.1.4.1)

The SUT and client systems in this benchmark used the same hardware platform running the same system and RDBMS versions, as the one used in the Sequent NUMACenter 2000 TPC-C benchmark publish on 13-Oct-1998. The only ACID test reran for this benchmark was the Instantaneous Interruption and Loss of Memory test. For details of the execution of other ACID tests, please refer to the Sequent NUMACenter 2000 TPC-C Full Disclosure Report publish on 13-Oct-1998.

Instantaneous Interruption and Loss of Memory

An instantaneous system interruption was caused by powering down the entire system. The test was executed on a fully scaled database of 8,704 warehouse under a full load of 87,040 users. The following steps were executed:

1. The sum of D_NEXT_O_ID was taken.
2. 87040 users were logged on to the database and ran transactions for 5 minutes after steady state.
3. The server was powered down.
4. The RTE was allowed to continue running. Completed transactions enroute from the clients were recorded. Error messages from the clients reporting loss of network to the server began appearing in the RTE log and screen.
5. The RTE was eventually shut down.
6. The server was powered on. Oracle was restarted and automatically recovered.
7. A new count of D_NEXT_O_ID was taken.
8. This number was compared with the number of new orders reported by the RTE.

7 Clause 4 -- Scaling and Database Population Related Items

7.1 Table Cardinality

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. (8.1.5.1)

The database was originally built with 8704 warehouses. No rows were deleted from the warehouse table prior to the run per Clause 4.2.2 of the TPC specification. 8,000 warehouses were active during performance and repeatability runs. 44 warehouses in each of the 16 partitions were inactive (total 704 inactive).

Table 3: Table Cardinality

Table	Cardinality as Benchmarked
Warehouse	8,704
District	87,040
Customer	261,120,000
History	261,120,000
Orders	261,120,000
New Order	78,336,000
Order Line	2,610,963,835
Stock	870,400,000
Item	100,000
Deleted Warehouses	0

7.2 Constant Values

The following values were used as constant value inputs to the NURand function for this benchmark.

Table 4: Constant Values

Function	Constant C Value
C_LAST (Build)	1
C_LAST (Run)	87

7.3 Data Distribution

The distribution of tables and logs across all media must be explicitly depicted for the tested and priced systems. (8.1.5.2)

***Comment**: Detailed diagrams for layout of database files on disks can widely vary, and it is difficult to provide exact guideline suitable for all implementations. The intent is to provide sufficient detail to allow independent reconstruction of the test database. The two figures below are examples of database layout descriptions and are not intended to depict or imply any optimal layout for the TPC-C database.*

8.1.5.3 A statement must be provided that describes:

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

Oracle 8.0.4 is a relational DBMS. The interface used was Oracle OCI stored procedures accessed with Remote Procedure Calls.

Refer to Appendix B for Data Distribution.

7.4 Partition Mapping

The mapping of database partitions/replications must be explicitly described.
***Comment**: The intent is to provide sufficient detail about partitioning and replication to allow independent reconstruction of the test database. (8.1.5.4)*
An description of a database partitioning scheme is presented below as an example. The nomenclature of this example was outlined using the CUSTOMER table (in Clause 8.1.2.1), and has been extended to use the ORDER and ORDER_LINE tables as well.

Horizontal partitioning was used on all tables and indexes except for the STOCK and CUSTOMER tables. The functionality for this was provided by Oracle8. For further details of the partitioning of the database, see Appendix B.

7.5 180 Day Space Calculation

Details of the 180 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). (8.1.5.5)

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

1. The free space on the redo log was queried from the Oracle catalog.
2. Transactions were run against the database with a full load of users.
3. The increase in size to the redo logs was divided by the number of transactions, giving bytes used per new order.
4. This amount was multipied by the reported tpm rate times 480 minutes, giving total space needed for 8 hours.

For the dynamic tables, the following steps were followed:

1. The database was queried for the size of the dynamic tables.
2. The sum of D_NEXT_O_ID was queried from the DISTRICT table.
3. A full performance run was executed.
4. Steps 1 and 2 were repeated.
5. The change in size of the dynamic tables was divided by the number of new orders in the run giving growth per new order.
6. The number in previous step was multiplied by the reported tpm rate times 480 minutes.
7. The numbers in steps 1 and 5 were added giving space need for 8 hours.
8. The space allocated was verified to be larger than the space needed.

180 Day Space Calculations

Warehouses: 8704	**tpmC: 93,900.85**

Segment	**Rows**	**Data 2K pages**	**Index 2K pages**	**Extra 5%**	**Total with 5%**
Warehouse	8,704	8,704	1,600	515	10,819
District	87,040	87,040	16,000	5,152	108,192
Item	100,000	6,667	822	374	7,863
Customer	261,120,000	131,040,080	10,045,363	7,054,272	148,139,715
New_order	78,336,000	698,651	976,193	83,742	1,758,586
Stock	870,400,000	174,720,004	9,342,881	9,203,144	193,266,029
History (D)	261,120,000	7,410,446			7,410,446
Orders (D)	261,120,000	5,324,021	8,068,797		13,392,818
Order_line (D)	2,610,963,835	97,921,999	35,581,698		133,503,697
Roll_back + System		3,143,680			3,143,680
Totals (in 2K pages)		420,361,292	64,033,354	16,347,200	500,741,846

Dynamic space (MB)	216,125.91
Static space (MB)	766,148.39
Free space (MB)	653,753.13
Daily growth (MB)	37,742.57
Daily spread (MB)	0
180 day space (MB)	7,559,811.04
Log block size	512
Log blocks/tpmC	30.83
8hr log w/o mirror (GB)	662.60
180 day space (GB)	**7,382.63**
Logical logs w/ mirror (GB)	**1,325.21**
Total	**8,707.84**

	Capacity (GB)	**# Required**	**# Configured**
Database	8.49	870	1,200
Logs	17.09	78	96
O/S & Oracle			1
Total drives			1,297

8 Clause 5 -- Performance Metrics and Response Time Related Items

8.1 Measured TpmC

Measured tpmC must be reported. (8.1.6.1)

Measured TpmC	93,900.85
Price per TpmC	$131.67

8.2 Response Times

Ninetieth percentile, maximum and average response times must be reported for all transaction types as well as for the Menu response time. (8.1.6.2)

Table 5: Transaction Response Times

Transaction	Average	90%	Maximum
New Order	0.90	1.600	133.32
Payment	0.71	1.300	89.23
Order Status	0.76	1.400	6.15
Interactive Delivery	0.70	1.100	6.11
Deferred Delivery	1.01	1.493	228.20
Stock Level	1.15	2.100	6.94
Menu	0.20	0.440	10.29

8.3 Think Times & Key Times

The minimum, the average, and the maximum keying and think times must be reported for each transaction type. (8.1.6.3)

Table 6: Transaction Key Times

Transaction	Minimum	Average	Maximum
New Order	18.88	18.89	19.22
Payment	3.88	3.89	4.17
Order Status	2.88	2.89	3.13
Delivery	2.88	2.89	3.08
Stock Level	2.88	2.89	3.09

Table 7: Transaction Think Times

Transaction	Minimum	Average	Maximum
New Order	0.00	12.02	120.39
Payment	0.00	12.02	120.44
Order Status	0.00	10.07	100.86
Delivery	0.00	5.03	50.37
Stock Level	0.00	5.03	50.36

8.4 Response Time Distribution Curves

Response Time frequency distribution curves (see Clause 5.6.1) must be reported for each transaction type. (8.1.6.4)

Figure 1: New Order Response Time Distribution

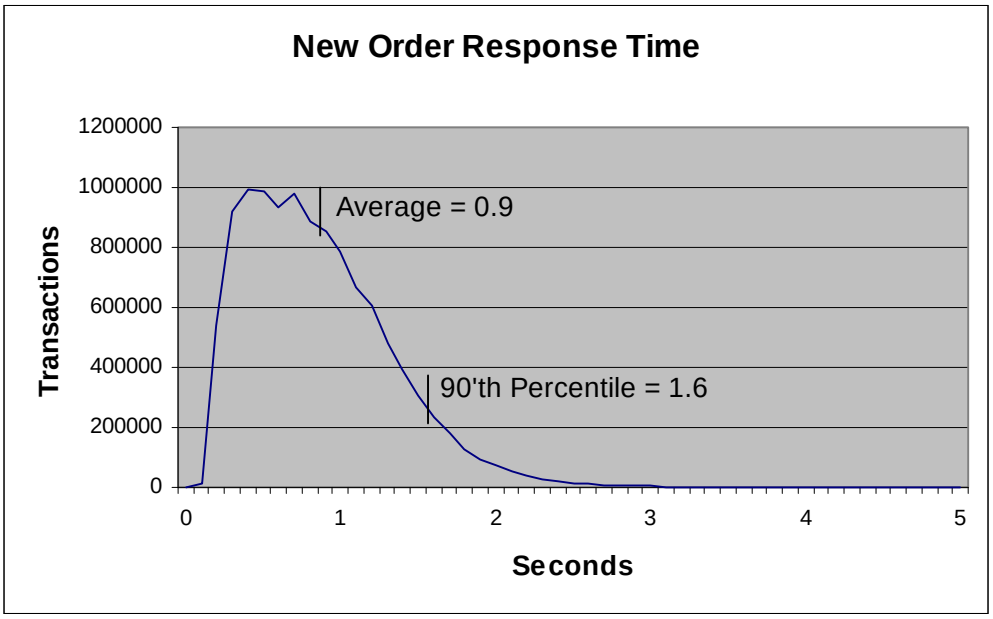


Figure 2: Payment Response Time Distribution

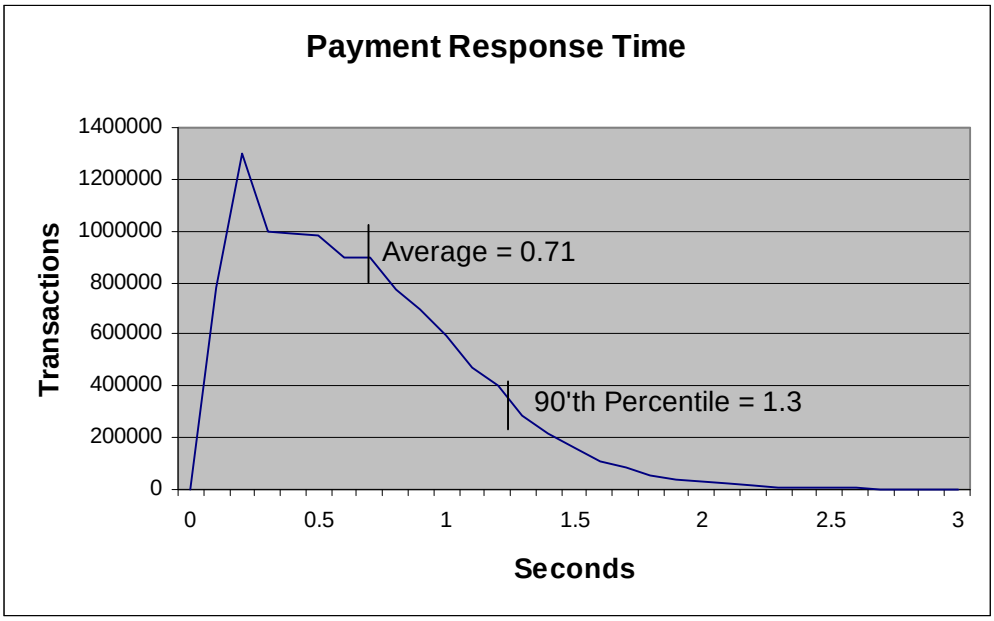


Figure 3: Order Status Response Time Distribution

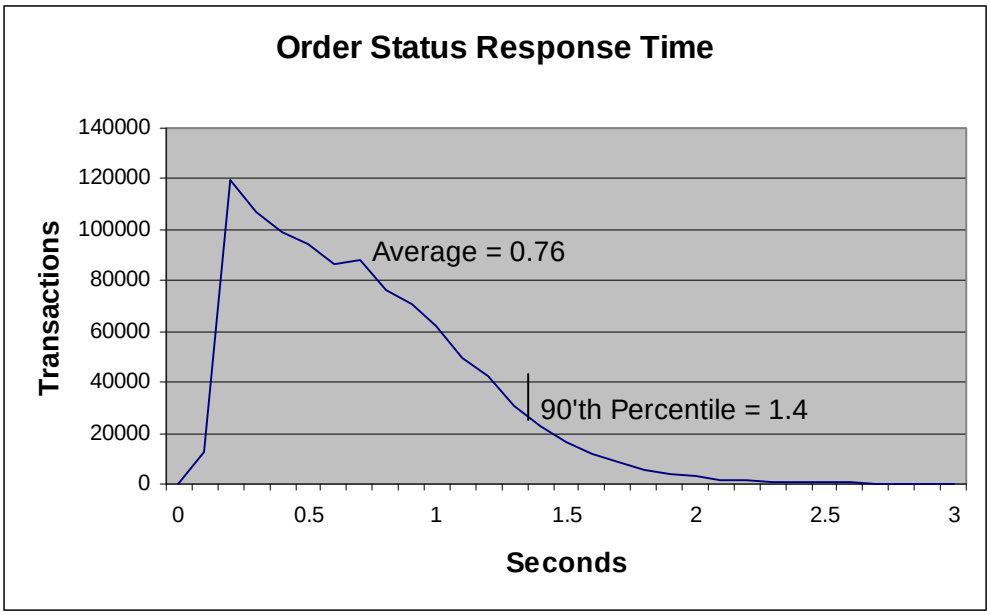


Figure 4: Delivery Response Time Distribution

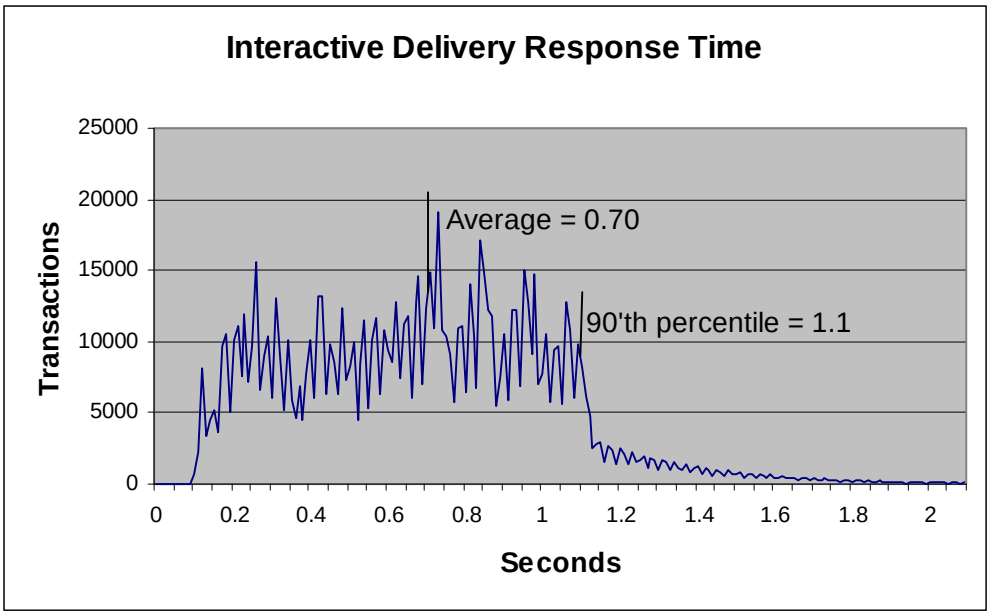
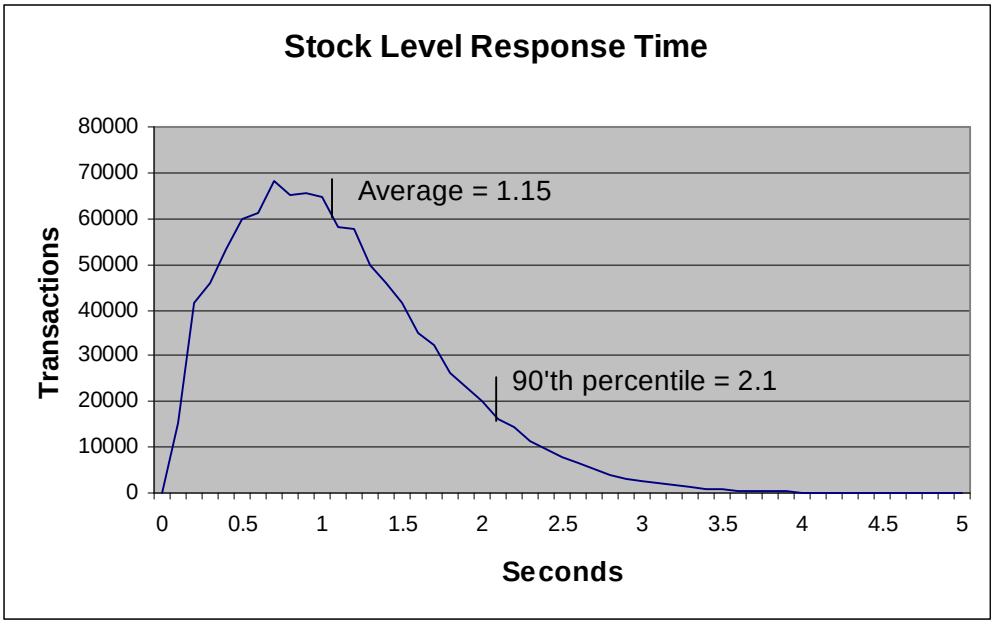


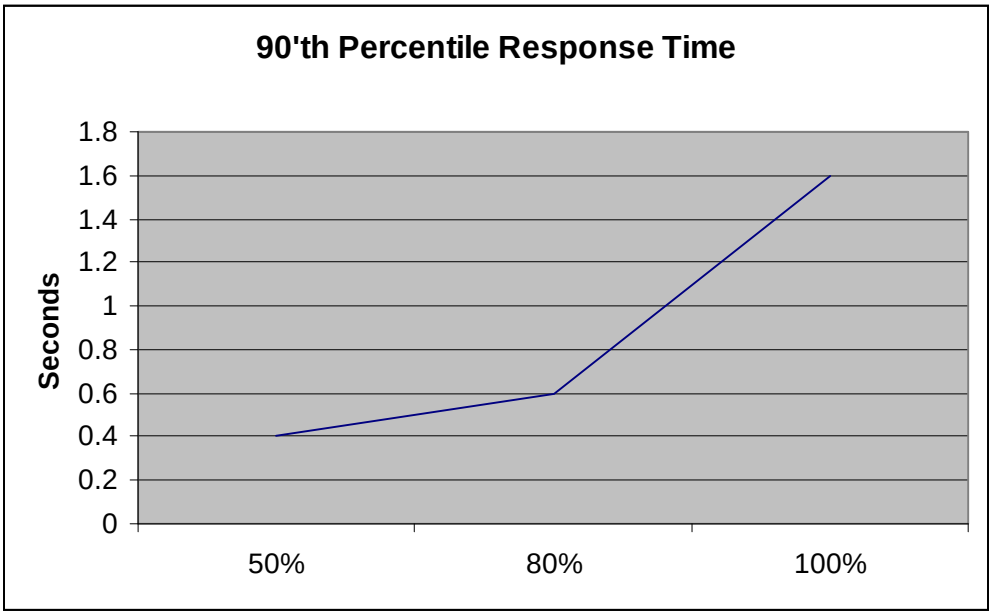
Figure 7: Stock Level Response Time Distribution



New-Order Response Time vs. Throughput Graph

The performance curve for response times versus throughput (see Clause 5.6.2) must be reported for the New-Order transaction. (8.1.6.5)

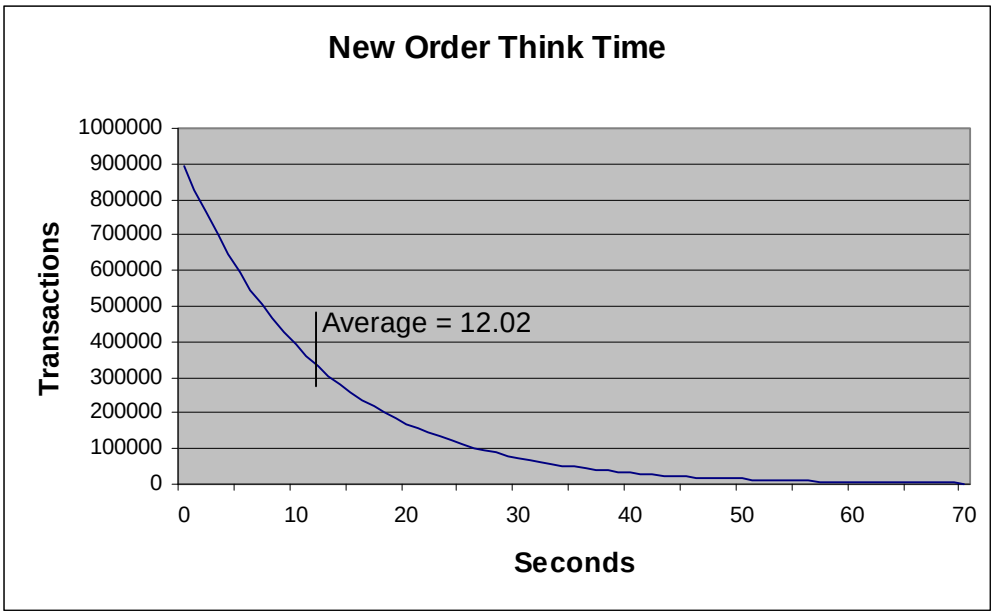
Figure 8: New Order Response Time vs. Throughput



8.5 New-Order Think Time Distribution Graph

Think Time frequency distribution curves (see Clause 5.6.3) must be reported for the New-Order transaction (8.1.6.6)

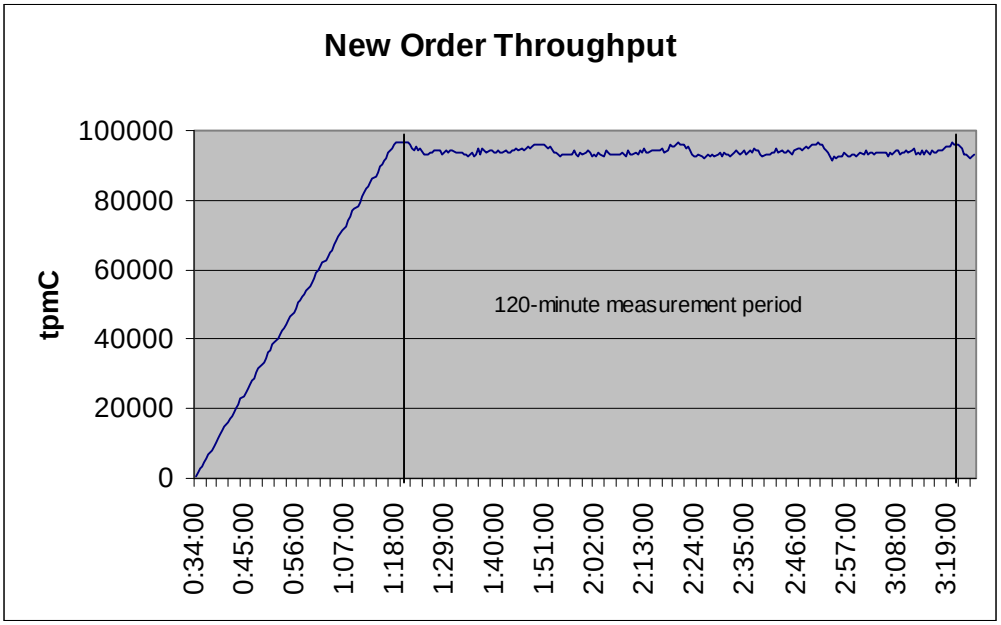
Figure 9: New Order Think Time Distribution



8.6 Steady-State Graph

A graph of throughput versus elapsed time (see Clause 5.6.5) must be reported for the New-Order transaction. (8.1.6.8)

Figure 10: New Order Throughput vs. Time



8.7 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. (8.1.6.9)

Steady state was determined using real time monitor utilities from both the operating system and the RTE. Steady state was further confirmed by the throughput data collected during the run and graphed in Figure 10.

8.8 Work Performed During Steady State

A description of how the work normally performed during a sustained test (for example checkpointing, writing redo/undo log records, etc.), actually occurred during the measurement interval must be reported. (8.1.6.10)

The RTE generated the required input data to choose a transaction from the menu. This data was timestamped. The menu response for the requested transaction was verified and timestamped in the RTE log files.

The RTE generated the required input data for the chosen transaction. It waited to complete the minimum required key time before transmitting the HTTP request to the client. The transmission was timestamped. The return of the screen with the required response data was timestamped. The difference between these two timestamps was the response time for that transaction and was logged in the RTE log.

The RTE then waited the required think time interval before repeating the process starting at selecting another transaction from the menu.

The RTE transmissions were sent to the web-based application program running on the client machines through Ethernet LANs. These web clients managed the emulated web browser interface as well as all requests to the database on the server. The applications communicated with the database server over another Ethernet LAN using the Tuxedo transaction monitor and Oracle8 OCI calls.

To perform checkpoints at specific intervals, we set Oracle8 *recovery interval* to an allowable value and wrote a script to schedule multiple checkpoints at specific intervals. Oracle8 logged the checkpoint beginning and ending time in the alert file. The script included a wait time of 30 minutes between checkpoints. The checkpoint script was started manually 18 minutes after the RTE had all users logged in and sending transactions.

At each checkpoint, Oracle8 wrote to disk all memory pages that had been updated but not yet physically written to disk. Four checkpoints were performed during measured period, protected zone requirements did not apply.

8.9 Reproducibility Methodology

A description of the method used to determine the reproducibility of the measurement results must be reported. (8.1.6.11)

The measurement procedure was repeated and the throughput was verified to be within 1% of the reported result.

8.10 Measurement Interval

A statement of the duration of the measurement interval for the reported Maximum Qualified Throughput (tpmC) must be included. (8.1.6.12)

The measurement interval was 120 minutes.

8.11 Transaction Mix

8.1.6.13 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.

The percentage of the total mix for each transaction type must be disclosed. (8.1.6.14)

Table 8: Transaction Mix

Transaction	Percentage
New Order	44.78%
Payment	43.07%
Order Status	4.04%
Delivery	4.04%
Stock Level	4.05%

8.12 Other Metrics

The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed. (8.1.6.15)
The average number of order-lines entered per New-Order transaction must be disclosed. (8.1.6.16)
The percentage of remote order-lines entered per New-Order transaction must be disclosed. (8.1.6.17)
The percentage of remote Payment transactions must be disclosed. (8.1.6.18)
The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed. (8.1.6.19)
The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed. (8.1.6.20)

Table 9: Transaction Statistics

Transaction	Function	Value
New Order	Home Warehouse Items	99.00%
	Remote Warehouse Items	1.00%
	Rolled Back Transactions	0.99%
	Average Lines Per Order	10.00
Payment	Home Warehouse	85.01%
	Remote Warehouse	14.99%
	Non-Primary Key Access	60.03%
Order Status	Non-Primary Key Access	60.02%
Delivery	Skipped Transactions	0

8.13 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. (8.1.6.21)

There were 4 checkpoints in the measurement interval. The first checkpoint started 26 seconds after the start of the measurement interval. The checkpoint interval was 30 minutes.

9 Clause 6 -- SUT, Driver, and Communication Definition Related Items

9.1 RTE Parameters

The RTE input parameters, code fragments, functions, etc. used to generate each transaction input field must be disclosed. (8.1.7.1)

Comment: *The intent is to demonstrate the RTE was configured to generate transaction input data as specified in Clause 2.*

The RTE input parameters and code fragments are listed in Appendix C - Tunable Parameters.

9.2 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. (8.1.7.2)

No components were emulated other than the emulated user workstations.

9.3 Benchmarked and Targeted System Configuration Diagrams

A complete functional diagram of both the benchmark 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). (8.1.7.3)

The driver system performed transaction data generation and communication to the client through the standard web browser (HTTP) protocol. It also captured and timestamped the SUT output data for post-processing of the reported metrics. No other functionality was included on the driver system.

Figures 1 & 2 of this report contain detailed diagrams of both the benchmark configuration and the priced configuration.

9.4 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). (8.1.7.4)

Sixteen 100Mbps ethernet connections were used between the clients and the database server. Ninety-six 10Mbps ethernet connections were used to connect emulated used to client machines.

9.5 Network Bandwidth

The bandwidth of the network(s) used in the tested/priced configuration must be disclosed. (8.1.7.5)

The bandwidth of the tested and priced networks were as follows:

- 10 BaseT (10 Mbit/sec) network segments between the RTE/Emulated Users and the Clients.
- 100 BaseT (100 Mbit/sec) between the Clients and Server.

9.6 Operator Intervention

If the configuration requires operator intervention (see Clause 6.6.6), the mechanism and the frequency of this intervention must be disclosed. (8.1.7.6)

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

10 Clause 7 -- Pricing Related Items

10.1 Hardware and Software List

A detailed list of hardware and software used in the priced system must be reported. Each separately orderable item must have vendor part number, description, and release/revision level, and either general availability status or committed delivery 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) of price(s) must also be reported. (8.1.8.1)

The total 5-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. (8.1.8.2)

A detailed price list is included in the abstract at the beginning of this report.

10.2 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 date for the priced system must be the date at which all components are committed to be available. (8.1.8.3)

Hardware Availability Date:	June 15th 1999
Software Availability Date:	Currently available

10.3 Measured TpmC

A statement of the measured tpmC, as well as the respective calculations for 5-year pricing, price/performance (price/tpmC), and the availability date must be included. (8.1.8.4)

Maximum Qualified Throughput:	93,900.85 TpmC
Price Performance Metric:	$131.67 per TpmC

10.4 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. (8.1.8.5)

This system is priced for the United States of America.

10.5 Usage Pricing

For any usage pricing, the sponsor must disclose (8.1.8.6):

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

***Comment**: Usage pricing may include, but is not limited to, the operating system and database management software.*

The component pricing based on usage is shown below:

- 8 Microsoft Windows NT Server 4.0 Licenses
- 1 DYNIX/ptx 4.4.4
- 1 Oracle8 8.0.4
- 1 Microsoft Visual C++ 32 bit Edition
- 5 Year Support for Hardware Components.

10.6 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). Clause 6.1 describes the Server and Client components. An example of the standard pricing sheet is shown in Appendix B. (8.1.8.7)
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. See example in Appendix B. (8.1.8.8)

Comment: *By standardizing the pricing spreadsheet and adding subtotals the value of the FDR and executive summary will be enhanced. This will allow the reader to more easily compare results and determine pricing.*

A detailed price list is included in the abstract at the beginning of this report. All third party quotations are included at the end of this report as Appendix D.

11 Clause 9 -- Audit Related Items

11.1 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. (8.1.9.1)
A review of the pricing model is required to ensure that all components required are priced (see Clause 9.2.8). The auditor is not required to review the final Full Disclosure Report or the final pricing prior to issuing the attestations letter. (8.1.9.2)

This TPC-C benchmark has been audited by:

Tom Sawyer
2229 Benita Dr.
Suite 101, Rancho Cordova
CA 95670
Phone: (916) 635-2822

11.2 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 Benchmark™ C", the Full Disclosure Report must have been submitted to the TPC Administrator as well as written permission obtained to distribute same.

Requests for this TPC Benchmark C Full Disclosure Report should be sent to:

Sequent Computer Systems, Inc.
15450 SW Koll Parkway
Beaverton, OR 97006-6063
Attention: Martin O'Sullivan

11.3 Auditor's Letter of Attestation

The Auditor's Letter of Attestation is on the next page.

PERFORMANCE METRICS INC.
TPC Certified Auditors

December 17, 1998

Horace Tong
Senior Staff Engineer
Solutions Integration Group
Sequent Computer Systems, Inc.
15450 SW Koll Parkway
Beaverton, OR 97006-6063

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

Platform: Sequent NUMACenter 2000 NE300
Database Server: NUMA-Q 2000 E300
Database Manager: Oracle8 Enterprise Edition Version 8.0.4
Operating System: DYNIX/ptx Version 4.4.4
Application Server: N300
Operating System: Windows NT 4.0
Transaction Manager: BEA Tuxedo CFS Version 6.3

Server: E300				
CPU's	Memory	Disks	90% Response	tpmC
64 Pentium II Xeon @ 405MHz	Main: 64 GB Cache: 2MB each	1,201 @ 9.1GB 96 @ 18GB	1.60 sec	93,900.85
8 Clients: N300 (each)				
4 Pentium II Xeon @ 400MHz	Main: 1 GB Cache: 512K	1 @ 9GB	Na	na

2229 Benita Dr. Suite 101, Rancho Cordova, CA 95670
(916) 635-2822 fax: (916) 858-0109 email: Lorna@PerfMetrics.com

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

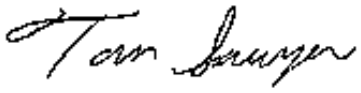
- The transactions were correctly implemented.
- The database files were properly sized and populated.
- The database was properly scaled with 8704 warehouses; 704 were not used during the measurement – see Auditor Notes. The W_YTD and D_NEXT_O_ID values for the unused warehouses were verified to be the same before and after the measurement.
- The ACID tests were performed on a previously reported system that used the same hardware and binaries. The Durability tests for loss-of-system and loss-of-memory were performed on the measured configuration
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was present on the tested system.
- Eight hours of growth space for the dynamic tables was present on the tested system.
- The data for the 180 day space calculation was verified
- The steady state portion of the test was 120 minutes.
- Four checkpoints were taken during the measured interval.
- One checkpoint was taken before the measured interval.
- The system pricing was checked for major components and maintenance.

Auditor Notes:

The implementation used 16 instances of Oracle. The warehouse range was divided into 16 parts and 44 warehouses at the end of each part were skipped. The RTE was configured to distribute transactions evenly over the remaining portions of the warehouse range. This was verified by examining the D_NEXT_O_ID values over the measured range. A separate query was used to validate that the skipped warehouses were not used as remote warehouses in order-lines.

Sincerely,

Tom Sawyer



Auditor

2229 Benita Dr. Suite 101, Rancho Cordova, CA 95670
(916) 635-2822 fax: (916) 858-0109 email: Lorna@PerfMetrics.com

Appendix A – Application Source Code

Tpcc.dll Source Code

Tpcc.def

```
LIBRARY         tpcc
˜

˜
DESCRIPTION  'Dell TPC-C Client / Web Server'
˜

˜
EXPORTS
˜
   GetExtensionVersion
˜
   HttpExtensionProc
```

Tpcc.c

```
//--------------------tpcc.c: Dell TPC-C Client / Web Server----------------------------
//
// Copyright (c) 1997 Dell Computer Corporation, All Rights Reserved
//
// Author: Dave Jaffe/James Jordan                                          Last
modified: 9/24/97
//
// Audited: Richard Gimarc  Performance Metrics Inc.  9/24/97
//
// source code for tpcc.dll MS ISAPI DLL for TPC-C Benchmark
//
#include "tpcc_dell.h"
#include "tmclient.c"
//
//
//---------------------------------DllMain-----------------------------------------------
// called when DLL loads
//


/*
 * Copyright (c) 1984, 1985, 1986, 1987, 1988, 1989, 1990,
 * 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998
 * Sequent Computer Systems, Inc.   All rights reserved.
 *
 * This software is furnished under a license and may be used
 * only in accordance with the terms of that license and with the
 * inclusion of the above copyright notice.   This software may not
 * be provided or otherwise made available to, or used by, any
 * other person.  No title to or ownership of the software is
 * hereby transferred.
 */

static char *RCSID = "@(#)$Id: tpcc.c,v 1.6 1998/09/29 07:05:51 administrator
Exp administrator $";

static char env_tuxconfig[128];

BOOL WINAPI DllMain(HINSTANCE hInst, ULONG ulReason, LPVOID
lpReserved)
        {

        char msg[128];
        char tuxconfig[128];
        char tux_env_filename[128];
        FILE *fp;

        switch(ulReason)
                {
                case DLL_PROCESS_ATTACH:
                        {
                        if(!OpenErrorLog()) return FALSE;
                        WriteErrorLog("Opened error log");
                        ReadTPCCRegParams();
                        InitializeCriticalSection(&login_crit_sec);

                        InitializeCriticalSection(&TLS_crit_sec);

                                // Obtain frequency and initial count
of 64-bit counter
                                QueryPerformanceFrequency(&freq);
                                freqd = (double) freq.QuadPart;

                        QueryPerformanceCounter(&tick_count0);

                                sprintf(tux_env_filename,
"%s\\tux_env", getenv("TUXDIR"));
                                if ((fp = fopen(tux_env_filename,
"r")) == NULL)
                                        {
                                        sprintf(msg,
"ERROR: open %s failed: %s", tux_env_filename,
                        strerror(errno));
                                        WriteErrorLog(msg);
                                        return FALSE;
                                        }
                                else
                                        {
                                        fscanf(fp, "%s",
tuxconfig);
                                        fclose(fp);
                                        sprintf(env_tuxconfig,
"TUXCONFIG=%s", tuxconfig);
                                        sprintf(msg, "Thread
%d: env_tuxconfig = (%s)",
                        GetCurrentThreadId(), env_tuxconfig);
                                        WriteErrorLog(msg);
                                         }

                                if (_putenv(env_tuxconfig))
                                        {
                                        sprintf(msg,
"ERROR: Thread %d: _putenv(%s) failed",
                        GetCurrentThreadId(), env_tuxconfig);
                                        WriteErrorLog(msg);
                                        return FALSE;
                                        }
                                else    {
                                        sprintf(msg, "Thread
%d: _putenv(%s) OK",
                        GetCurrentThreadId(), env_tuxconfig);
                                        WriteErrorLog(msg);
                                        }

                                if (!TMClientInit())
                                        return FALSE;
                                break;

                                }
                        case DLL_PROCESS_DETACH:
                                {
                                TMClientExit();
                                sprintf(msg, "Thread %d: exiting",
GetCurrentThreadId());
                                WriteErrorLog(msg);
                                fclose(fp_errorlog);
                                break;
                                }
                        default:
                                break;
                        }
        return TRUE;
        } // DllMain

//
//
//---------------------------------GetExtensionVersion---------------------------------
// called by IIS when DLL loads
//
BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO *pVer)
        {
        pVer->dwExtensionVersion =
MAKELONG(HSE_VERSION_MINOR, HSE_VERSION_MAJOR);
        lstrcpyn(pVer->lpszExtensionDesc, "Sequent TPC-C Client / Web
Server",
                        HSE_MAX_EXT_DLL_NAME_LEN);
```

```
		return TRUE;
		}  // GetExtensionVersion
//
//
//--------------------------------HttpExtensionProc-----------------------------------
// called by IIS for each request
//
DWORD WINAPI HttpExtensionProc(EXTENSION_CONTROL_BLOCK
*pECB)
		{
		enum _command Command;
		int User;
		char* cmd_input_str;
		char html[4096];
		char response[4096];
		BOOL KeepConn = TRUE;
		DWORD len;

		char var_string[128];
		int strsize;
		char msg[128];

		// Get the input string  (assumes GET)
		if (ParseInput(pECB->lpszQueryString, &Command, &User,
&cmd_input_str))
				{
				switch(Command)
						{
						case NewOrderFormRequest:
								sprintf(html,
NewOrderForm, dll_path, User, pUserData[User]->w_id);
								break;
						case NewOrder:

		ProcessNewOrder(cmd_input_str, User, html);
								break;
						case PaymentFormRequest:
								sprintf(html,
PaymentForm, dll_path, User, pUserData[User]->w_id);
								break;
						case Payment:

		ProcessPayment(cmd_input_str, User, html);
								break;
						case OrderStatusFormRequest:
								sprintf(html,
OrderStatusForm, dll_path, User, pUserData[User]->w_id);
								break;
						case OrderStatus:

		ProcessOrderStatus(cmd_input_str, User, html);
								break;
						case DeliveryFormRequest:
								sprintf(html,
DeliveryForm, dll_path, User, pUserData[User]->w_id);
								break;
						case Delivery:

		ProcessDelivery(cmd_input_str, User, html);
								break;
						case StockLevelFormRequest:
								sprintf(html,
StockLevelForm, dll_path, User, pUserData[User]->w_id,

		pUserData[User]->d_id);
								break;
						case StockLevel:

		ProcessStockLevel(cmd_input_str, User, html);
								break;
						case Login:

		ProcessLogin(cmd_input_str, html, &KeepConn);
								break;
						case Logout:
								ProcessLogout(User);
								sprintf(html,
LoginForm, dll_path);
								KeepConn = FALSE;
								break;
						default:
								strcpy(html,
"<HTML>ERROR: Command not recognized</HTML>");
								KeepConn = TRUE;
								break;
						}
				}
		else
				{
				strcpy(html, "<HTML> ERROR: Can't parse
input</HTML>");
				}

skip_cmd:
		sprintf(response,
				"Content-Type: text/html\r\n"
				"Content-Length: %d\r\n"
				"%s\r\n",
				strlen(html),
				(KeepConn ? "Connection: Keep-Alive\r\n" : ""));

		strcat(response, html);

		len = strlen(response);

   pECB->ServerSupportFunction( pECB->ConnID,
HSE_REQ_SEND_RESPONSE_HEADER,
				"200 OK", &len, (LPDWORD) response);

   pECB->dwHttpStatusCode=200; // 200 OK

   return (KeepConn ? HSE_STATUS_SUCCESS_AND_KEEP_CONN :
HSE_STATUS_SUCCESS);

		} // HttpExtensionProc
//
//
//--------------------------------ParseInput-----------------------------------------
// parse input string (URL) from browser
// determine command, user, and command input string (remainder of string)
// returns TRUE if successful, FALSE if parsing error
// forms of input string:  CMD=NewOrderFormRequest&USER=1234
//			|		|
//			pCMD		pUSER
//
//			CMD=NewOrder&USER=1234&INPUT1=...
//			|	|	|
//			pCMD	pUSER	pINPUTS
//
BOOL ParseInput(char* pCMD, enum _command* pCommand, int* User, char**
pInputs)
		{
		int i, len;
		char* pUSER;
		char cmd_str[32];
		char user_str[6];
		char* pTemp;


		// Check input string starts with CMD=
		if (strncmp(pCMD, "CMD=", 4)) return FALSE;

		// Copy command string to cmd_str, look up in array of command
names
		pUSER = 1 + strchr(pCMD, '&');
		len = pUSER - pCMD - 5;
		strncpy(cmd_str, pCMD + 4, len);
		cmd_str[len] = '\0';
		for(i=0; i<=Logout; i++) if (!strcmp(command_name[i], cmd_str))
break;
		*pCommand = (enum _command) i;

		// Check next PARAM=value pair is USER
		if (strncmp(pUSER, "USER=", 5)) return FALSE;  // string does
not contain USER=

		// Copy user string to user_str, convert to integer
		if ((pTemp = strchr(pUSER, '&')) == NULL) //  USER is end of
input
				{
				strcpy(user_str, pUSER + 5);
				*pInputs = NULL;
				}
		else
				{
				len = pTemp - pUSER - 5;
				strncpy(user_str, pUSER + 5, len);
				user_str[len] = '\0';
				*pInputs = pTemp + 1;
				}

		*User = atoi(user_str);

		return TRUE;

		} // ParseInput
//
```

```
//
//---------------------------------ProcessLogin----------------------------------------
// process login request; assign user number; create user's command button bar
//
void ProcessLogin(char* cmd_input_str, char* html, BOOL* KeepConn)
                {
                char text[256];
                short w_id, d_id;
                int User = -1;
                int rc, i;
                BOOL success = FALSE;

                *KeepConn = FALSE;

                // If input cannot be parsed or warehouse or district is out of range
tell user
                // to re-submit
                if ((rc = ParseLogin(cmd_input_str, &w_id, &d_id)) != SUCCESS)
                                sprintf(text, "ERROR: %s", err_text[rc]);
                else if (!(0 < w_id && w_id <= n_warehouses_total))
                                strcpy(text, "ERROR: Warehouse out of range - use
Back button and resubmit");
                else if (!(0 < d_id && d_id <= 10))
                                strcpy(text, "ERROR: District out of range - use
Back button and resubmit");
                else // Everything checks out - assign the user a User number if
client not full
                                {
                                // Find first open slot in UserData array and mark it
taken
                                EnterCriticalSection(&login_crit_sec);
                                for (i=0; i < max_users_this_client; i++)
                                                {
                                                if (pUserData[i] == 0)
                                                                {
                                                                User = i;
                                                                pUserData[i] =
(UserData*) 1;
                                                                break;
                                                                }
                                                }
                                LeaveCriticalSection(&login_crit_sec);

                                if (User == -1) // No open slots - tell user to get lost
                                                strcpy(text,
                                                                "ERROR: Max
number of users has been reached - please retry later");

                                else                                                    //
Success! - initialize User's data
                                                {
                                                success = TRUE;
                                                // Assign user space for his data
                                                if(!(pUserData[User] = (UserData*)
malloc(sizeof(UserData))))
                                                                strcpy(text,

                "ERROR: no space to allocate for this user - see sys admin");
                                                else
                                                                {
                                                                // Initialize User's data
area with w_id, d_id and button bar
                                                                pUserData[User]-
>w_id = w_id;
                                                                pUserData[User]-
>d_id = d_id;

                sprintf(pUserData[User]->ButtonBar, ButtonBar,

                dll_path, User, dll_path, User, dll_path, User, dll_path, User,

                dll_path, User, dll_path, User);

                                                                sprintf(text,
"Congratulations! You have successfully logged into "
                                                                                "the
Sequent TPC-C Client!\n"
                                                                                "User:
%d  Thread: 0x%X  w_id: %d  d_id: %d\n",
                                                                                User,
GetCurrentThreadId(), w_id, d_id);

                                                                *KeepConn = TRUE;
                                                                }
                                                }
                                }

                sprintf(html, ResponseHTML, "TPC-C Main Menu",
                                success ? pUserData[User]->ButtonBar : "", text);
                } // ProcessLogin
//
//
//---------------------------------ParseLogin----------------------------------------
// parse login input string
// determine w_id, d_id
// returns SUCCESS if successful, error code if parsing error
// form of input string: W_ID=1234&D_ID=12
//                       |      |
//                       pW     pD
//
int ParseLogin(char* pW, short* w_id, short* d_id)
                {
                int len;
                char* pD;
                char wid_str[6];

                pD = 1 + strchr(pW, '&');

                // Check input string
                if (strncmp(pW, "W_ID=", 5)) return ERR_GENERIC;      // first
param not W_ID
                if ((len = pD - pW - 6) < 1) return ERR_BLANK_WID;      // blank
W_ID field
                if (strncmp(pD, "D_ID=", 5)) return ERR_GENERIC;      //
second param not D_ID
                if (strchr(pD, '&') != NULL) return ERR_GENERIC;      //
D_ID not last PARAM
                if (*(pD+5) == '\0') return ERR_BLANK_DID;
                                // blank D_ID field

                // Copy W_ID string to wid_str, convert to integer, checking for
non-numerics
                strncpy(wid_str, pW + 5, len);
                wid_str[len] = '\0';
                if (!IsValidNonNegShort(wid_str, w_id)) return
ERR_NONNUM_WID;

                //convert D_ID string to integer, checking for non-numerics
                if (!IsValidNonNegShort(pD+5, d_id)) return
ERR_NONNUM_DID;

                return SUCCESS;

                } // ParseLogin
//
//
//---------------------------------ProcessNewOrder----------------------------------------
// process NewOrder form
//
void ProcessNewOrder(char* cmd_input_str, int User, char* html)
                {
                int i,rc;
                char text1[2048];
                char text[2048];
                NEW_ORDER_DATA* pNO = (NEW_ORDER_DATA*)
pUserData[User]->TransData;

                memset(pUserData[User]->TransData,'\0',sizeof(pUserData[User]-
>TransData));
                pNO->w_id = pUserData[User]->w_id;

                if ((rc = ParseNewOrder(cmd_input_str, pNO)) != SUCCESS)
                                sprintf(text, "ERROR: %s", err_text[rc]);

                else
                                {
                                // call NewOrder database lookup
                                rc = TMNewOrder(pNO);

                                switch(rc)
                                                {
                                                case SUCCESS:
                                                                pNO->total_amount
*=                                                                              (1 +
pNO->w_tax + pNO->d_tax) * (1 - pNO->c_discount);

                                                                sprintf(text, "
New Order\n"

                "Warehouse: %04d"
                                                                                "
District: %02d                  Date: %s\n"

                "Customer:  %04d"
                                                                                "
Name: %-16s   Credit: %-2s   %%Disc: %05.2f\n"
```

```
                                                "Order
Number: %08d  Number of Lines: %02d       "
                "W_tax: %05.2f   D_tax: %05.2f\n\n"
                                                "
Supp_W  Item_Id  Item Name              Qty  Stock  B/G  "
                                                "Price
Amount\n",
                                                pNO-
>w_id, pNO->d_id, pNO->o_entry_d, pNO->c_id, pNO->c_last,
                                                pNO-
>c_credit, 100.0*pNO->c_discount, pNO->o_id, pNO->o_ol_cnt,
                100.0*pNO->w_tax, 100.0*pNO->d_tax);
                                                for(i=0; i < pNO-
>o_ol_cnt; i++)
                                                {
                sprintf(text1, " %04d   %06d   %-24s  %02d    %03d    "
                "%1c   $%06.2f  $%07.2f\n",
                pNO->Ol[i].ol_supply_w_id,
                pNO->Ol[i].ol_i_id,
                pNO->Ol[i].ol_i_name,
                pNO->Ol[i].ol_quantity,
                pNO->Ol[i].ol_stock,
                pNO->Ol[i].ol_brand_generic,
                pNO->Ol[i].ol_i_price,
                pNO->Ol[i].ol_amount
                );
                strcat(text, text1);
                                                }
                                                for (i=pNO-
>o_ol_cnt; i<15; i++) strcat(text, "\n");
                                                sprintf(text1,
"Execution Status: Transaction committed"
                                                "
Total:  $%08.2f\n", pNO->total_amount);
                                                strcat(text, text1);
                                                break;
                                        case DEADLOCK:
                                                strcat(text, "ERROR:
DEADLOCK");
                                                break;
                                        case INVALID_ITEM:
                                                sprintf(text, "
New Order\n"
                "Warehouse: %04d"
                                                "
District:  %02d                 Date:\n"
                "Customer:  %04d"
                                                "
Name: %-16s   Credit: %-2s   %%Disc:\n"
                                                "Order
Number: %08d  Number of Lines:          "
                "W_tax:       D_tax:\n\n"
                                                "
Supp_W  Item_Id  Item Name              Qty  Stock  B/G  "
                                                "Price
Amount\n",
                                                pNO-
>w_id, pNO->d_id, pNO->c_id, pNO->c_last, pNO->c_credit,
                                                pNO-
>o_id);
                                                strcat(text,
"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n"
                "Execution Status: Item number is not valid");
                                                break;
                                        case SQL_ERROR:
                                                strcpy(text, "ERROR:
SQL Error");
                                                break;
                                        } // End switch
                                } // End else

                sprintf(html, ResponseHTML, "New Order
Response",pUserData[User]->ButtonBar,text);

        } // ProcessNewOrder
//
//
//---------------------------------ParseNewOrder---------------------------------------
// parse NewOrder input string
// fill in New Order data structure
// returns SUCCESS if successful, error code if parsing error
// form of input string:
//    D_ID=12&C_ID=1234&W00=1234&I00=123456&Q00=12&W02....
Q14=xx
//   |     |       |       |       |
//   pD     pC       pW      pI        pQ
//
int ParseNewOrder(char* pD, NEW_ORDER_DATA* pNO)
        {
        int len, i, n_blanks;
        char str[7];
        char *pC, *pW, *pI, *pQ, *pTemp;
        BOOL last_line;

        pNO->o_all_local = TRUE;

        // Check input string starts with D_ID=
        if (strncmp(pD, "D_ID=", 5)) return ERR_GENERIC;

        // Copy D_ID string to str, convert to integer, checking for non-
numerics
        pC = 1 + strchr(pD, '&');
        if ((len = pC - pD - 6) < 1) return ERR_BLANK_DID;
                // blank D_ID field
        strncpy(str, pD + 5, len);
        str[len] = '\0';

        if (!IsValidNonNegShort(str, &pNO->d_id)) return
ERR_NONNUM_DID; // non-num D_ID

        // Copy C_ID string to str, convert to integer, checking for non-
numerics
        pW = 1 + strchr(pC, '&');
        if ((len = pW - pC - 6) < 1) return ERR_BLANK_CID;
                // blank C_ID field
        strncpy(str, pC + 5, len);
        str[len] = '\0';
        if (!IsValidNonNegLong(str, &pNO->c_id)) return
ERR_NONNUM_CID;                 // non-num C_ID

        // Parse triplets of Wnn, Inn, Qnn
        //   - item valid only if all 3 fields are valid
        //   - scanning stops after all blank line or end-of-string encountered
        //   - check all fields for non-numeric or negative input

        i=0;
        do
                {
                n_blanks = 0;
                last_line = FALSE;

                // Supplying w_id
                pI = 1 + strchr(pW, '&');
                if ((len = pI - pW - 5) < 1) ++n_blanks;
                // blank Wxx field
                else
                        {
                        strncpy(str, pW + 4, len);
                        str[len] = '\0';
                        if (!IsValidNonNegShort(str, &pNO-
>Ol[i].ol_supply_w_id))
                                return
ERR_NONNUM_OL_S_WID;
                        pNO->o_all_local &= (pNO->w_id
== pNO->Ol[i].ol_supply_w_id);
                        }

                // Item id
                pQ = 1 + strchr(pI, '&');
                if ((len = pQ - pI - 5) < 1) ++n_blanks;
                // blank Ixx field
                else
                        {
                        strncpy(str, pI + 4, len);
                        str[len] = '\0';
                        if (!IsValidNonNegLong(str, &pNO-
>Ol[i].ol_i_id))
```

```
                                        return
ERR_NONNUM_OL_I_ID;
                                }

                        // Quantity
                        if ((pTemp = strchr(pQ, '&')) == NULL) // End of
input
                                {
                                last_line = TRUE;
                                pW = strchr(pQ, '\0');
                                }
                        else
                                {
                                pW = pTemp + 1;
                                }
                        if ((len = pW - pQ - (5 - last_line)) < 1 )
++n_blanks;
                        else
                                {
                                strncpy(str, pQ + 4, len);
                                str[len] = '\0';
                                if (!IsValidNonNegShort(str, &pNO-
>Ol[i].ol_quantity))
                                        return
ERR_NONNUM_OL_QUAN;
                                }

                        if (n_blanks == 3) break;
                        if (n_blanks == 1 || n_blanks == 2) return
ERR_INCOMPLETE_OL;

                        ++i;

                        }  while (!last_line); // End do

                pNO->o_ol_cnt = i;


                return SUCCESS;

                } // ParseNewOrder
//
//
//--------------------------------ProcessPayment-------------------------------------
// process Payment form
//
void ProcessPayment(char* cmd_input_str, int User, char* html)
                {
                int rc;
                char text[2048];
                PAYMENT_DATA* pPmt = (PAYMENT_DATA*)
pUserData[User]->TransData;

                memset(pUserData[User]->TransData,'\0',sizeof(pUserData[User]-
>TransData));
                pPmt->w_id = pUserData[User]->w_id;

                if ((rc = ParsePayment(cmd_input_str, pPmt)) != SUCCESS)
                        sprintf(text, "ERROR: %s", err_text[rc]);

                else
                        {
                        // call Payment database lookup
#ifdef DEBUG
                        strcpy(text, "calling TMPayment()\n");
                        WriteErrorLog(text);
#endif /* DEBUG */
                        rc = TMPayment(pPmt);

                        switch(rc)
                                {
                                case SUCCESS:
#ifdef DEBUG
                                                strcpy(text, "DEBUG:
TMPayment() returned SUCCESS");
                                                WriteErrorLog(text);
#endif /* DEBUG */
                                                sprintf(text, "
Payment\n"
                                                                "Date:
%s\n\n"

                "Warehouse: %04d                "

                "District: %02d\n"
                                                                "%-20s
%-20s\n"
                                                                "%-20s
%-2s %5.5s-%4.4s       "
                                                                "%-20s
%-2s %5.5s-%4.4s\n\n"

                "Customer: %04d  Cust-Warehouse: %04d  Cust-District: %02d\n"

                "Name:   %-16s %-2s %-16s     Since:  %s\n"
                                                                "
%-20s                   Credit: %-2s\n"
                                                                "
%-20s                   %%Disc:  %05.2f\n"
                                                                "
%-20s %-2s %5.5s-%4.4s       "

                "Phone:  %-6.6s-%-3.3s-%-3.3s-%-4.4s\n\n"

                "Amount Paid:          $%07.2f      New Cust-Balance: $%014.2f\n"
                                                                "Credit
Limit:   $%013.2f\n\n"
                                                                "Cust-
Data: %-50.50s\n"
                                                                "
%-50.50s\n"
                                                                "
%-50.50s\n"
                                                                "
%-50.50s\n",
                                                                pPmt-
>h_date, pPmt->w_id, pPmt->d_id,
                                                                pPmt-
>w_street_1, pPmt->d_street_1,
                                                                pPmt-
>w_street_2, pPmt->d_street_2,
                                                                pPmt-
>w_city, pPmt->w_state, pPmt->w_zip, pPmt->w_zip+5,
                                                                pPmt-
>d_city, pPmt->d_state, pPmt->d_zip, pPmt->d_zip+5,
                                                                pPmt-
>c_id, pPmt->c_w_id, pPmt->c_d_id,
                                                                pPmt-
>c_first, pPmt->c_middle, pPmt->c_last, pPmt->c_since,
                                                                pPmt-
>c_street_1, pPmt->c_credit,
                                                                pPmt-
>c_street_2, 100.0*pPmt->c_discount,
                                                                pPmt-
>c_city, pPmt->c_state, pPmt->c_zip, pPmt->c_zip+5,
                                                                pPmt-
>c_phone, pPmt->c_phone+6 ,pPmt->c_phone+9, pPmt->c_phone+12,
                                                                pPmt-
>h_amount, pPmt->c_balance, pPmt->c_credit_lim,
                                                                pPmt-
>c_data, pPmt->c_data+50, pPmt->c_data+100,pPmt->c_data+150);
#ifdef DEBUG

                WriteErrorLog(text);
#endif /* DEBUG */
                                                        break;
                                                case DEADLOCK:
                                                        strcpy(text, "ERROR:
DEADLOCK");
                                                        break;
                                                case SQL_ERROR:
                                                        strcpy(text, "ERROR:
SQL Error");
                                                        break;
                                                default:
                                                        sprintf(text, "ERROR:
TMPayment() returned %d\n", rc);
                                                        WriteErrorLog(text);
                                                        break;
                                        } // End switch
                                } // End else

                sprintf(html, ResponseHTML, "Payment
Response",pUserData[User]->ButtonBar,text);

                } // ProcessPayment
//
//
//--------------------------------ParsePayment-------------------------------------
// parse Payment input string
// fill in Payment data structure
// returns SUCCESS if successful, error code if parsing error
// form of input string:
//
D_ID=12&C_ID=1234&C_W_ID=1234&C_D_ID=12&C_LAST=123..16&AM
OUNT=1234.56
```

```
//  |     |       |       |       |         |
//  pD    pC      pCL     pCD     pCL       pA
//
//  (either C_ID or C_LAST must be blank)
//
int ParsePayment(char* pD, PAYMENT_DATA* pPmt)
        {
        int len;
        char str[7];
        char *pC, *pCW, *pCD, *pCL, *pA;
        BOOL c_id_blank = FALSE, c_last_blank = FALSE;

        // Check input string starts with D_ID=
        if (strncmp(pD, "D_ID=", 5)) return ERR_GENERIC;

        // Copy D_ID string to str, convert to integer, checking for non-
numerics
        pC = 1 + strchr(pD, '&');
        if ((len = pC - pD - 6) < 1) return ERR_BLANK_DID;
                // blank D_ID field
        strncpy(str, pD + 5, len);
        str[len] = '\0';
        if (!IsValidNonNegShort(str, &pPmt->d_id)) return
ERR_NONNUM_DID;// non-num D_ID

        // Copy C_ID string to str, convert to integer, checking for non-
numerics
        pPmt->c_id = 0;
        pCW = 1 + strchr(pC, '&');
        if ((len = pCW - pC - 6) < 1) c_id_blank = TRUE;
        // blank C_ID field
        else
                {
                strncpy(str, pC + 5, len);
                str[len] = '\0';
                if (!IsValidNonNegLong(str, &pPmt->c_id)) return
ERR_NONNUM_CID;
                }

        // Copy C_W_ID string to str, convert to integer, checking for non-
numerics
        pCD = 1 + strchr(pCW, '&');
        if ((len = pCD - pCW - 8) < 1) return ERR_BLANK_CWID;
                // blank C_W_ID field
        strncpy(str, pCW + 7, len);
        str[len] = '\0';
        if(!IsValidNonNegShort(str, &pPmt->c_w_id))return
ERR_NONNUM_CWID;

        // Copy C_D_ID string to str, convert to integer, checking for non-
numerics
        pCL = 1 + strchr(pCD, '&');
        if ((len = pCL - pCD - 8) < 1) return ERR_BLANK_CDID;
        // blank C_D_ID field
        strncpy(str, pCD + 7, len);
        str[len] = '\0';
        if(!IsValidNonNegShort(str, &pPmt->c_d_id))return
ERR_NONNUM_CDID;

        // Copy C_LAST string to c_last
        pA = 1 + strchr(pCL, '&');
        if ((len = pA - pCL - 8) < 1)  c_last_blank = TRUE;          // blank
C_LAST field
        // both C_ID and C_LAST are blank or both are filled in (error
either way)
        if (c_id_blank == c_last_blank) return ERR_CUST_NAME_ID;
        strncpy(pPmt->c_last, pCL + 7, len);
        pPmt->c_last[len] = '\0';

        // Copy AMOUNT string to str, convert to double, checking for
non-numerics
        strcpy(str, pA + 7);
        if (!strlen(str)) return ERR_BLANK_AMOUNT;
                                        // blank AMOUNT field
        if (!IsValidNonNegDouble(str, &pPmt->h_amount)) return
ERR_NONNUM_AMOUNT;
        if (pPmt->h_amount > 9999.99) return ERR_AMT_TOO_LARGE;
        // amount is too large

        return SUCCESS;

        } // ParsePayment
//
//
//--------------------------------ProcessOrderStatus---------------------------------
// process OrderStatus form
//
void ProcessOrderStatus(char* cmd_input_str, int User, char* html)
        {
        int rc, i;
        char text[2048], ol_text[100];
        ORDER_STATUS_DATA* pOS = (ORDER_STATUS_DATA*)
pUserData[User]->TransData;

        memset(pUserData[User]->TransData,'\0',sizeof(pUserData[User]-
>TransData));
        pOS->w_id = pUserData[User]->w_id;
        if ((rc = ParseOrderStatus(cmd_input_str, pOS)) != SUCCESS)
                sprintf(text, "ERROR: %s", err_text[rc]);

        else
                {
                // call OrderStatus database lookup
                rc = TMOrderStatus(pOS);

                switch(rc)
                        {
                        case SUCCESS:
                                sprintf(text, "
Order-Status\n"

        "Warehouse: %04d   District: %02d\n"

        "Customer: %04d   Name: %-16s %-2s %-16s\n"
                                                "Cust-
Balance: $%09.2f\n\n"

        "Order-Number: %08d   Entry-Date: %-19s   Carrier-Number:
%02d\n"

        "Supply-W     Item-Id    Qty     Amount      Delivery-Date\n",
                                                pOS-
>w_id, pOS->d_id,
                                                pOS-
>c_id, pOS->c_first, pOS->c_middle, pOS->c_last,
                                                pOS-
>c_balance,
                                                pOS-
>o_id, pOS->o_entry_d, pOS->o_carrier_id);

                                                for (i=0; i<pOS-
>o_ol_cnt; i++)
                                                {

        sprintf(ol_text,

        "  %04d       %06d     %02d     $%08.2lf      %-10s\n",

        pOS->OlOrderStatusData[i].ol_supply_w_id,

        pOS->OlOrderStatusData[i].ol_i_id,

        pOS->OlOrderStatusData[i].ol_quantity,

        pOS->OlOrderStatusData[i].ol_amount,

        pOS->OlOrderStatusData[i].ol_delivery_d);

        strcat(text, ol_text);
                                                }

                                break;
                        case DEADLOCK:
                                strcat(text, "ERROR:
DEADLOCK");
                                break;
                        case SQL_ERROR:
                                strcpy(text, "ERROR:
SQL Error");
                                break;
                        } // End switch
                } // End else

        sprintf(html,ResponseHTML,"OrderStatus
Response",pUserData[User]->ButtonBar,text);

        } // ProcessOrderStatus
//
//
//--------------------------------ParseOrderStatus---------------------------------
// parse OrderStatus input string
// fill in OrderStatus data structure
// returns SUCCESS if successful, error code if parsing error
// form of input string:
//   D_ID=12&C_ID=1234&C_LAST=1234567890123456
//  |     |       |
//  pD    pC      pCL
//
```

```
//   (either C_ID or C_LAST must be blank)
//
int ParseOrderStatus(char* pD, ORDER_STATUS_DATA* pOS)
        {

        int len;
        char str[7];
        char *pC, *pCL;
        BOOL c_id_blank = FALSE, c_last_blank = FALSE;

        // Check input string starts with D_ID=
        if (strncmp(pD, "D_ID=", 5)) return ERR_GENERIC;

        // Copy D_ID string to str, convert to integer, checking for non-
numerics
        pC = 1 + strchr(pD, '&');
        if ((len = pC - pD - 6) < 1) return ERR_BLANK_DID;
                // blank D_ID field
        strncpy(str, pD + 5, len);
        str[len] = '\0';
        if (!IsValidNonNegShort(str, &pOS->d_id)) return
ERR_NONNUM_DID; // non-num D_ID

        // Copy C_ID string to str, convert to integer, checking for non-
numerics
        pOS->c_id = 0;
        pCL = 1 + strchr(pC, '&');
        if ((len = pCL - pC - 6) < 1) c_id_blank = TRUE;
        // blank C_ID field
        else
                {
                strncpy(str, pC + 5, len);
                str[len] = '\0';
                if (!IsValidNonNegLong(str, &pOS->c_id)) return
ERR_NONNUM_CID;//non-num C_ID
                }

        // Copy C_LAST string to c_last
        strcpy(pOS->c_last, pCL + 7);
        if (!strlen(pOS->c_last)) c_last_blank = TRUE;
                // blank C_LAST field
        // both are blank or both are filled in (error either way)
        if (c_id_blank == c_last_blank) return ERR_CUST_NAME_ID;
        return SUCCESS;
        } // ParseOrderStatus
//
//
//---------------------------------ProcessDelivery-----------------------------------
// process Delivery form
//
void ProcessDelivery(char* cmd_input_str, int User, char* html)
        {
        int rc;
        char text[256];
        LARGE_INTEGER tick_count;
        DELIVERY_DATA* pDel = (DELIVERY_DATA*)
pUserData[User]->TransData;

        memset(pUserData[User]->TransData,'\0',sizeof(pUserData[User]-
>TransData));
        pDel->w_id = pUserData[User]->w_id;

        QueryPerformanceCounter(&tick_count);
        // milliseconds since bootup
        pDel->queued_time = (int) floor(1000.0 *
                ((double)(tick_count.QuadPart)/freqd);

        if ((rc = ParseDelivery(cmd_input_str, pDel)) != SUCCESS)
                sprintf(text, "ERROR: %s", err_text[rc]);

        else
                {
                // call Delivery database lookup
                rc = TMDelivery(pDel);
                switch(rc)
                        {
                        case SUCCESS:
                    sprintf(text, "
Delivery\n"
                                "Warehouse: %04d\n\n"
                                "Carrier Number: %02d\n\n"
                                "Execution Status: Delivery has
been queued\n",
                                pDel->w_id, pDel-
>o_carrier_id);
                                break;
                        case DEADLOCK:
                                strcat(text, "ERROR:
DEADLOCK");
                                break;
                        case SQL_ERROR:
                                strcpy(text, "ERROR:
SQL Error");
                                break;
                        } // End switch

                } // End else

        sprintf(html,ResponseHTML,"Delivery
Response",pUserData[User]->ButtonBar,text);

        } // ProcessDelivery
//
//
//---------------------------------ParseDelivery-----------------------------------
// parse Delivery input string
// fill in Delivery data structure
// returns SUCCESS if successful, error code if parsing error
// form of input string:
//   CARRIER=12
//   |
//   pCAR
//
int ParseDelivery(char* pCAR, DELIVERY_DATA* pDel)
        {

        char str[7];

        // Check input string starts with CARRIER=
        if (strncmp(pCAR, "CARRIER=", 8)) return ERR_GENERIC;

        // Copy CARRIER string to str, convert to integer, checking for
non-numerics
        strcpy(str, pCAR + 8);
        if (!strlen(str)) return ERR_BLANK_CARRIER;
        // blank CARRIER field
        if (!IsValidNonNegShort(str, &pDel->o_carrier_id))
                return ERR_NONNUM_CARRIER; // CARRIER
field is negative or invalid
        return SUCCESS;
        } // ParseDelivery
//
//---------------------------------ProcessStockLevel-----------------------------------
// process StockLevel form
//
void ProcessStockLevel(char* cmd_input_str, int User, char* html)
        {
        int rc;
        char text[256];
        STOCK_LEVEL_DATA* pSL = (STOCK_LEVEL_DATA*)
pUserData[User]->TransData;

        memset(pUserData[User]->TransData,'\0',sizeof(pUserData[User]-
>TransData));
        pSL->w_id = pUserData[User]->w_id;
        pSL->d_id = pUserData[User]->d_id;
        if ((rc = ParseStockLevel(cmd_input_str, pSL)) != SUCCESS)
                sprintf(text, "ERROR: %s", err_text[rc]);

        else
                {
                // call StockLevel database lookup
                rc = TMStockLevel(pSL);

                switch(rc)
                        {
                        case SUCCESS:
                                sprintf(text, "
Stock-Level\n"
        "Warehouse: %04d   District: %02d\n\n"
                                "Stock
Level Threshold: %02d\n\n"
                                "low
stock: %03d\n",
                                pSL-
>w_id, pSL->d_id, pSL->threshold, pSL->low_stock);
                                break;
                        case DEADLOCK:
                                strcat(text, "ERROR:
DEADLOCK");
                                break;
                        case SQL_ERROR:
                                strcpy(text, "ERROR:
SQL Error");
                                break;
                        } // End switch
                } // End else
```

```
		sprintf(html,ResponseHTML,"StockLevel
Response",pUserData[User]->ButtonBar,text);

		} // ProcessStockLevel
//
//
//---------------------------------ParseStockLevel-----------------------------------
// parse StockLevel input string
// fill in StockLevel data structure
// returns SUCCESS if successful, error code if parsing error
// form of input string:
//   THRESHOLD=12
//   |
//   pT
//
int ParseStockLevel(char* pT, STOCK_LEVEL_DATA* pSL)
		{

		char str[7];

		// Check input string starts with THRESHOLD=
		if (strncmp(pT, "THRESHOLD=", 10)) return ERR_GENERIC;

		// Copy THRESHOLD string to str, convert to integer, checking
for non-numerics
		strcpy(str, pT + 10);
		if (!strlen(str)) return ERR_BLANK_THRESHOLD;
				// blank THRESHOLD field
		if (!IsValidNonNegShort(str, &pSL->threshold)) return
ERR_NONNUM_THRESHOLD;
		return SUCCESS;
		} // ParseStockLevel
//
//---------------------------------ProcessLogout-------------------------------------
// process logout request; reset entry in UserData array, free memory
//
void ProcessLogout(int User)
		{
		EnterCriticalSection(&login_crit_sec);
		pUserData[User] = 0;
		free(pUserData[User]);
		LeaveCriticalSection(&login_crit_sec);

		} // ProcessLogout
//
//
//-------------------------------ReadTPCCRegParams-----------------------------------
// read client-specific and database params from registry
//
void ReadTPCCRegParams()
		{
		char SubKey[100];
		char Value[100];
		HKEY Key;
		DWORD Type;
		DWORD lenData = 4;
		unsigned char Text[200];
		union
				{
				unsigned char data_char[4];
				unsigned int data_int;
				}
				RegDWORD;
		char message[100];


		strcpy(SubKey, "Software\\TPCC");

		strcpy(Value, "MaxUsersThisClient");
		RegOpenKeyEx(HKEY_LOCAL_MACHINE, SubKey, 0,
		KEY_QUERY_VALUE, &Key);
		RegQueryValueEx(Key, Value, 0, &Type, RegDWORD.data_char,
&lenData);
		max_users_this_client = (int) RegDWORD.data_int;
		sprintf(message, "max_users_this_client= %d",
max_users_this_client);
		WriteErrorLog(message);

		strcpy(Value, "NumberOfWarehousesTotal");
		RegOpenKeyEx(HKEY_LOCAL_MACHINE, SubKey, 0,
KEY_QUERY_VALUE, &Key);
		RegQueryValueEx(Key, Value, 0, &Type, RegDWORD.data_char,
&lenData);
		n_warehouses_total = (int) RegDWORD.data_int;
		sprintf(message, "n_warehouses_total= %d", n_warehouses_total);
		WriteErrorLog(message);

		lenData = 200;
		strcpy(Value, "DLLPath");
		RegOpenKeyEx(HKEY_LOCAL_MACHINE, SubKey, 0,
KEY_QUERY_VALUE, &Key);
		RegQueryValueEx(Key, Value, 0, &Type, Text, &lenData);
		strncpy(dll_path, (const char*)Text, lenData);
		sprintf(message, "dll_path= %s", dll_path);
		WriteErrorLog(message);

		} // ReadTPCCRegParams
//
//
//---------------------------------OpenErrorLog--------------------------------------
// create and open error log w/ filename error.log in directory log_path
//
BOOL OpenErrorLog()
		{
		char errorlog_fn[250];
		char SubKey[100];
		char Value[] = "LogPath";
		HKEY Key;
		DWORD Type;
		DWORD lenData = 200;
		unsigned char Text[200];

		// read log_path from registry key \Software\TPCC\LogPath
		strcpy(SubKey, "Software\\TPCC");
		RegOpenKeyEx(HKEY_LOCAL_MACHINE, SubKey, 0,
KEY_QUERY_VALUE, &Key);
		RegQueryValueEx(Key, Value, 0, &Type, Text, &lenData);
		ExpandEnvironmentStrings((const char*) Text, log_path, 250);

		// Create file pathname, open file, return FALSE if error
		sprintf(errorlog_fn, "%s\\error_client.log", log_path);
		if((fp_errorlog = fopen(errorlog_fn, "a+")) == NULL)	return
FALSE;
		else return TRUE;
		} // OpenErrorLog
//
//
//---------------------------------WriteErrorLog-------------------------------------
// create and open error log w/ filename error.log in directory log_path
//
void WriteErrorLog(char* message)
		{
		char d[10];
		char t[10];
		struct _timeb tb;

		_strdate(d);
		_strtime(t);
		_ftime(&tb);
		fprintf(fp_errorlog, "%s %s.%03u Thread: 0x%03X %s\n",
				d, t, tb.millitm, GetCurrentThreadId(), message);
		fflush(fp_errorlog);
		} // WriteErrorLog
//
//
//
//---------------------------------IsValidNonNegShort--------------------------------
// check if text string represents a valid, non-negative SHORT and convert
//
BOOL IsValidNonNegShort(char* str, short* number)
		{
		char *non_numeric_chars;
		BOOL all_trailing_chars_are_blanks;
		int i, l;

		*number = (short) strtol(str, &non_numeric_chars, 10);

		// If strtol says there are non numeric trailing chars we need to
check
		// in case they are all blank
		if (l=strlen(non_numeric_chars))
				{
				if (l == (int) strlen(str)) return FALSE;  // entire str
is non-numeric
				all_trailing_chars_are_blanks = TRUE;
				for (i=0; i<l; i++)
						{
						if (*(non_numeric_chars + i) != ' '
&&
								*(non_numeric_chars
+ i) != '+')		// blanks are +'s
								{
		all_trailing_chars_are_blanks = FALSE;
								break;
								}
						}
```

```
                        if (!all_trailing_chars_are_blanks) return FALSE;
                        }

                // Check number not negative
                if (*number < 0) return FALSE;


                return TRUE;
                }

//
//
//--------------------------------IsValidNonNegLong---------------------------------
// check if text string represents a valid, non-negative LONG and convert
//
BOOL IsValidNonNegLong(char* str, long* number)
                {
                char *non_numeric_chars;
                BOOL all_trailing_chars_are_blanks;
                int i, l;

                *number = (long) strtol(str, &non_numeric_chars, 10);

                // If strtol says there are non numeric trailing chars we need to
check
                // in case they are all blank
                if (l=strlen(non_numeric_chars))
                        {
                        if (l == (int) strlen(str)) return FALSE;  // entire str
is non-numeric
                        all_trailing_chars_are_blanks = TRUE;
                        for (i=0; i<l; i++)
                                {
                                if (*(non_numeric_chars + i) != ' '
&&
                                                *(non_numeric_chars
+ i) != '+')             // blanks are +'s
                                                {

                all_trailing_chars_are_blanks = FALSE;
                                                break;
                                                }
                                }
                        if (!all_trailing_chars_are_blanks) return FALSE;
                        }

                // Check number not negative
                if (*number < 0) return FALSE;


                return TRUE;
                }

//
//
//--------------------------------IsValidNonNegDouble---------------------------------
// check if text string represents a valid, non-negative DOUBLE and convert
//
BOOL IsValidNonNegDouble(char* str, double* number)
                {
                char *non_numeric_chars;
                BOOL all_trailing_chars_are_blanks;
                int i, l;

                *number = (double) strtod(str, &non_numeric_chars);

                // If strtol says there are non numeric trailing chars we need to
check
                // in case they are all blank
                if (l=strlen(non_numeric_chars))
                        {
                        if (l == (int) strlen(str)) return FALSE;  // entire str
is non-numeric
                        all_trailing_chars_are_blanks = TRUE;
                        for (i=0; i<l; i++)
                                {
                                if (*(non_numeric_chars + i) != ' '
&&
                                                *(non_numeric_chars
+ i) != '+')             // blanks are +'s
                                                {

                all_trailing_chars_are_blanks = FALSE;
                                                break;
                                                }
                                }
                        if (!all_trailing_chars_are_blanks) return FALSE;
                        }

                // Check number not negative
                if (*number < 0) return FALSE;


                return TRUE;
                }

//

//--------------------------------End of tpcc.cpp---------------------------------
```

tmclient.c

```
//----------------------tmclient.c: Dell TPC-C Client / Web Server----------------------
˜
//
˜
// Copyright (c) 1997 Dell Computer Corporation, All Rights Reserved
˜
//
˜
// Author: James Jordan/Dave Jaffe
                Last modified: 10/8/97
//
// Audited: Richard Gimarc  Performance Metrics Inc.  10/9/97
//
//
// Transaction Monitor client code; compiled into tpcc.dll ISAPI; included by
tpcc.h
//
//

/*
 * Copyright (c) 1984, 1985, 1986, 1987, 1988, 1989, 1990,
 * 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998
 * Sequent Computer Systems, Inc.   All rights reserved.
 *
 * This software is furnished under a license and may be used
 * only in accordance with the terms of that license and with the
 * inclusion of the above copyright notice.   This software may not
 * be provided or otherwise made available to, or used by, any
 * other person.  No title to or ownership of the software is
 * hereby transferred.
 */

static char *RCSID="@(#)$Id: tmclient.c,v 1.4 1998/10/01 01:55:49 administrator
Exp administrator $";

#ifdef CALL_TPALLOC_ONCE
int get_max_tpalloc_size(void)
{
                int size = 0;

                if (NO_len_ot > size)
                                size = NO_len_ot;
                if (PY_len_ot > size)
                                size = PY_len_ot;
                if (OS_len_ot > size)
                                size = OS_len_ot;
                if (DL_len_ot > size)
                                size = DL_len_ot;
                if (SL_len_ot > size)
                                size = SL_len_ot;
                return size;
}
#endif // CALL_TPALLOC_ONCE

//---------------------------------TMClientInit-----------------------------------------
// Initializes transaction monitor client
//
BOOL TMClientInit()
                {
                int  spid = 0;

                tls_idx = TlsAlloc();
                if (tls_idx < 0)
                                {
                                WriteErrorLog("ERROR: TMInit -> TLS index
failed to allocate!");
                                return SQL_ERROR;
                                }

#ifdef CALL_TPALLOC_ONCE
                if ((tls_tpalloc_idx = TlsAlloc()) < 0)
                                {
                                WriteErrorLog("ERROR: TMInit -> TLS tpalloc
index failed to allocate!");
                                return SQL_ERROR;
```

```
                        }
#endif // CALL_TPALLOC_ONCE


   return TRUE;
        } // TMInit
//
//
//---------------------------------TMClientExit----------------------------------------
// Exits transaction monitor client
//
void TMClientExit()
                        {
                        if ((TlsFree(tls_idx)) == 0) WriteErrorLog("Failed
to free TLS");
#ifdef CALL_TPALLOC_ONCE
                        tpfree(TlsGetValue(tls_tpalloc_idx));
                        if ((TlsFree(tls_tpalloc_idx)) == 0)
WriteErrorLog("ERROR: Failed to free tpalloc TLS");
                        tpterm();
#endif // CALL_TPALLOC_ONCE

                        }
//
//
//
//---------------------------------TMNewOrder----------------------------------------
// Transaction Monitor NewOrder client function: calls Tuxedo server TMNO
//
int TMNewOrder(NEW_ORDER_DATA* p1)
        {
                        int rc;
                        NEW_ORDER_DATA *p2;

                        if (IsTpInit() == -1)
                                {
                                TPPrintError("IsTPInit_NO");
                                return SQL_ERROR;
                                }

#ifdef CALL_TPALLOC_ONCE
                        p2 = (NEW_ORDER_DATA
*)TlsGetValue(tls_tpalloc_idx);
#else // CALL_TPALLOC_ONCE
                        if((p2 = (NEW_ORDER_DATA *)
tpalloc("CARRAY", NULL, NO_len_ot)) == NULL)
                                {
                                TPPrintError("TPALLOC_NO");
                                return SQL_ERROR;
                                }
#endif // CALL_TPALLOC_ONCE

                        memcpy(p2,p1,NO_len_ot);
                        rc = tpcall("TMNO",(char *) p2,
NO_len_ot,(char**)&p2,&NO_len_ot, TPSIGRSTRT);
                        if ( rc == -1)
                                {
                                TPPrintError("TPCALL_NO");
                                return SQL_ERROR;
                                };
                        memcpy(p1,p2,NO_len_ot);
#ifndef CALL_TPALLOC_ONCE
                        tpfree((char*)p2);
#endif // CALL_TPALLOC_ONCE
                        return tpurcode;
        }
//
//
//---------------------------------TMPayment----------------------------------------
// Transaction Monitor Payment client function: calls Tuxedo server TMPY
//
int TMPayment(PAYMENT_DATA* p1)
        {
                        int rc;
                        PAYMENT_DATA *p2;

                        if (IsTpInit() == -1)
                                {
                                TPPrintError("IsTPInit_PY");
                                return SQL_ERROR;
                                }

#ifdef CALL_TPALLOC_ONCE
                        p2 = (PAYMENT_DATA
*)TlsGetValue(tls_tpalloc_idx);
#else // CALL_TPALLOC_ONCE
                        if((p2 = (PAYMENT_DATA *)
tpalloc("CARRAY", NULL, PY_len_ot)) == NULL)
                                {
                                TPPrintError("TPALLOC_PY");
                                return SQL_ERROR;
                                }
#endif // CALL_TPALLOC_ONCE
                        memcpy(p2,p1,PY_len_ot);
                        rc = tpcall("TMPY",(char *) p2,
PY_len_ot,(char**)&p2,&PY_len_ot, TPSIGRSTRT);
                        if (rc  == -1)
                                {
                                TPPrintError("TPCALL_PY");
                                return SQL_ERROR;
                                };
                        memcpy(p1,p2,PY_len_ot);
#ifndef CALL_TPALLOC_ONCE
                        tpfree((char*)p2);
#endif // CALL_TPALLOC_ONCE
                        return tpurcode;
        }
//
//
//---------------------------------TMOrderStatus----------------------------------------
// Transaction Monitor OrderStatus client function: calls Tuxedo server TMOS
//
int TMOrderStatus(ORDER_STATUS_DATA* p1)
        {
                        int rc;
                        ORDER_STATUS_DATA *p2;

                        if (IsTpInit() == -1)
                                {
                                TPPrintError("IsTPInit_OS");
                                return SQL_ERROR;
                                }

#ifdef CALL_TPALLOC_ONCE
                        p2 = (ORDER_STATUS_DATA
*)TlsGetValue(tls_tpalloc_idx);
#else // CALL_TPALLOC_ONCE
                        if((p2 = (ORDER_STATUS_DATA *)
tpalloc("CARRAY", NULL, OS_len_ot)) == NULL)
                                {
                                TPPrintError("TPALLOC_OS");
                                return SQL_ERROR;
                                }
#endif // CALL_TPALLOC_ONCE
                        memcpy(p2,p1,OS_len_ot);
                        rc = tpcall("TMOS",(char *) p2,
PY_len_ot,(char**)&p2,&PY_len_ot, TPSIGRSTRT);
                        if ( rc == -1)
                                {
                                TPPrintError("TPCALL_OS");
                                return SQL_ERROR;
                                };
                        memcpy(p1,p2,OS_len_ot);
#ifndef CALL_TPALLOC_ONCE
                        tpfree((char*)p2);
#endif // CALL_TPALLOC_ONCE
                        return tpurcode;
        }
//
//
//---------------------------------TMDelivery----------------------------------------
// Transaction Monitor Delivery client function: calls Tux server TMDL
asynchronously
//
int TMDelivery(DELIVERY_DATA *p1)
        {
                        int rc;
                        DELIVERY_DATA *p2;

                        if (IsTpInit() == -1)
                                {
                                TPPrintError("IsTPInit_DL");
                                return SQL_ERROR;
                                }


#ifdef CALL_TPALLOC_ONCE
                        p2 = (DELIVERY_DATA
*)TlsGetValue(tls_tpalloc_idx);
#else // CALL_TPALLOC_ONCE
                        if((p2 = (DELIVERY_DATA *)
tpalloc("CARRAY", NULL, DL_len_ot)) == NULL)
                                {
                                TPPrintError("TPALLOC_DL");
                                return SQL_ERROR;
                                }
#endif // CALL_TPALLOC_ONCE
                        memcpy(p2,p1,DL_len_ot);
```

```
                        rc = tpacall("TMDL",(char *) p2,
DL_len_ot,TPNOREPLY);
                        Sleep((int)(1000.0*((double)
rand()/(double)RAND_MAX)));
                        if ( rc  == -1)
                                {
                                TPPrintError("TPCALL_DL");
                                return SQL_ERROR;
                                };
#ifndef CALL_TPALLOC_ONCE
                        tpfree((char*)p2);
#endif // CALL_TPALLOC_ONCE
                        return SUCCESS;
                }
//
//
//---------------------------------TMStockLevel-----------------------------------------
// Transaction Monitor StockLevel client function: calls Tuxedo server TMSL
//
int TMStockLevel(STOCK_LEVEL_DATA* p1)
                {
                        int rc;
                        STOCK_LEVEL_DATA *p2;

                        if (IsTpInit() == -1)
                                {
                                TPPrintError("IsTPInit_SL");
                                return SQL_ERROR;
                                }

#ifdef CALL_TPALLOC_ONCE
                        p2 = (STOCK_LEVEL_DATA
*)TlsGetValue(tls_tpalloc_idx);
#else // CALL_TPALLOC_ONCE
                        if((p2 = (STOCK_LEVEL_DATA *)
tpalloc("CARRAY", NULL, SL_len_ot)) == NULL)
                                {
                                TPPrintError("TPALLOC_SL");
                                return SQL_ERROR;
                                }
#endif // CALL_TPALLOC_ONCE
                        memcpy(p2,p1,SL_len_ot);
                        rc = tpcall("TMSL",(char *) p2,
PY_len_ot,(char**)&p2,&PY_len_ot, TPSIGRSTRT);
                        if (rc  == -1)
                                {
                                TPPrintError("TPCALL_SL");
                                return SQL_ERROR;
                                };
                        memcpy(p1,p2,SL_len_ot);
#ifndef CALL_TPALLOC_ONCE
                        tpfree((char*)p2);
#endif // CALL_TPALLOC_ONCE
                        return tpurcode;
                }
//
//
//---------------------------------IsTPInit()-------------------------------------------
//
int IsTpInit()
                {
                TPINIT *tpinf;
#ifdef CALL_TPALLOC_ONCE
                void *tpallocp;
                int retry_tpalloc = 0;
#endif // CALL TPALLOC_ONCE
                int x=1, rc=-1, cnt=0;
                static int num_tpinits=0;

                if(!(TlsGetValue(tls_idx)))
                        {
                        EnterCriticalSection(&TLS_crit_sec);
                        while(rc == -1)
                                {

                                // tpalloc with retry
                                while (retry_tpalloc < 5)
                                        {
                                        tpinf = (TPINIT *)
tpalloc("TPINIT","",sizeof(TPINIT));
                                        if (tpinf != NULL)
                                                {
                                                tpinf-
>flags |= TPMULTICONTEXTS ;
                                                break;
                                                }
                                                else
                                                {
                                                TPPrintError("ERROR: TPALLOC(TPINIT struct)");

                                                retry_tpalloc++;

                                                Sleep(2 * retry_tpalloc);
                                                }
                                        } // while retry tpalloc
                                if (retry_tpalloc >= 5)
                                        {

                                LeaveCriticalSection(&TLS_crit_sec);

                                return(SQL_ERROR);
                                        }


                                itoa(++num_tpinits, tpinf->cltname,
10);
                                rc=tpinit(tpinf);
                                if (rc != -1) TlsSetValue(tls_idx,&x);
                                else TPPrintError("TPINIT");
                                tpfree((char*)tpinf);

                                if(cnt > 5)
                                        {
#ifdef CALL_TPALLOC_ONCE

                                LeaveCriticalSection(&TLS_crit_sec);

                                return(SQL_ERROR);
#else
                                        rc=SQL_ERROR;
                                        break;
#endif // CALL_TPALOC_ONCE
                                        }
                                cnt++;
                                Sleep(2 * cnt);
                                }
                        LeaveCriticalSection(&TLS_crit_sec);
#ifndef CALL_TPALLOC_ONCE
                        return (rc);
#endif
                        }

#ifdef CALL_TPALLOC_ONCE
                if(!(TlsGetValue(tls_tpalloc_idx)))
                        {
                        EnterCriticalSection(&TLS_crit_sec);          // begin
crit
                        tpallocp = (void *) tpalloc("CARRAY", NULL,
get_max_tpalloc_size());
                        if (tpallocp == NULL)
                                {

                LeaveCriticalSection(&TLS_crit_sec);
                                TPPrintError("ERROR:
TPALLOC(CARRAY)");
                                return SQL_ERROR;
                                }
                        TlsSetValue(tls_tpalloc_idx, tpallocp);
                        LeaveCriticalSection(&TLS_crit_sec);          // end
crit
                        }
#endif // CALL_TPALLOC_ONCE

                return 0;
                } //             End IsTPInit()
//
//
//---------------------------------TPPrintError()---------------------------------------
// print tuxedo error message
//
//

static void TPPrintError(char *msg)
                {
                 char errbuf[512];

                sprintf(errbuf, "TPERROR: %s: %s\n",
                                msg, tpstrerror(tperrno));
                WriteErrorLog(errbuf);
                return;
                } // end of TPPrintError

//
//
//
//---------------------------------End of tmclient.c------------------------------------
```

Appendix A – Application Source Code

Commands For Compilinig and Linking tpcc.dll

```
cl.exe /nologo /Ox /MD /D CALL_TPALLOC_ONCE /D WIN32
/D _WINDOWS /D _TMSTHREADS  /I d:\tuxedo\include /c tpcc.c
link.exe /nologo /subsystem:windows /dll /incremental:no
/machine:I386 /def:tpcc.def /implib:tpcc.lib /libpath:d:\tuxedo\lib kernel32.lib
advapi32.lib libtux.lib libtux2.lib libbuft.lib libgp.lib libwsc.lib /out:"tpcc.dll"
tpcc.obj
```

Tuxedo Server Source Code

Tmserver_dell.h

```
//----------------------tmserver.h: Dell TPC-C Transaction Monitor Server-----------
˜
//
˜
// Copyright (c) 1997 Dell Computer Corporation, All Rights Reserved
˜
//
˜
// Author: James Jordan                                                    Last
modified: 10/8/97
˜
//

// Audited: Richard Gimarc, Performance Metrics Inc. 10/9/97
//
// header file for Transaction Monitor tpcc server code
//

#ifndef _TM_SERVER_
#define _TM_SERVER_

#include <windows.h>
#include <stdio.h>
#include <process.h>
#include <httpext.h>
#include <sys/types.h>
#include <sys/timeb.h>
#include <math.h>
#include <stdarg.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <io.h>
#include <ctype.h>

//                  tuxedo header
#include <tmenv.h>
#include <xa.h>
#include <atmi.h>
#include <userlog.h>

// Oracle Include files for OCI
#include "tpcc_ora.h"
// tpcc include files
#include "trans_dell.h"                 //                  transaction data
structure definitions

//Functions
void ReadTPCCRegParams();
BOOL OpenDeliveryLog();
int  OpenErrorLog();
void WriteErrorLog(char* message);


// Transaction Monitor functions
void TMNO (TPSVCINFO *rqst);
void TMPY (TPSVCINFO *rqst);
void TMOS (TPSVCINFO *rqst);
void TMDL (TPSVCINFO *rqst);
void TMSL (TPSVCINFO *rqst);

#ifndef ORACLE
// Database-specific functions for sqlserver
void DBInit();
void DBExit();
BOOL DBOpenConnection(DBPROCESS** ppDbproc, char* server, char*
database, char* user,
                                                        char* password,
char* app, int* spid, long* pack_size);
void DBCloseConnection(DBPROCESS* pDbproc);
int  DBNewOrder(DBPROCESS* pDbproc, NEW_ORDER_DATA*
pNewOrder);
int  DBPayment(DBPROCESS* pDbproc, PAYMENT_DATA* pPayment);
int  DBOrderStatus(DBPROCESS* pDbproc, ORDER_STATUS_DATA*
pOrderStatus);
int  DBDelivery(DBPROCESS* pDbproc, DELIVERY_DATA* pDelivery);
int  DBStockLevel(DBPROCESS* pDbproc, STOCK_LEVEL_DATA*
pStockLevel);
#endif
// Variables read from registry key Software\TPCC on LocalMachine
//    Same on all clients:
char server[32];                                    // Name of database
server machine

                        //   (REG_SZ: DatabaseServer)
char database_name[32];             // Name of database (REG_SZ:
DatabaseName)
char database_user[32];             // Database user login name
(REG_SZ: DatabaseUser)
char database_passwd[32];    // Database user login password

                        //   (REG_SZ: DatabasePassword)
char log_path[250];                                 // Path for delivery
and other logs

                        //   (REG_EXPAND: LogPath)

// Return codes from TM stored procedure calls
#define SQL_ERROR           -1          // Usually incorrect
C_ID or C_LAST
#define SUCCESS                         0           //
Success
#define DEADLOCK            1           // Still deadlocked
after specified DEADLOCK_RETRYs
#define     INVALID_ITEM    2           // Invalid item in
NewOrder

#define DEADLOCK_WAIT 10
#define DEADLOCK_RETRY      5

//Global variables
FILE *fp_delivlog, *fp_errorlog;

LARGE_INTEGER freq;
double freqd;

CRITICAL_SECTION deliv_write_crit_sec;

#ifndef ORACLE
// dbproc handle for connection to database.
DBPROCESS* hDB;

#include "sqlfuncs.c"               //    MSSQL tpc-c transaction
function wrappers
#endif


#endif// _TM_SERVER_
```

tpcc_dell.h

```
//----------------------tpcc.h: Dell TPC-C Client / Web Server--------------------------
˜
//
˜
// Copyright (c) 1997 Dell Computer Corporation, All Rights Reserved
˜
//
˜
// Author: Dave Jaffe                                                      Last
modified: 10/8/97
//
// Audited: Richard Gimarc  Performance Metrics Inc.  10/9/97
//
// header file for tpcc.dll MS ISAPI DLL for TPC-C Benchmark
//

/*
 * Copyright (c) 1984, 1985, 1986, 1987, 1988, 1989, 1990,
 * 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998
 * Sequent Computer Systems, Inc.   All rights reserved.
 *
 * This software is furnished under a license and may be used
 * only in accordance with the terms of that license and with the
 * inclusion of the above copyright notice.   This software may not
 * be provided or otherwise made available to, or used by, any
 * other person.  No title to or ownership of the software is
 * hereby transferred.
```

Appendix A – Application Source Code

```
 */

static char *RCSID = "$@(#)$Id: tpcc.h,v 1.6 1998/09/29 08:12:11 administrator
Exp administrator $";

#ifndef _TPCC_H_
#define _TPCC_H_

#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <process.h>
#include <httpext.h>
#include <sys/types.h>
#include <sys/timeb.h>
#include <math.h>
#include <time.h>
//                  tuxedo header
#include <tmenv.h>
#include <xa.h>
#include <atmi.h>

#include "html_dell.h"
#include "trans_dell.h"                         // Transaction data structures

// Commands
enum _command
                    {
                    Login,
                    NewOrderFormRequest,
                    NewOrder,
                    PaymentFormRequest,
                    Payment,
                    OrderStatusFormRequest,
                    OrderStatus,
                    DeliveryFormRequest,
                    Delivery,
                    StockLevelFormRequest,
                    StockLevel,
                    Logout
                    };

char* command_name[] =
                    {
                    "Login",
                    "NewOrderFormRequest",
                    "NewOrder",
                    "PaymentFormRequest",
                    "Payment",
                    "OrderStatusFormRequest",
                    "OrderStatus",
                    "DeliveryFormRequest",
                    "Delivery",
                    "StockLevelFormRequest",
                    "StockLevel",
                    "Logout"
                    };


// Functions

BOOL ParseInput(char* pCMD, enum _command* Command, int* User, char**
pInputs);
void ProcessNewOrder(char* cmd_input_str, int User, char* html);
void ProcessPayment(char* cmd_input_str, int User, char* html);
void ProcessOrderStatus(char* cmd_input_str, int User, char* html);
void ProcessStockLevel(char* cmd_input_str, int User, char* html);
void ProcessDelivery(char* cmd_input_str, int User, char* html);

void ProcessLogin(char* cmd_input_str, char* html, BOOL* KeepConn);
int  ParseLogin(char* cmd_input_str, short* w_id, short* d_id);
int  ParseNewOrder(char* cmd_input_str, NEW_ORDER_DATA*
pNewOrderData);
int  ParsePayment(char* cmd_input_str, PAYMENT_DATA* pPaymentData);
int  ParseOrderStatus(char* cmd_input_str, ORDER_STATUS_DATA*
pOrderStatusData);
int  ParseDelivery(char* cmd_input_str, DELIVERY_DATA* pDeliveryData);
int  ParseStockLevel(char* cmd_input_str, STOCK_LEVEL_DATA*
pStockLevelData);
void ProcessLogout(int User);
void ReadTPCCRegParams();
BOOL OpenDeliveryLog();
void WriteErrorLog(char* message);
BOOL OpenErrorLog();
BOOL IsValidNonNegShort(char* str, short* number);
BOOL IsValidNonNegLong(char* str, long* number);
BOOL IsValidNonNegDouble(char* str, double* number);

// Transaction Monitor client functions

BOOL TMClientInit();
void TMClientExit();
int  TMNewOrder(NEW_ORDER_DATA* pNewOrder);
int  TMPayment(PAYMENT_DATA* pPayment);
int  TMOrderStatus(ORDER_STATUS_DATA* pOrderStatus);
int  TMDelivery(DELIVERY_DATA *pDel);
int  TMStockLevel(STOCK_LEVEL_DATA* pStockLevel);

//                  tuxedo functions
static DWORD tls_idx;                                                   //
thread local storage index
static int IsTpInit();                                                  // use
TLS to determine if tpinit needs to be run
static void TPPrintError(char *msg);        // writes tperrno to logfile
#ifdef CALL_TPALLOC_ONCE
static DWORD tls_tpalloc_idx;
#endif /* CALL_TPALLOC_ONCE */

// Critical sections
CRITICAL_SECTION login_crit_sec;
CRITICAL_SECTION TLS_crit_sec;

// MAX_USERS_ALL_CLIENTS is maximum number of users between all clients
#define MAX_USERS_ALL_CLIENTS 15000

// Variables read from registry key Software\TPCC on LocalMachine:

//    Client specific:
int max_users_this_client;      // Maximum number of users client can handle

                                //  (REG_DWORD: MaxUsersThisClient)
//    Same on all clients:
int n_warehouses_total;                 // Number of warehouses in database

                                //   (REG_DWORD: NumberOfWarehouses)
char dll_path[250];                                             // HTTP path of
tpcc.dll

                                //   (REG_SZ: DLLPath)
char log_path[250];                                             // Path for delivery
and other logs

                                //   (REG_EXPAND: LogPath)


// Global variables
long NO_len_ot = sizeof(NEW_ORDER_DATA);
long PY_len_ot = sizeof(PAYMENT_DATA);
long OS_len_ot = sizeof(ORDER_STATUS_DATA);
long DL_len_ot = sizeof(DELIVERY_DATA);
long SL_len_ot = sizeof(STOCK_LEVEL_DATA);

// Return codes from TM stored procedure calls
#define SQL_ERROR                   -1          // Usually incorrect
C_ID or C_LAST
#define SUCCESS                                 0           //
Success
#define DEADLOCK                    1           // Still deadlocked
after specified DEADLOCK_RETRYs
#define     INVALID_ITEM            2           // Invalid item in
NewOrder

#define DEADLOCK_WAIT 10
#define DEADLOCK_RETRY              5

typedef struct
                    {
                    short w_id;
                    short d_id;
                    BYTE TransData[2048];
                    char ButtonBar[2048];
                    } UserData;

UserData* pUserData[MAX_USERS_ALL_CLIENTS];

FILE *fp_delivlog, *fp_errorlog;

LARGE_INTEGER freq, tick_count0;
double freqd;

// Error codes and associated error text from browser input parse routines
#define ERR_GENERIC                                             1
#define ERR_BLANK_WID                           2
#define ERR_NONNUM_WID                                          3
#define ERR_BLANK_DID                           4
#define ERR_NONNUM_DID                                          5

#define ERR_BLANK_CID                           6
#define ERR_NONNUM_CID                                          7
```

```
#define ERR_BLANK_CWID                          8
#define ERR_NONNUM_CWID                         9
#define ERR_BLANK_CDID                  10
#define ERR_NONNUM_CDID                         11
#define ERR_BLANK_AMOUNT                12
#define ERR_NONNUM_AMOUNT               13
#define ERR_BLANK_CARRIER               14
#define ERR_NONNUM_CARRIER              15
#define ERR_BLANK_THRESHOLD             16
#define ERR_NONNUM_THRESHOLD    17
#define ERR_NONNUM_OL_S_WID             18
#define ERR_NONNUM_OL_I_ID              19
#define ERR_NONNUM_OL_QUAN              20
#define ERR_INCOMPLETE_OL               21
#define ERR_CUST_NAME_ID                22
#define ERR_AMT_TOO_LARGE               23

char* err_text[] =
                {
                "Success",
                "Generic error",
                "Blank warehouse id field",
                "Non-numeric or negative input in warehouse id field",
                "Blank district id field",
                "Non-numeric or negative input in district id field",
                "Blank customer id field",
                "Non-numeric or negative input in customer id field",
                "Blank customer warehouse id field",
                "Non-numeric or negative input in customer warehouse id field",
                "Blank customer district id field",
                "Non-numeric or negative input in customer district id field",
                "Blank amount field",
                "Non-numeric or negative input in amount field",
                "Blank carrier field",
                "Non-numeric or negative input in carrier field",
                "Blank threshold field",
                "Non-numeric or negative input in threshold field",
                "Non-numeric or negative input in supplying warehouse id field",
                "Non-numeric or negative input in item id field",
                "Non-numeric or negative input in quantity field",
                "Incomplete order line",
                "Either customer name or id must be entered",
                "Amount must be 9999.99 or less"
                };


#endif // _TPCC_H_ not defined
//
//---------------------------------End of tpcc.h---------------------------------------
```

trans_dell.h

```
//--------------------trans.h: Dell TPC-C Client / Web Server--------------------------
˜
//
˜
// Copyright (c) 1997 Dell Computer Corporation, All Rights Reserved
//
// Author: Dave Jaffe                                   Last
modified: 9/24/97
//
// Audited: Richard Gimarc  Performance Metrics Inc.  9/24/97
//
// transaction header file for tpcc.dll MS ISAPI DLL for TPC-C Benchmark

#ifndef _TRANS_H_
#define _TRANS_H_

// String length constants
#define ITEM_NAME_LEN                           24
#define ADDRESS_LEN
                20
#define STATE_LEN
                2
#define ZIP_LEN
                        9
#define FIRST_NAME_LEN                          16
#define MIDDLE_NAME_LEN
                2
#define LAST_NAME_LEN                           16
#define PHONE_LEN
                16
#define CREDIT_LEN
                2
#define CUST_DATA_LEN                           200
#define MAX_OL_NEW_ORDER_ITEMS          15
#define MAX_OL_ORDER_STATUS_ITEMS       15
#define DATETIME_LEN                            19
#define DATE_LEN
                10

// transaction structures

typedef struct
                {
                short
                ol_supply_w_id;
                short
                ol_quantity;
                short
                ol_stock;
                long                                    ol_i_id;
                double
                ol_i_price;
                double
                ol_amount;
                char
                ol_i_name[ITEM_NAME_LEN+1];
                char
                ol_brand_generic;
                } OL_NEW_ORDER_DATA;

typedef struct
                {
                short
                ol_supply_w_id;
                short
                ol_quantity;
                long                                    ol_i_id;
                double
                ol_amount;
                char
                ol_delivery_d[DATE_LEN+1];
                } OL_ORDER_STATUS_DATA;

typedef struct
                {
                short                                   w_id;
                short                                   d_id;
                short
                o_ol_cnt;
                short
                o_commit_flag;
                short
                o_all_local;
                short
                num_deadlocks;
                long                                    c_id;
                long                                    o_id;
                double
                c_discount;
                double                                  w_tax;
                double                                  d_tax;
                double
                total_amount;
                char
                c_last[LAST_NAME_LEN+1];
                char
                c_credit[CREDIT_LEN+1];
                char
                o_entry_d[DATETIME_LEN+1];
                OL_NEW_ORDER_DATA
                Ol[MAX_OL_NEW_ORDER_ITEMS];
                } NEW_ORDER_DATA;

typedef struct
                {
                short                                   w_id;
                short                                   d_id;
                short                                   c_d_id;
                short
                c_w_id;
                short
                num_deadlocks;
                long                                    c_id;
                double
                h_amount;
                double
                c_credit_lim;
                double
                c_discount;
                double
                c_balance;
                char
                w_street_1[ADDRESS_LEN+1];
                char
                w_street_2[ADDRESS_LEN+1];
```

```
        char
        w_city[ADDRESS_LEN+1];
        char
        w_state[STATE_LEN+1];
        char
        w_zip[ZIP_LEN+1];
        char
        d_street_1[ADDRESS_LEN+1];
        char
        d_street_2[ADDRESS_LEN+1];
        char
        d_city[ADDRESS_LEN+1];
        char
        d_state[STATE_LEN+1];
        char
        d_zip[ZIP_LEN+1];
        char
        c_first[FIRST_NAME_LEN+1];
        char
        c_middle[MIDDLE_NAME_LEN + 1];
        char
        c_last[LAST_NAME_LEN+1];
        char
        c_street_1[ADDRESS_LEN+1];
        char
        c_street_2[ADDRESS_LEN+1];
        char
        c_city[ADDRESS_LEN+1];
        char
        c_state[STATE_LEN+1];
        char
        c_zip[ZIP_LEN+1];
        char
        c_phone[PHONE_LEN+1];
        char
        c_credit[CREDIT_LEN+1];
        char
        c_data[CUST_DATA_LEN+1];
        char
        h_date[DATETIME_LEN+1];
        char
        c_since[DATE_LEN+1];
        } PAYMENT_DATA;

typedef struct
        {
        short                                   w_id;
        short                                   d_id;
        short
        o_carrier_id;
        short
        o_ol_cnt;
        short
        num_deadlocks;
        long                                    c_id;
        long                                    o_id;
        double
        c_balance;
        char
        c_first[FIRST_NAME_LEN+1];
        char
        c_middle[MIDDLE_NAME_LEN+1];
        char
        c_last[LAST_NAME_LEN+1];
        char
        o_entry_d[DATETIME_LEN+1];
        OL_ORDER_STATUS_DATA
        OlOrderStatusData[MAX_OL_ORDER_STATUS_ITEMS];
        } ORDER_STATUS_DATA;

typedef struct
        {
        short                                   w_id;
        short
        o_carrier_id;
        int
        queued_time;
        long
        o_id[10];
        } DELIVERY_DATA;

typedef struct
        {
        short                                   w_id;
        short                                   d_id;
        short
        threshold;
        short
        num_deadlocks;
        long
        low_stock;
        } STOCK_LEVEL_DATA;

#endif // _TRANS_H_ not defined
//
//--------------------------------End of trans.h--------------------------------------
```

html_dell.h

```
//---------------------html.h: Dell TPC-C Client / Web Server-------------------------
˜
//
˜
// Copyright (c) 1997 Dell Computer Corporation, All Rights Reserved
//
// Author: Dave Jaffe                                                   Last
modified: 9/4/97
//
// Audited: Richard Gimarc  Performance Metrics Inc. 9/24/97
//
// html header file for tpcc.dll MS ISAPI DLL for TPC-C Benchmark

char* ButtonBar =
        "<CENTER><TABLE><TR>\n"
        "<TD>\n"
        "<FORM METHOD=GET ACTION=\"%s\">\n"
        "<INPUT TYPE=HIDDEN NAME=CMD
VALUE=NewOrderFormRequest>\n"
        "<INPUT TYPE=HIDDEN NAME=USER VALUE=%d>\n"
        "<INPUT TYPE=SUBMIT VALUE=\"(1) New Order\">\n"
        "</FORM>\n"
        "<TD>\n"
        "<FORM METHOD=GET ACTION=\"%s\">\n"
        "<INPUT TYPE=HIDDEN NAME=CMD
VALUE=PaymentFormRequest>\n"
        "<INPUT TYPE=HIDDEN NAME=USER VALUE=%d>\n"
        "<INPUT TYPE=SUBMIT VALUE=\"(2) Payment\">\n"
        "</FORM>\n"
        "<TD>\n"
        "<FORM METHOD=GET ACTION=\"%s\">\n"
        "<INPUT TYPE=HIDDEN NAME=CMD
VALUE=OrderStatusFormRequest>\n"
        "<INPUT TYPE=HIDDEN NAME=USER VALUE=%d>\n"
        "<INPUT TYPE=SUBMIT VALUE=\"(3) Order Status\">\n"
        "</FORM>\n"
        "<TD>\n"
        "<FORM METHOD=GET ACTION=\"%s\">\n"
        "<INPUT TYPE=HIDDEN NAME=CMD
VALUE=DeliveryFormRequest>\n"
        "<INPUT TYPE=HIDDEN NAME=USER VALUE=%d>\n"
        "<INPUT TYPE=SUBMIT VALUE=\"(4) Delivery\">\n"
        "</FORM>\n"
        "<TD>\n"
        "<FORM METHOD=GET ACTION=\"%s\">\n"
        "<INPUT TYPE=HIDDEN NAME=CMD
VALUE=StockLevelFormRequest>\n"
        "<INPUT TYPE=HIDDEN NAME=USER VALUE=%d>\n"
        "<INPUT TYPE=SUBMIT VALUE=\"(5) Stock\">\n"
        "</FORM>\n"
        "<TD>\n"
        "<FORM METHOD=GET ACTION=\"%s\">\n"
        "<INPUT TYPE=HIDDEN NAME=CMD VALUE=Logout>\n"
        "<INPUT TYPE=HIDDEN NAME=USER VALUE=%d>\n"
        "<INPUT TYPE=SUBMIT VALUE=\"(9) Exit\">\n"
        "</FORM>\n"
        "</TR></TABLE></CENTER>\r\n";

char* ResponseHTML =
        "<HTML>\n"
        "<HEAD><TITLE>%s</TITLE></HEAD>\n"
        "<BODY>\n"
        "%s"
        "<PRE>\n"
        "%s"
        "</PRE>\n"
        "</BODY>\n"
        "</HTML>\n";

char* NewOrderForm =
        "<HTML>\n"
        "<HEAD><TITLE>New Order Form</TITLE></HEAD>\n"
        "<BODY>\n"
        "<FORM METHOD=GET ACTION=\"%s\">\n"
        "<INPUT TYPE=HIDDEN NAME=CMD
VALUE=NewOrder>\n"
        "<INPUT TYPE=HIDDEN NAME=USER VALUE=%d>\n"
        "<PRE>\n"
        "                              New Order\n"
        "Warehouse: %04d"
```

```
"   District:  <INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=D_ID>"
"                    Date:\n"
"Customer:  <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=C_ID>"
"   Name:                Credit:     %%Disc:\n"
"Order Number:          Number of Lines:          W_tax:
D_tax:\n\n"
" Supp_W  Item_Id  Item Name              Qty  Stock  B/G  Price
Amount\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W00>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I00>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q00>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W01>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I01>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q01>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W02>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I02>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q02>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W03>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I03>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q03>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W04>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I04>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q04>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W05>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I05>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q05>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W06>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I06>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q06>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W07>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I07>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q07>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W08>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I08>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q08>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W09>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I09>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q09>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W10>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I10>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q10>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W11>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I11>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q11>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W12>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I12>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q12>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W13>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I13>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q13>\n"
" <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W14>   "
"<INPUT TYPE=TEXT SIZE=6 MAXLENGTH=6 NAME=I14>
"
"<INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=Q14>\n"
"</PRE>\n"
"<INPUT TYPE=SUBMIT VALUE=Submit>\n"
"<INPUT TYPE=RESET VALUE=Reset>\n"
"</FORM>\n"
"</BODY>\n"
"</HTML>\n";

char* PaymentForm =
"<HTML>\n"
"<HEAD><TITLE>Payment Form</TITLE></HEAD>\n"
"<BODY>\n"
"<FORM METHOD=GET ACTION=\"%s\">\n"
"<INPUT TYPE=HIDDEN NAME=CMD VALUE=Payment>\n"
"<INPUT TYPE=HIDDEN NAME=USER VALUE=%d>\n"
"<PRE>\n"
"                              Payment\n"
"Warehouse: %04d                    "
"District: <INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=D_ID>\n\n\n\n\n"
"Customer: <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=C_ID>"
"  Cust-Warehouse: <INPUT TYPE=TEXT SIZE=4
MAXLENGTH=4 NAME=C_W_ID>"
"  Cust-District: <INPUT TYPE=TEXT SIZE=2
MAXLENGTH=2 NAME=C_D_ID>\n"
"Name:   <INPUT TYPE=TEXT SIZE=17 MAXLENGTH=16
NAME=C_LAST>"
"                        Since:\n"
"                                    Credit:\n"
"                                    %%Disc:\n"
"                                    Phone:\n"
"Amount Paid:          $<INPUT TYPE=TEXT SIZE=7
MAXLENGTH=7 NAME=AMOUNT>"
"      New Cust-Balance:\n"
"Credit Limit:\n"
"Cust-Data:\n"
"</PRE>\n"
"<INPUT TYPE=SUBMIT VALUE=Submit>\n"
"<INPUT TYPE=RESET VALUE=Reset>\n"
"</FORM>\n"
"</BODY>\n"
"</HTML>\n";

char* OrderStatusForm =
"<HTML>\n"
"<HEAD><TITLE>Order Status Form</TITLE></HEAD>\n"
"<BODY>\n"
"<FORM METHOD=GET ACTION=\"%s\">\n"
"<INPUT TYPE=HIDDEN NAME=CMD
VALUE=OrderStatus>\n"
"<INPUT TYPE=HIDDEN NAME=USER VALUE=%d>\n"
"<PRE>\n"
"                              Order-Status\n"
"Warehouse: %04d"
"   District:  <INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=D_ID>\n"
"Customer:  <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=C_ID>   "
"Name:  <INPUT TYPE=TEXT SIZE=17 MAXLENGTH=16
NAME=C_LAST>\n"
"Cust-Balance:\n\n"
"Order-Number:          Entry-Date:                Carrier-
Number:\n"
"Supply-W    Item-Id   Qty    Amount     Delivery-Date\n"
"</PRE>"
"<INPUT TYPE=SUBMIT VALUE=Submit>\n"
"<INPUT TYPE=RESET VALUE=Reset>\n"
"</FORM>\n"
"</BODY>\n"
"</HTML>\n";

char* DeliveryForm =
"<HTML>\n"
"<HEAD><TITLE>Delivery Form</TITLE></HEAD>\n"
"<BODY>\n"
"<FORM METHOD=GET ACTION=\"%s\">\n"
"<INPUT TYPE=HIDDEN NAME=CMD VALUE=Delivery>\n"
```

```
                "<INPUT TYPE=HIDDEN NAME=USER VALUE=%d>\n"
                "<PRE>\n"
                "                              Delivery\n"
                "Warehouse: %04d\n\n"
                "Carrier Number:  <INPUT TYPE='TEXT SIZE=2
MAXLENGTH=2 NAME=CARRIER>\n\n"
                "Execution Status:\n"
                "</PRE>"
                "<INPUT TYPE=SUBMIT VALUE=Submit>\n"
                "<INPUT TYPE=RESET VALUE=Reset>\n"
                "</FORM>\n"
                "</BODY>\n"
                "</HTML>\n";

char* StockLevelForm =
                "<HTML>\n"
                "<HEAD><TITLE>Stock Level Form</TITLE></HEAD>\n"
                "<BODY>\n"
                "<FORM METHOD=GET ACTION=\"%s\">\n"
                "<INPUT TYPE=HIDDEN NAME=CMD
VALUE=StockLevel>\n"
                "<INPUT TYPE=HIDDEN NAME=USER VALUE=%d>\n"
                "<PRE>\n"
                "                              Stock-Level\n"
                "Warehouse: %04d   District:  %02d\n\n"
                "Stock Level Threshold:  <INPUT TYPE=TEXT SIZE=2
MAXLENGTH=2 NAME=THRESHOLD>\n\n"
                "low stock:\n"
                "</PRE>\n"
                "<INPUT TYPE=SUBMIT VALUE=Submit>\n"
                "<INPUT TYPE=RESET VALUE=Reset>\n"
                "</FORM>\n"
                "</BODY>\n"
                "</HTML>\n";

char* LoginForm =
                "<HTML>\n"
                "<HEAD><TITLE>Dell TPC-C Client
Login</TITLE></HEAD>\n"
                "<BODY>\n"
                "<H1>Dell TPC-C Client Login</H1>"
                "<FORM METHOD=GET ACTION=\"%s\">\n"
                "<INPUT TYPE=HIDDEN NAME=CMD VALUE=Login>\n"
                "<INPUT TYPE=HIDDEN NAME=USER VALUE=-1>\n"
                "Warehouse: <INPUT TYPE=TEXT SIZE=4 MAXLENGTH=4
NAME=W_ID><BR>\n"
                "District:  <INPUT TYPE=TEXT SIZE=2 MAXLENGTH=2
NAME=D_ID><BR>\n"
                "<INPUT TYPE=SUBMIT VALUE=Submit>\n"
                "<INPUT TYPE=RESET VALUE=Reset>\n"
                "</FORM>\n"
                "</BODY>\n"
                "</HTML>\n";
//
//--------------------------------End of html.h--------------------------------------
```

tpcc_info_ora.h

```
/*
~
 * $Header: tpcc_info.h 7030100.1 95/07/19 15:11:37 plai Generic<base> $ Copyr
(c) 1995 Oracle
 */
/*==========================================================
=========+
|        Copyright (c) 1995  Oracle Corp, Redwood Shores, CA      |
|                  OPEN SYSTEMS PERFORMANCE GROUP                 |
|                     All Rights Reserved                         |

+==========================================================
=========+
| FILENAME
|    tpcc_info.h
| DESCRIPTION
|    Include file for TPC-C benchmark programs.

+==========================================================
=========*/

#ifndef TPCC_INFO_H
#define TPCC_INFO_H

/* this set is duplicated in c_Defs.h, c_Defs.h is used for batch driver */
#define MENTXN    0         /* menu txn */
#define NEWTXN    1         /* new order transaction */
#define PAYTXN    2         /* payment transaction */
#define ORDTXN    3         /* order status transaction */
#define DELTXN    4         /* delivery transaction */
#define STOTXN    5         /* stock level transaction */
#define ALLTXN    6         /* for processing all txns */
#define ALLTXNNODEL 7       /* for processing all txns except delivery */
/* New order */

struct newinstruct {
  int w_id;
  int d_id;
  int c_id;
  int ol_i_id[15];
  int ol_supply_w_id[15];
  int ol_quantity[15];
};

struct newoutstruct {
  int terror;
  int o_id;
  int o_ol_cnt;
  char c_last[17];
  char c_credit[3];
  float c_discount;
  float w_tax;
  float d_tax;
  char o_entry_d[20];
  float total_amount;
  char i_name[15][25];
  int s_quantity[15];
  char brand_generic[15];
  float i_price[15];
  float ol_amount[15];
  char status[26];
  int retry;
};

struct newstruct {
  struct newinstruct newin;
  struct newoutstruct newout;
};


/* Payment */

struct payinstruct {
  int w_id;
  int d_id;
  int c_w_id;
  int c_d_id;
  int c_id;
  int bylastname;
  int h_amount;
  char c_last[17];
};

struct payoutstruct {
  int terror;
  char w_street_1[21];
  char w_street_2[21];
  char w_city[21];
  char w_state[3];
  char w_zip[10];
  char d_street_1[21];
  char d_street_2[21];
  char d_city[21];
  char d_state[3];
  char d_zip[10];
  int  c_id;
  char c_first[17];
  char c_middle[3];
  char c_last[17];
  char c_street_1[21];
  char c_street_2[21];
  char c_city[21];
  char c_state[3];
  char c_zip[10];
  char c_phone[17];
  char c_since[11];
  char c_credit[3];
  double c_credit_lim;
  float c_discount;
  double c_balance;
  char c_data[201];
  char h_date[20];
  int retry;
};

struct paystruct {
  struct payinstruct payin;
  struct payoutstruct payout;
};
```

```
/* Order status */

struct ordinstruct {
   int w_id;
   int d_id;
   int c_id;
   int bylastname;
   char c_last[17];
};

struct ordoutstruct {
   int terror;
   int c_id;
   char c_last[17];
   char c_first[17];
   char c_middle[3];
   double c_balance;
   int o_id;
   char o_entry_d[20];
   int o_carrier_id;
   int o_ol_cnt;
   int ol_supply_w_id[15];
   int ol_i_id[15];
   int ol_quantity[15];
   float ol_amount[15];
   char ol_delivery_d[15][11];
   int retry;
};

struct ordstruct {
   struct ordinstruct ordin;
   struct ordoutstruct ordout;
};


/* Delivery */

struct delinstruct {
   int w_id;
   int o_carrier_id;
   double qtime;
   int in_timing_int;
};

struct deloutstruct {
   int terror;
   int retry;
};

struct delstruct {
   struct delinstruct delin;
   struct deloutstruct delout;
};


/* Stock level */

struct stoinstruct {
   int w_id;
   int d_id;
   int threshold;
};

struct stooutstruct {
   int terror;
   int low_stock;
   int retry;
};

struct stostruct {
   struct stoinstruct stoin;
   struct stooutstruct stoout;
};

/* used these definitions in client code only */
typedef struct delstruct  DeliveryData, *pDeliveryData;
typedef struct newstruct  NewOrderData, *pNewOrderData;
typedef struct paystruct  PaymentData, *pPaymentData;
typedef struct ordstruct  OrderStatusData, *pOrderStatusData;
typedef struct stostruct  StockLevelData, *pStockLevelData;

#endif
```

tpcc_ora.h

```
/*
~
 * $Header: tpcc.h 7030100.1 95/07/19 15:10:55 plai Generic<base> $ Copyr (c)
1993 Oracle
~
 */
~
/*=========================================================
=========+
|       Copyright (c) 1995  Oracle Corp, Redwood Shores, CA     |
|                 OPEN SYSTEMS PERFORMANCE GROUP                |
|                     All Rights Reserved                       |

+=========================================================
=========+
| FILENAME
|    tpcc.h
| DESCRIPTION
|    Include file for TPC-C benchmark programs.

+=========================================================
=========*/

#ifndef TPCC_H
#define TPCC_H

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

#include <oratypes.h>
#include <oci.h>
#include <ocidfn.h>

typedef struct cda_def csrdef;
typedef struct cda_def ldadef;


/* TPC-C transaction functions */

extern int TPCinit ();
extern int TPCnew ();
extern int TPCpay ();
extern int TPCord ();
extern int TPCdel ();
extern int TPCsto ();
extern int TPCexit ();
extern int TPCtrace ();
extern int TPCdumpinit ();
extern void TPCdumpnew ();
extern void TPCdumppay ();
extern void TPCdumpord ();
extern void TPCdumpdel ();
extern void TPCdumpsto ();
extern void TPCdumpexit ();


/* Error codes */

#define RECOVERR -10
#define IRRECERR -20
#define NOERR    111
#define DEL_ERROR -666
#define DEL_DATE_LEN 7
#define NDISTS 10
#define NITEMS 15
#define SQL_BUF_SIZE 8192

/* NULL value definitions */
#define NULL_DATE   "01-01-1899"
#define NULL_CARRIER_ID 11
#endif
```

dpbcore_ora.h

```
/* Copyright (c) Oracle Corporation 1993, 1992.  All Rights Reserved. */
~

~
/*
~
NAME    DPBCORE.H

DESCRIPTION
  Header for CORE function
```

```
NOTES
  Desktop Performance Group

MODIFIED      (MM/DD/YY)
  B Moriarty   06/02/95 -  add dpbetime() for accurate elapsed time measure
  B Moriarty   05/26/95 -  add dpboradt() for new reporting
  B Moriarty   05/10/95 -  add dpbcpu() for tpcc
  C Kelly      04/21/94 -  add dpbinpgm() and dpbxtpgm() for Netware NLMs
  C Kelly      02/24/93 -  add dpbfsync()
  B Moriarty   11/12/93 -  add dpbgetprty()
  R Keller     10/18/93 -  add dpbprty()
  R Keller     03/06/92 -  initial version

*/

#ifndef __dpbcore__
# define __dpbcore__


#include <stdio.h>
#include "dpbpcntl.h"

#ifdef __STDC__                         /* ANSI C                  */
int          dpbfsync(FILE *);          /* fsync for ACID          */
int          dpbgetprty(char *,char *,int);    /* get O/S priority        */
void         dpbinpgm(void);            /* pgm. init. function     */
unsigned long dpbpchk(pcntl *);         /* check on forked process  */
unsigned long dpbproc(char *[], pcntl *);       /* spawn/fork new process   */
int          dpbprty(char *);           /* set O/S priority        */
clock_t      dpbtimef(void);            /* get time                */
clock_t      dpbcpu(void);              /* get CPU time            */
void         dpbwait(clock_t);          /* wait routine in millisec */
void         dpbxtpgm(void);            /* pgm exit routine        */
int          dpboradt(char *);          /* sys date time in ora form*/
clock_t      dpbetime(void);            /* elapsed time            */
#else                                   /* K&R C                   */
int          dpbfsync();                /* fsync for ACID          */
int          dpbgetprty();              /* get O/S priority        */
void         dpbinpgm();                /* pgm. init. function     */
unsigned long dpbpchk();                /* check on forked process  */
unsigned long dpbproc();                /* spawn/fork new process   */
int          dpbprty();                 /* set O/S priority        */
clock_t      dpbtimef();                /* get time                */
clock_t      dpbcpu();                  /* get cpu time            */
void         dpbwait();                 /* wait routine in millisec */
void         dpbxtpgm();                /* pgm exit routine        */
int          dpboradt();                /* sys date time in ora form*/
clock_t      dpbetime();                /* elapsed time            */
#endif /* __STDC__ */


#endif /* __dpbcore__ */
```

dpbpcntl.h

```
/* Copyright (c) Oracle Corporation 1993, 1992.  All Rights Reserved. */

/*
NAME    DPBPCNTL.H

DESCRIPTION
  OSD structures for process control

NOTES
  Desktop Performance Group

MODIFIED      (MM/DD/YY)
  R Keller     02/03/93 -  initial version

*/

#ifndef __dpbpcntl__
# define __dpbpcntl__



#ifdef ORA_OS2                          /* IBM OS/2 2.x            */
# define INCL_DOSPROCESS
# include <os2.h>
typedef struct _pcntl
{
  RESULTCODES rcodes;
} pcntl;
#endif /* ORA_OS2 */                    /* IBM OS/2 2.x            */

#ifdef ORA_NT                           /* Microsoft Windows NT    */
# include <windows.h>                   /*                         */
typedef struct _pcntl
{
  PROCESS_INFORMATION proc_info;
} pcntl;
#endif /* ORA_NT */                     /* Microsoft Windows NT    */



#ifdef ORA_AUX                          /* Apple A/UX              */
typedef struct _pcntl
{
  int dummy;
} pcntl;
#endif /* ORA_AUX */                    /* Apple A/UX              */



#ifdef ORA_NW                           /* Novell Netware          */
typedef struct _pcntl
{
  int dummy;
} pcntl;
#endif /* ORA_NW */                     /* Novell Netware          */



#endif /* __dpbpcntl__ */
```

plnew_ora.c

```
/*=========================================================
=========+
~
|       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA       |
~
|                  OPEN SYSTEMS PERFORMANCE GROUP                 |
|                      All Rights Reserved                        |

+=========================================================
=========+
| FILENAME
|    plnew.c
| DESCRIPTION
|    OCI version (using PL/SQL stored procedure) of
|    NEW ORDER transaction in TPC-C benchmark.
|                  modified by jpj. 1-19-98 for DELL tpc-c web client

+=========================================================
=========*/

#if defined(ISO1) || defined(ISO7)
#include <windows.h>
#endif

#include "tpcc_ora.h"
#include "tpccpl_ora.h"
#ifdef _TUX
#include <userlog.h>
#endif

extern int SelItemStk ();
#ifdef OPS
extern int UpdStk ();
#else
extern int UpdStk2 ();
#endif

#ifdef OPS
#define SQLTXT2A "UPDATE stock SET s_order_cnt = s_order_cnt + 1, \
   s_ytd = s_ytd + :ol_quantity, s_remote_cnt = s_remote_cnt + :s_remote, \
   s_quantity = s_quantity - :ol_quantity + "

#define SQLTXT2B "DECODE (SIGN (s_quantity - :ol_quantity - 10), -1, 91, 0)
\
   WHERE s_i_id = :ol_i_id AND s_w_id = :ol_supply_w_id"
#else
#define SQLTXT2A "UPDATE stock SET s_order_cnt = s_order_cnt + 1, \
   s_ytd = s_ytd + :ol_quantity, s_remote_cnt = s_remote_cnt + :s_remote, \
   s_quantity = :s_quantity "

#define SQLTXT2B " WHERE rowid = :s_rowid"
#endif

#define SQLTXT3A "\
```

```
SELECT 0,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :10 AND s_w_id = :30 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3B "\
SELECT 1,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :11 AND s_w_id = :31 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3C "\
SELECT 2,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :12 AND s_w_id = :32 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3D "\
SELECT 3,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :13 AND s_w_id = :33 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3E "\
SELECT 4,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :14 AND s_w_id = :34 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3F "\
SELECT 5,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :15 AND s_w_id = :35 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3G "\
SELECT 6,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :16 AND s_w_id = :36 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3H "\
SELECT 7,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :17 AND s_w_id = :37 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3I "\
SELECT 8,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :18 AND s_w_id = :38 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3J "\
SELECT 9,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :19 AND s_w_id = :39 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3K "\
SELECT 10,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :20 AND s_w_id = :40 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3L "\
SELECT 11,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :21 AND s_w_id = :41 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3M "\
SELECT 12,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :22 AND s_w_id = :42 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3N "\
SELECT 13,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :23 AND s_w_id = :43 AND s_i_id = i_id
UNION ALL \
"

#define SQLTXT3O "\
SELECT 14,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :24 AND s_w_id = :44 AND s_i_id = i_id"


#define SQLTXT4A "INSERT INTO order_line \
  (ol_o_id, ol_d_id, ol_w_id, oL_number, ol_delivery_d, ol_i_id, \
   ol_supply_w_id, ol_quantity,ol_amount,ol_dist_info) \
"
#define SQLTXT4B "VALUES (:ol_o_id, :ol_d_id, \
  :ol_w_id, :ol_number, :null_date, :ol_i_id, :ol_supply_w_id, :ol_quantity, \
  :ol_amount, :ol_dist_info)"

#define NITEMS 15
#define ROWIDLEN 20
#define OCIROWLEN 20

struct newctx {
  sb2 nol_i_id_ind[NITEMS];
  sb2 nol_supply_w_id_ind[NITEMS];
  sb2 nol_quantity_ind[NITEMS];
  sb2 nol_amount_ind[NITEMS];
  sb2 i_name_ind[NITEMS];
  sb2 s_quantity_ind[NITEMS];
  sb2 i_price_ind[NITEMS];
  sb2 ol_w_id_ind[NITEMS];
  sb2 ol_d_id_ind[NITEMS];
  sb2 ol_o_id_ind[NITEMS];
  sb2 ol_number_ind[NITEMS];
  sb2 cons_ind[NITEMS];
  sb2 s_rowid_ind[NITEMS];
  sb2 s_remote_ind[NITEMS];
  sb2 s_quant_ind[NITEMS];
  sb2 i_data_ind[NITEMS];
  sb2 s_data_ind[NITEMS];
  sb2 s_dist_info_ind[NITEMS];
  sb2 ol_dist_info_ind[NITEMS];
  sb2 null_date_ind[NITEMS];

  ub2 nol_i_id_len[NITEMS];
  ub2 nol_supply_w_id_len[NITEMS];
  ub2 nol_quantity_len[NITEMS];
  ub2 nol_amount_len[NITEMS];
  ub2 i_name_len[NITEMS];
  ub2 s_quantity_len[NITEMS];
  ub2 i_price_len[NITEMS];
  ub2 ol_w_id_len[NITEMS];
  ub2 ol_d_id_len[NITEMS];
  ub2 ol_o_id_len[NITEMS];
  ub2 ol_number_len[NITEMS];
  ub2 cons_len[NITEMS];
  ub2 s_rowid_len[NITEMS];
  ub2 s_remote_len[NITEMS];
  ub2 s_quant_len[NITEMS];
  ub2 i_data_len[NITEMS];
  ub2 s_data_len[NITEMS];
  ub2 s_dist_info_len[NITEMS];
  ub2 ol_dist_info_len[NITEMS];
  ub2 null_date_len[NITEMS];

  ub2 nol_i_id_rcode[NITEMS];
  ub2 nol_supply_w_id_rcode[NITEMS];
  ub2 nol_quantity_rcode[NITEMS];
  ub2 nol_amount_rcode[NITEMS];
  ub2 i_name_rcode[NITEMS];
  ub2 s_quantity_rcode[NITEMS];
  ub2 i_price_rcode[NITEMS];
  ub2 ol_w_id_rcode[NITEMS];
  ub2 ol_d_id_rcode[NITEMS];
  ub2 ol_o_id_rcode[NITEMS];
  ub2 ol_number_rcode[NITEMS];
  ub2 cons_rcode[NITEMS];
  ub2 s_rowid_rcode[NITEMS];
  ub2 s_remote_rcode[NITEMS];
  ub2 s_quant_rcode[NITEMS];
  ub2 i_data_rcode[NITEMS];
  ub2 s_data_rcode[NITEMS];
  ub2 s_dist_info_rcode[NITEMS];
  ub2 ol_dist_info_rcode[NITEMS];
  ub2 null_date_rcode[NITEMS];

  int ol_w_id[NITEMS];
  int ol_d_id[NITEMS];
  int ol_o_id[NITEMS];
  int ol_number[NITEMS];
  int cons[NITEMS];

  OCIRowid  *s_rowid_ptr[NITEMS];

 int s_remote[NITEMS];
 char i_data[NITEMS][51];
 char s_data[NITEMS][51];
 char s_dist_info[NITEMS][25];
 unsigned char null_date[NITEMS][7]; /* base date for null date entry */
 OCIStmt *curn;
```

```
OCIStmt *curn1;
OCIStmt *curn2;
OCIStmt *curn3[10];
OCIStmt *curn4;
OCIBind *w_id_bp;
OCIBind *d_id_bp;
OCIBind *c_id_bp;
OCIBind *o_all_local_bp;
OCIBind *o_all_cnt_bp;
OCIBind *w_tax_bp;
OCIBind *d_tax_bp;
OCIBind *o_id_bp;
OCIBind *c_discount_bp;
OCIBind *c_credit_bp;
OCIBind *c_last_bp;
OCIBind *retries_bp;
OCIBind *cr_date_bp;
OCIBind *ol_i_id_bp;
OCIBind *ol_supply_w_id_bp;
OCIBind *s_quantity_bp;
OCIBind *s_rowid_bp;
OCIBind *ol_quantity_bp;
OCIBind *s_remote_bp;
OCIBind *id_bp[10][15];
OCIBind *sd_bp[10][15];
OCIDefine *Dcons[10];
OCIDefine *Ds_rowid[10];
OCIDefine *Di_price[10];
OCIDefine *Di_data[10];
OCIDefine *Ds_dist_info[10];
OCIDefine *Ds_data[10];
OCIDefine *Ds_quantity[10];
OCIDefine *Di_name[10];
OCIBind *ol_o_id_bp;
OCIBind *ol_d_id_bp;
OCIBind *ol_w_id_bp;
OCIBind *ol_number_bp;
OCIBind *ol_amount_bp;
OCIBind *ol_dist_info_bp;
OCIBind *null_date_bp;
sb2 w_id_ind;
ub2 w_id_len;
ub2 w_id_rc;

sb2 d_id_ind;
ub2 d_id_len;
ub2 d_id_rc;

sb2 c_id_ind;
ub2 c_id_len;
ub2 c_id_rc;

sb2 o_all_local_ind;
ub2 o_all_local_len;
ub2 o_all_local_rc;

sb2 o_ol_cnt_ind;
ub2 o_ol_cnt_len;
ub2 o_ol_cnt_rc;

sb2 w_tax_ind;
ub2 w_tax_len;
ub2 w_tax_rc;

sb2 d_tax_ind;
ub2 d_tax_len;
ub2 d_tax_rc;

sb2 o_id_ind;
ub2 o_id_len;
ub2 o_id_rc;

sb2 c_discount_ind;
ub2 c_discount_len;
ub2 c_discount_rc;

sb2 c_credit_ind;
ub2 c_credit_len;
ub2 c_credit_rc;

sb2 c_last_ind;
ub2 c_last_len;
ub2 c_last_rc;

sb2 retries_ind;
ub2 retries_len;
ub2 retries_rc;

sb2 cr_date_ind;
ub2 cr_date_len;
ub2 cr_date_rc;


int cs;
int norow;
};

typedef struct newctx newctx;

newctx *nctx;

plnewinit ()

{

  int i, j;
  text stmbuf[SQL_BUF_SIZE];
  text formatbuf[SQL_BUF_SIZE];
  char id[4];
  char sd[4];

  nctx = (newctx *) malloc (sizeof(newctx));
  memset(nctx,(char)0,sizeof(newctx));
  nctx->cs = 1;
  nctx->norow=0;
  for(i=0;i<NITEMS;i++) {
    OCIERROR(errhp, OCIDescriptorAlloc(tpcenv,(dvoid**)&nctx-
>s_rowid_ptr[i],
       OCI_DTYPE_ROWID,0,(dvoid**)0));
  }
  nctx->w_id_ind = TRUE;
  nctx->w_id_len = sizeof(w_id);
  nctx->d_id_ind = TRUE;
  nctx->d_id_len = sizeof(d_id);
  nctx->c_id_ind = TRUE;
  nctx->c_id_len = sizeof(c_id);
  nctx->o_all_local_ind = TRUE;
  nctx->o_all_local_len = sizeof(o_all_local);
  nctx->o_ol_cnt_ind = TRUE;
  nctx->o_ol_cnt_len = sizeof(o_ol_cnt);
  nctx->w_tax_ind = TRUE;
  nctx->w_tax_len = 0;
  nctx->d_tax_ind = TRUE;
  nctx->d_tax_len = 0;
  nctx->o_id_ind = TRUE;
  nctx->o_id_len = sizeof(o_id);
  nctx->c_discount_ind = TRUE;
  nctx->c_discount_len = 0;
  nctx->c_credit_ind = TRUE;
  nctx->c_credit_len = 0;
  nctx->c_last_ind = TRUE;
  nctx->c_last_len = 0;
  nctx->retries_ind = TRUE;
  nctx->retries_len = sizeof(retries);
  nctx->cr_date_ind = TRUE;
  nctx->cr_date_len = sizeof(cr_date);

   /* open first cursor */
  OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&nctx->curn1),
                 OCI_HTYPE_STMT, 0, (dvoid**)0));
  if(sqlfile("new.sql", stmbuf))
    return(1);
  OCIERROR(errhp,OCIStmtPrepare(nctx->curn1, errhp, stmbuf, strlen((char
*)stmbuf),

                 OCI_NTV_SYNTAX, OCI_DEFAULT));

   /* bind variables */
  OCIBNDR(nctx->curn1, nctx->w_id_bp, errhp, ":w_id",ADR(w_id),SIZ(w_id),
                 SQLT_INT, &nctx->w_id_ind, &nctx->w_id_len, &nctx-
>w_id_rc);
  OCIBNDR(nctx->curn1, nctx->d_id_bp, errhp, ":d_id",ADR(d_id),SIZ(d_id),
                 SQLT_INT, &nctx->d_id_ind, &nctx->d_id_len, &nctx-
>d_id_rc);
  OCIBNDR(nctx->curn1, nctx->c_id_bp, errhp, ":c_id",ADR(c_id),SIZ(c_id),
                 SQLT_INT, &nctx->c_id_ind, &nctx->c_id_len, &nctx-
>c_id_rc);
  OCIBNDR(nctx->curn1, nctx->o_all_local_bp, errhp, ":o_all_local",
                 ADR(o_all_local), SIZ(o_all_local),SQLT_INT, &nctx-
>o_all_local_ind,
                 &nctx->o_all_local_len, &nctx->o_all_local_rc);
  OCIBNDR(nctx->curn1, nctx->o_all_cnt_bp, errhp, ":o_ol_cnt",ADR(o_ol_cnt),
                 SIZ(o_ol_cnt),SQLT_INT,
                 &nctx->o_ol_cnt_ind, &nctx->o_ol_cnt_len, &nctx-
>o_ol_cnt_rc);
  OCIBNDR(nctx->curn1, nctx->w_tax_bp, errhp,
":w_tax",ADR(w_tax),SIZ(w_tax),
```

```
                SQLT_INT, &nctx->w_tax_ind, &nctx->w_tax_len, &nctx-
>w_tax_rc);
  OCIBNDR(nctx->curn1, nctx->d_tax_bp, errhp,
":d_tax",ADR(d_tax),SIZ(d_tax),
                SQLT_INT, &nctx->d_tax_ind, &nctx->d_tax_len, &nctx-
>d_tax_rc);
  OCIBNDR(nctx->curn1, nctx->o_id_bp, errhp, ":o_id",ADR(o_id),SIZ(o_id),
                SQLT_INT, &nctx->o_id_ind, &nctx->o_id_len, &nctx-
>o_id_rc);
  OCIBNDR(nctx->curn1, nctx->c_discount_bp, errhp, ":c_discount",
                ADR(c_discount), SIZ(c_discount),SQLT_INT,
                &nctx->c_discount_ind, &nctx->c_discount_len, &nctx-
>c_discount_rc);
  OCIBNDR(nctx->curn1, nctx->c_credit_bp, errhp, ":c_credit",c_credit,
                SIZ(c_credit),SQLT_CHR,
                &nctx->c_credit_ind, &nctx->c_credit_len, &nctx->c_credit_rc);
  OCIBNDR(nctx->curn1, nctx->c_last_bp, errhp, ":c_last",c_last,SIZ(c_last),
                SQLT_STR, &nctx->c_last_ind, &nctx->c_last_len, &nctx-
>c_last_rc);
  OCIBNDR(nctx->curn1, nctx->retries_bp, errhp, ":retry",ADR(retries),
                SIZ(retries),SQLT_INT,
                &nctx->retries_ind, &nctx->retries_len, &nctx->retries_rc);
  OCIBNDR(nctx->curn1, nctx->cr_date_bp, errhp,
":cr_date",cr_date,SIZ(cr_date),        SQLT_DAT, &nctx->cr_date_ind, &nctx-
>cr_date_len, &nctx->cr_date_rc);


   /* open second cursor */

  OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)(&nctx->curn2),
OCI_HTYPE_STMT,
                                                0, (dvoid**)0));
  sprintf ((char *) stmbuf, "%s%s", SQLTXT2A, SQLTXT2B);
  OCIERROR(errhp,OCIStmtPrepare(nctx->curn2, errhp, stmbuf,
        strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));


   /* bind variables */
#ifdef OPS
  OCIBNDRA(nctx->curn2, nctx->ol_i_id_bp,errhp,":ol_i_id",nol_i_id,
                SIZ(int), SQLT_INT, nctx->nol_i_id_ind,nctx->nol_i_id_len,
                nctx->nol_i_id_rcode);
  OCIBNDRA(nctx->curn2, nctx->ol_supply_w_id_bp, errhp, ":ol_supply_w_id",
                nol_supply_w_id,SIZ(int),SQLT_INT, nctx-
>nol_supply_w_id_ind,
                nctx->nol_supply_w_id_len, nctx->nol_supply_w_id_rcode);
#else
  OCIBNDRA(nctx->curn2, nctx->s_quantity_bp,errhp,":s_quantity",s_quantity,
                SIZ(int), SQLT_INT,nctx->s_quant_ind,nctx->s_quant_len,
                nctx->s_quant_rcode);
  OCIBNDRA(nctx->curn2, nctx->s_rowid_bp, errhp, ":s_rowid",nctx-
>s_rowid_ptr,
                sizeof(nctx->s_rowid_ptr[0]),SQLT_RDD,nctx->s_rowid_ind,
                nctx->s_rowid_len,nctx->s_rowid_rcode);
#endif
  OCIBNDRA(nctx->curn2, nctx-
>ol_quantity_bp,errhp,":ol_quantity",nol_quantity,
                SIZ(int),SQLT_INT,nctx->nol_quantity_ind,nctx-
>nol_quantity_len,
                nctx->nol_quantity_rcode);
  OCIBNDRA(nctx->curn2, nctx->s_remote_bp, errhp, ":s_remote",nctx-
>s_remote,
                SIZ(int), SQLT_INT,nctx->s_remote_ind,nctx->s_remote_len,
                nctx->s_remote_rcode);

   /* open third cursor and bind variables */

  for (i = 0; i < 10; i++)
  {
    j = i + 1;
    OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)&((nctx->curn3)[i]),

OCI_HTYPE_STMT, 0, (dvoid**)0));
    sprintf (formatbuf, "%s%s%s%s%s%s%s%s%s%s%s%s%s%s",

                SQLTXT3A,

                SQLTXT3B,

                SQLTXT3C,

                SQLTXT3D,

                SQLTXT3E,

                SQLTXT3F,

                SQLTXT3G,

                SQLTXT3H,

                SQLTXT3I,

                SQLTXT3J,

                SQLTXT3K,

                SQLTXT3L,

                SQLTXT3M,

                SQLTXT3N,

                SQLTXT3O
    );
    sprintf ((char *) stmbuf, formatbuf, j, j, j, j, j, j, j, j, j, j, j, j,
                j, j, j);

    OCIERROR(errhp,OCIStmtPrepare((nctx->curn3)[i], errhp, stmbuf,
                                                strlen((char
*)stmbuf),OCI_NTV_SYNTAX,
                                                OCI_DEFAULT));
    OCIERROR(errhp,
        OCIAttrSet(nctx->curn3[i],OCI_HTYPE_STMT,(dvoid*)&nctx-
>norow,0,
            OCI_ATTR_PREFETCH_ROWS,errhp));
      for (j = 0; j < NITEMS; j++)
      {
                sprintf (id, ":%d", j + 10);
                sprintf (sd, ":%d", j + 30);
      OCIBNDRA((nctx->curn3)[i],(nctx->id_bp)[i][j],errhp,id,ADR(nol_i_id[j]),
                  SIZ(int),SQLT_INT,
                  &nctx->nol_i_id_ind[j],&nctx->nol_i_id_len[j],
                  &nctx->nol_i_id_rcode[j]);
      OCIBNDRA((nctx->curn3)[i],(nctx->sd_bp)[i][j],errhp,sd,
                  ADR(nol_supply_w_id[j]),SIZ(int),SQLT_INT,
                  &nctx->nol_supply_w_id_ind[j],&nctx-
>nol_supply_w_id_len[j],
                  &nctx->nol_supply_w_id_rcode[j]);
                nctx->nol_i_id_ind[j] = NA;
                nctx->nol_supply_w_id_ind[j] = NA;
                nctx->nol_i_id_len[j] = sizeof(int);
                nctx->nol_supply_w_id_len[j] = sizeof(int);
      }

    OCIDFNRA((nctx->curn3)[i],(nctx->Dcons)[i],errhp,1,&(nctx->cons[0]),
                SIZ(nctx->cons[0]),SQLT_INT,
                nctx->cons_ind,nctx->cons_len, nctx->cons_rcode);
    OCIDFNRA((nctx->curn3)[i], (nctx->Ds_rowid)[i],errhp,2,
          nctx->s_rowid_ptr,
                sizeof(nctx->s_rowid_ptr[0]),
                SQLT_RDD,nctx->s_rowid_ind,nctx->s_rowid_len,
                nctx->s_rowid_rcode);
    OCIDFNRA((nctx->curn3)[i], (nctx->Di_price)[i],errhp,3,i_price,SIZ(int),
                SQLT_INT, nctx->i_price_ind,nctx->i_price_len,nctx-
>i_price_rcode);

    OCIDFNRA((nctx->curn3)[i], (nctx->Di_name)[i],errhp,4,i_name,
                SIZ(i_name[0]),SQLT_STR, nctx->i_name_ind,nctx-
>i_name_len,
                nctx->i_name_rcode);
    OCIDFNRA((nctx->curn3)[i], (nctx->Di_data)[i],errhp,5,nctx->i_data,
                SIZ(nctx->i_data[0]),
                SQLT_STR,nctx->i_data_ind,nctx->i_data_len,nctx-
>i_data_rcode);
    OCIDFNRA((nctx->curn3)[i], (nctx->Ds_dist_info)[i],errhp,6,
                nctx->s_dist_info, SIZ(nctx->s_dist_info[0]),SQLT_STR,
                nctx->s_dist_info_ind, nctx->s_dist_info_len,
                nctx->s_dist_info_rcode);
    OCIDFNRA((nctx->curn3)[i],(nctx->Ds_data)[i],errhp,7,nctx->s_data,
                SIZ(nctx->s_data[0]),SQLT_STR,nctx->s_data_ind,
                nctx->s_data_len,nctx->s_data_rcode);
    OCIDFNRA((nctx->curn3)[i],(nctx->Ds_quantity)[i],errhp,8,s_quantity,
                SIZ(int),SQLT_INT, nctx->s_quantity_ind,nctx-
>s_quantity_len,
                nctx->s_quantity_rcode);
  }

   /* open fourth cursor */

  OCIHandleAlloc(tpcenv, (dvoid **)(&nctx->curn4), OCI_HTYPE_STMT, 0,
                        (dvoid**)0);
  sprintf ((char *) stmbuf, "%s%s", SQLTXT4A, SQLTXT4B);
  OCIStmtPrepare(nctx->curn4, errhp, stmbuf, strlen((char *)stmbuf),
                        OCI_NTV_SYNTAX, OCI_DEFAULT);

   /* bind variables */
```

```
  OCIBNDRA(nctx->curn4, nctx->ol_o_id_bp,errhp,":ol_o_id",nctx->ol_o_id,
                SIZ(int),SQLT_INT, nctx->ol_o_id_ind,nctx->ol_o_id_len,
                nctx->ol_o_id_rcode);
  OCIBNDRA(nctx->curn4, nctx->ol_d_id_bp,errhp,":ol_d_id",nctx->ol_d_id,
                SIZ(int),SQLT_INT, nctx->ol_d_id_ind,nctx->ol_d_id_len,
                nctx->ol_d_id_rcode);
  OCIBNDRA(nctx->curn4, nctx->ol_w_id_bp,errhp,":ol_w_id",nctx->ol_w_id,
                SIZ(int),SQLT_INT, nctx->ol_w_id_ind,nctx->ol_w_id_len,
                nctx->ol_w_id_rcode);
  OCIBNDRA(nctx->curn4, nctx->ol_number_bp,errhp,":ol_number",nctx-
>ol_number,
                SIZ(int),SQLT_INT, nctx->ol_number_ind,nctx-
>ol_number_len,
                nctx->ol_number_rcode);
  OCIBNDRA(nctx->curn4, nctx->ol_i_id_bp,errhp,":ol_i_id",nol_i_id,SIZ(int),
                SQLT_INT, nctx->nol_i_id_ind,nctx->nol_i_id_len,
                nctx->nol_i_id_rcode);
  OCIBNDRA(nctx->curn4, nctx->ol_supply_w_id_bp,errhp,":ol_supply_w_id",
                nol_supply_w_id,SIZ(int),SQLT_INT, nctx-
>nol_supply_w_id_ind,
                nctx->nol_supply_w_id_len, nctx->nol_supply_w_id_rcode);
  OCIBNDRA(nctx->curn4, nctx-
>ol_quantity_bp,errhp,":ol_quantity",nol_quantity,
                SIZ(int),SQLT_INT, nctx->nol_quantity_ind,nctx-
>nol_quantity_len,
                nctx->nol_quantity_rcode);
  OCIBNDRA(nctx->curn4, nctx->ol_amount_bp,errhp,":ol_amount",nol_amount,
                SIZ(int),SQLT_INT, nctx->nol_amount_ind,nctx-
>nol_amount_len,
                nctx->nol_amount_rcode);
  OCIBNDRA(nctx->curn4, nctx->ol_dist_info_bp,errhp,":ol_dist_info",
                nctx->s_dist_info, SIZ(nctx->s_dist_info[0]),SQLT_AFC,
                nctx->ol_dist_info_ind, nctx->ol_dist_info_len,
                nctx->ol_dist_info_rcode);
  OCIBNDRA(nctx->curn4, nctx->null_date_bp,errhp,":null_date",nctx-
>null_date,
                SIZ(nctx->null_date[0]), SQLT_DAT,nctx->null_date_ind,
                nctx->null_date_len, nctx->null_date_rcode);

  /* set up the null date Null date is 15-sep-11 */
 for (i=0; i < NITEMS; i++)
 {
   nctx->null_date[i][0] = 118;
     nctx->null_date[i][1] = 111;
   nctx->null_date[i][2] = 1;
   nctx->null_date[i][3] = 1;
   nctx->null_date[i][4] = 1;
   nctx->null_date[i][5] = 1;
   nctx->null_date[i][6] = 1;
 }

   return (0);

}



plnew ()
{

   int i, j, k;
   int rpc, rpc3, rowoff, iters,rcount;
   ub4 flags;

#if defined(ISO1) || defined(ISO7)
   int reread;
   char sdate[30];

#ifdef __STDC__
#  define PROTO(args)    args
#else
#  define PROTO(args)    ()
#endif
void sysdate PROTO((char *));
   sysdate (sdate);
   printf ("New Order started at: %s\n", sdate);
#endif

retry:

#ifdef ISO7
   reread = 1;
#endif

   status = 0;                  /* number of invalid items */

   /* get number of order lines, and check if all are local */
   o_ol_cnt = NITEMS;
   o_all_local = 1;
   for (i = 0; i < NITEMS; i++) {
     if (nol_i_id[i] == 0) {
                  o_ol_cnt = i;
                  break;
     }
     if (nol_supply_w_id[i] != w_id) {
                  nctx->s_remote[i] = 1;
                  o_all_local = 0;
     }
     else
                  nctx->s_remote[i] = 0;
   }
   nctx->w_id_ind = TRUE;
   nctx->w_id_len = sizeof(w_id);
   nctx->d_id_ind = TRUE;
   nctx->d_id_len = sizeof(d_id);
   nctx->c_id_ind = TRUE;
   nctx->c_id_len = sizeof(c_id);
   nctx->o_all_local_ind = TRUE;
   nctx->o_all_local_len = sizeof(o_all_local);
   nctx->o_ol_cnt_ind = TRUE;
   nctx->o_ol_cnt_len = sizeof(o_ol_cnt);
   nctx->w_tax_ind = TRUE;
   nctx->w_tax_len = 0;
   nctx->d_tax_ind = TRUE;
   nctx->d_tax_len = 0;
   nctx->o_id_ind = TRUE;
   nctx->o_id_len = sizeof(o_id);
   nctx->c_discount_ind = TRUE;
   nctx->c_discount_len = 0;
   nctx->c_credit_ind = TRUE;
   nctx->c_credit_len = 0;
   nctx->c_last_ind = TRUE;
   nctx->c_last_len = 0;
   nctx->retries_ind = TRUE;
   nctx->retries_len = sizeof(retries);
   nctx->cr_date_ind = TRUE;
   nctx->cr_date_len = sizeof(cr_date);

   execstatus = OCIStmtExecute(tpcsvc,nctx-
>curn1,errhp,1,0,0,0,OCI_DEFAULT);
   if(execstatus != OCI_SUCCESS){
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     errcode = OCIERROR(errhp,execstatus);
     if(errcode == NOT_SERIALIZABLE) {
                  retries++;
                  goto retry;
     } else if (errcode == RECOVERR) {
                  retries++;
                  goto retry;
     } else {
                  return -1;
     }
   }
   /* initialization for array operations */

   for (i = 0; i < o_ol_cnt; i++) {
     nctx->ol_w_id[i] = w_id;
     nctx->ol_d_id[i] = d_id;
     nctx->ol_number[i] = i + 1;
     nctx->null_date_ind[i]= TRUE;
     nctx->nol_i_id_ind[i] = TRUE;
     nctx->nol_supply_w_id_ind[i] = TRUE;
     nctx->nol_quantity_ind[i] = TRUE;
     nctx->nol_amount_ind[i] = TRUE;
     nctx->ol_w_id_ind[i] = TRUE;
     nctx->ol_d_id_ind[i] = TRUE;
     nctx->ol_o_id_ind[i] = TRUE;
     nctx->ol_number_ind[i] = TRUE;
     nctx->ol_dist_info_ind[i] = TRUE;
     nctx->s_remote_ind[i] = TRUE;
     nctx->s_quant_ind[i] = TRUE;
     nctx->cons_ind[i] = TRUE;
     nctx->s_rowid_ind[i] = TRUE;

     nctx->nol_i_id_len[i] = sizeof(int);
     nctx->nol_supply_w_id_len[i] = sizeof(int);
     nctx->nol_quantity_len[i] = sizeof(int);
     nctx->nol_amount_len[i] = sizeof(int);
     nctx->ol_w_id_len[i] = sizeof(int);
     nctx->ol_d_id_len[i] = sizeof(int);
     nctx->ol_o_id_len[i] = sizeof(int);
     nctx->ol_number_len[i] = sizeof(int);
     nctx->ol_dist_info_len[i] = nctx->s_dist_info_len[i];
     nctx->null_date_len[i]=sizeof(nctx->null_date[0]);
     nctx->s_remote_len[i] = sizeof(int);
     nctx->s_quant_len[i] = sizeof(int);
```

```
    nctx->s_rowid_len[i] = sizeof(nctx->s_rowid_ptr[0]);
    nctx->cons_len[i] = sizeof(int);
  }
  for (i = o_ol_cnt; i < NITEMS; i++) {
    nctx->nol_i_id_ind[i] = NA;
    nctx->nol_supply_w_id_ind[i] = NA;
    nctx->nol_quantity_ind[i] = NA;
    nctx->nol_amount_ind[i] = NA;
    nctx->ol_w_id_ind[i] = NA;
    nctx->ol_d_id_ind[i] = NA;
    nctx->ol_o_id_ind[i] = NA;
    nctx->ol_number_ind[i] = NA;
    nctx->ol_dist_info_ind[i] = NA;
    nctx->null_date_ind[i]= NA;
    nctx->s_remote_ind[i] = NA;
    nctx->s_quant_ind[i] = NA;
    nctx->cons_ind[i] = NA;
    nctx->s_rowid_ind[i] = NA;

    nctx->nol_i_id_len[i] = 0;
    nctx->nol_supply_w_id_len[i] = 0;
    nctx->nol_quantity_len[i] = 0;
    nctx->nol_amount_len[i] = 0;
    nctx->ol_w_id_len[i] = 0;
    nctx->ol_d_id_len[i] = 0;
    nctx->ol_o_id_len[i] = 0;
    nctx->ol_number_len[i] = 0;
    nctx->ol_dist_info_len[i] = 0;
    nctx->null_date_len[i]= 0;
    nctx->s_remote_len[i] = 0;
    nctx->s_quant_len[i] = 0;
    nctx->s_rowid_len[i] = 0;
    nctx->cons_len[i] = 0;
  }

#ifdef OPS
  rpc = UpdStk ();
  if (rpc == -2)
    goto retry;
  else if (rpc == -1)
    return (-1);
#endif

#ifdef ISO7
iso7:
#endif

  rpc3 = SelItemStk ();
  if (rpc3 == -2)
    goto retry;
  else if (rpc3 == -1)
    return (-1);

#ifdef ISO7
  sysdate (sdate);
  printf ("Item table read at: %s\n", sdate);
  for (i = 0; i < o_ol_cnt; i++) {
    if (nctx->nol_i_id_ind[i] != NA)
                printf ("   i_id = %d, i_price = %d\n", nol_i_id[i], i_price[i]);
  }
  if (reread) {
    sleep (30);
    reread = 0;
    status = 0;
    goto iso7;
  }
#endif

  /* compute order line amounts, total amount and stock quantities */

  total_amount = 0.0;
  for (i = 0; i < o_ol_cnt; i++) {
    nctx->ol_o_id[i] = o_id;
    if (nctx->nol_i_id_ind[i] != NA) {
#ifndef OPS
                s_quantity[i] -= nol_quantity[i];
                if (s_quantity[i] < 10)
                  s_quantity[i] += 91;
#endif
                nol_amount[i] = (nol_quantity[i] * i_price[i]);
                total_amount += nol_amount[i];
                if (strstr (nctx->i_data[i], "ORIGINAL") &&
                    strstr (nctx->s_data[i], "ORIGINAL"))
                  brand_gen[i] = 'B';
                else
                  brand_gen[i] = 'G';
    }
  }
  total_amount *= ((float)(10000 - c_discount)/10000) * (1.0 +
((float)(d_tax)/10000) + ((float)(w_tax)/10000));
  total_amount = total_amount/100;

#ifndef OPS
  rpc = UpdStk2 ();
  if (rpc == -2)
    goto retry;
  else if (rpc == -1)
    return (-1);
#endif

  /* number of items selected != number of stock updated */

  if (rpc3 != rpc) {
#ifdef TUX
    userlog ("Error in TPC-C server %d: %d rows of item read, ",
                proc_no, rpc3);
    userlog ("                  but %d rows of stock updated\n", rpc);
#else
    fprintf (stderr, "Error in TPC-C server %d: %d rows of item read, ",
                proc_no, rpc3);
    fprintf (stderr, "                  but %d rows of stock update\n", rpc);
#endif
    /* rollback */
    OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
    return (-1);
  }

  /* array insert into order line table */
#ifdef ISO1
  flags= OCI_DEFAULT;
#else
  flags= (status ? OCI_DEFAULT :
(OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS));
#endif
  if ((o_ol_cnt - status) > 0)
  {
    execstatus = OCIStmtExecute(tpcsvc,nctx->curn4,errhp,o_ol_cnt - status,
                                                        0,0,0,flags);
    if(execstatus != OCI_SUCCESS) {
                OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
                errcode = OCIERROR(errhp,execstatus);
                if(errcode == NOT_SERIALIZABLE) {
                  retries++;
                  goto retry;
                } else if (errcode == RECOVERR) {
                  retries++;
                  goto retry;
                } else {
                  return -1;
                }
    }
    OCIAttrGet(nctx->curn4,OCI_HTYPE_STMT,&rcount, NULL,
                                OCI_ATTR_ROW_COUNT, errhp);
    if (rcount != (o_ol_cnt - status))
    {
#ifdef TUX
                userlog ("Error in TPC-C server %d: array insert failed\n",
                                proc_no);
#else
                fprintf (stderr, "Error in TPC-C server %d: array insert failed\n",
                                proc_no);
#endif
                /* rollback */
                OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
                return (-1);
    }
  }

#ifdef ISO1
  sysdate (sdate);
  printf ("Sleep before commit/rollback at: %s\n", sdate);
  sleep (30);
  sysdate (sdate);
  printf ("Wake up after sleep at: %s\n", sdate);
#endif

  /* commit if no invalid item */

  if (status) {
    OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
    fflush(stdout);
  }
#ifdef ISO1
  else {
    OCITransCommit(tpcsvc, errhp, OCI_DEFAULT);
  }
#endif
```

```
#if defined(ISO1) || defined(ISO7)
  sysdate (sdate);
  printf ("New Order completed at: %s\n", sdate);
#endif

  return (0);

}


void plnewdone ()

{

  int i;

  if (nctx)
  {
    OCIHandleFree((dvoid *)nctx->curn1,OCI_HTYPE_STMT);
    OCIHandleFree((dvoid *)nctx->curn2,OCI_HTYPE_STMT);
    for (i = 0; i < 10; i++)
      OCIHandleFree((dvoid *)(nctx->curn3)[i],OCI_HTYPE_STMT);
    OCIHandleFree((dvoid *)nctx->curn4,OCI_HTYPE_STMT);
    free (nctx);
  }
}

/* the arrays are initialized based on a successful select from */
/* stock/item. We need to shift the values in the orderline array */
/* one position up to compensate when we have an invalid item */

shiftitemstock (i, j)

int i, j;

{

  /* shift up the values for the stock table */
   nctx->s_remote[i] =  nctx->s_remote[j];

  /* shift up the order_line values */

  nctx->nol_i_id_ind[i]=nctx->nol_i_id_ind[j];
  nol_i_id[i] = nol_i_id[j];

  nctx->nol_quantity_ind[i] =  nctx->nol_quantity_ind[j];
  nol_quantity[i] = nol_quantity[j];

  nctx->nol_supply_w_id_ind [i] =  nctx->nol_supply_w_id_ind[j];
  nol_supply_w_id[i] = nol_supply_w_id[j];
  return 0;
}
swapitemstock (i, j)

int i, j;

{

  int k;
  int tempi;
  int tempf;
  char tempstr[52];
  ub2 tempub2;
  sb2 tempsb2;
  OCIRowid  *tmprid;

  tempsb2 = nctx->cons_ind[i];
  nctx->cons_ind[i] = nctx->cons_ind[j];
  nctx->cons_ind[j] = tempsb2;
  tempub2 = nctx->cons_len[i];
  nctx->cons_len[i] = nctx->cons_len[j];
  nctx->cons_len[j] = tempub2;
  tempub2 = nctx->cons_rcode[i];
  nctx->cons_rcode[i] = nctx->cons_rcode[j];
  nctx->cons_rcode[j] = tempub2;
  tempi = nctx->cons[i];
  nctx->cons[i] = nctx->cons[j];
  nctx->cons[j] = tempi;

  tempsb2 = nctx->s_rowid_ind[i];
  nctx->s_rowid_ind[i] = nctx->s_rowid_ind[j];
  nctx->s_rowid_ind[j] = tempsb2;
  tempub2 = nctx->s_rowid_len[i];
  nctx->s_rowid_len[i] = nctx->s_rowid_len[j];
  nctx->s_rowid_len[j] = tempub2;
  tempub2 = nctx->s_rowid_rcode[i];
  nctx->s_rowid_rcode[i] = nctx->s_rowid_rcode[j];
  nctx->s_rowid_rcode[j] = tempub2;
  tmprid = nctx->s_rowid_ptr[i];
  nctx->s_rowid_ptr[i]= nctx->s_rowid_ptr[j];
  nctx->s_rowid_ptr[j]=tmprid;

  tempsb2 = nctx->i_price_ind[i];
  nctx->i_price_ind[i] = nctx->i_price_ind[j];
  nctx->i_price_ind[j] = tempsb2;
  tempub2 = nctx->i_price_len[i];
  nctx->i_price_len[i] = nctx->i_price_len[j];
  nctx->i_price_len[j] = tempub2;
  tempub2 = nctx->i_price_rcode[i];
  nctx->i_price_rcode[i] = nctx->i_price_rcode[j];
  nctx->i_price_rcode[j] = tempub2;
  tempf = i_price[i];
  i_price[i] = i_price[j];
  i_price[j] = tempf;

  tempsb2 = nctx->i_name_ind[i];
  nctx->i_name_ind[i] = nctx->i_name_ind[j];
  nctx->i_name_ind[j] = tempsb2;
  tempub2 = nctx->i_name_len[i];
  nctx->i_name_len[i] = nctx->i_name_len[j];
  nctx->i_name_len[j] = tempub2;
  tempub2 = nctx->i_name_rcode[i];
  nctx->i_name_rcode[i] = nctx->i_name_rcode[j];
  nctx->i_name_rcode[j] = tempub2;
  strncpy (tempstr, i_name[i], 25);
  strncpy (i_name[i], i_name[j], 25);
  strncpy (i_name[j], tempstr, 25);

  tempsb2 = nctx->i_data_ind[i];
  nctx->i_data_ind[i] = nctx->i_data_ind[j];
  nctx->i_data_ind[j] = tempsb2;
  tempub2 = nctx->i_data_len[i];
  nctx->i_data_len[i] = nctx->i_data_len[j];
  nctx->i_data_len[j] = tempub2;
  tempub2 = nctx->i_data_rcode[i];
  nctx->i_data_rcode[i] = nctx->i_data_rcode[j];
  nctx->i_data_rcode[j] = tempub2;
  strncpy (tempstr, nctx->i_data[i], 51);
  strncpy (nctx->i_data[i], nctx->i_data[j], 51);
  strncpy (nctx->i_data[j], tempstr, 51);

  tempsb2 = nctx->s_quantity_ind[i];
  nctx->s_quantity_ind[i] = nctx->s_quantity_ind[j];
  nctx->s_quantity_ind[j] = tempsb2;
  tempub2 = nctx->s_quantity_len[i];
  nctx->s_quantity_len[i] = nctx->s_quantity_len[j];
  nctx->s_quantity_len[j] = tempub2;
  tempub2 = nctx->s_quantity_rcode[i];
  nctx->s_quantity_rcode[i] = nctx->s_quantity_rcode[j];
  nctx->s_quantity_rcode[j] = tempub2;
  tempi = s_quantity[i];
  s_quantity[i] = s_quantity[j];
  s_quantity[j] = tempi;

  tempsb2 = nctx->s_dist_info_ind[i];
  nctx->s_dist_info_ind[i] = nctx->s_dist_info_ind[j];
  nctx->s_dist_info_ind[j] = tempsb2;
  tempub2 = nctx->s_dist_info_len[i];
  nctx->s_dist_info_len[i] = nctx->s_dist_info_len[j];
  nctx->s_dist_info_len[j] = tempub2;
  tempub2 = nctx->s_dist_info_rcode[i];
  nctx->s_dist_info_rcode[i] = nctx->s_dist_info_rcode[j];
  nctx->s_dist_info_rcode[j] = tempub2;
  strncpy (tempstr, nctx->s_dist_info[i], 25);
  strncpy (nctx->s_dist_info[i], nctx->s_dist_info[j], 25);
  strncpy (nctx->s_dist_info[j], tempstr, 25);

  tempsb2 = nctx->s_data_ind[i];
  nctx->s_data_ind[i] = nctx->s_data_ind[j];
  nctx->s_data_ind[j] = tempsb2;
  tempub2 = nctx->s_data_len[i];
  nctx->s_data_len[i] = nctx->s_data_len[j];
  nctx->s_data_len[j] = tempub2;
  tempub2 = nctx->s_data_rcode[i];
  nctx->s_data_rcode[i] = nctx->s_data_rcode[j];
  nctx->s_data_rcode[j] = tempub2;
  strncpy (tempstr, nctx->s_data[i], 51);
  strncpy (nctx->s_data[i], nctx->s_data[j], 51);
  strncpy (nctx->s_data[j], tempstr, 51);
  return 0;
}

SelItemStk ()


{
```

```
  int i, j, rpc3,rcount;

  /* array select from item and stock tables */
  execstatus=OCIStmtExecute(tpcsvc,(nctx->curn3)[d_id-1],errhp,o_ol_cnt,
                                0,0,0,OCI_DEFAULT);
  if((execstatus != OCI_SUCCESS) && (execstatus != OCI_NO_DATA)) {
    errcode = OCIERROR(errhp,execstatus);
    if(errcode == NOT_SERIALIZABLE) {
      retries++;
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      return (-2);
    } else if (errcode == RECOVERR) {
    /* In case of NO_DATA this should NOT return, but simply fall through */
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      retries++;
      return (-2);
    } else {
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      return (-1);
    }
  }
  /* mark invalid items */
  OCIAttrGet((nctx->curn3)[d_id-1], OCI_HTYPE_STMT,&rcount,NULL,
                        OCI_ATTR_ROW_COUNT, errhp);
  rpc3 = rcount;

  /* the result is in order, so we have to shift up to fill    */
  /* the slot for the line with the invalid item.              */
  /* If more than one item is wrong, this is not an simulated  */
  /* error and we'll blow off */

  if ((status = o_ol_cnt - rcount) >1)
  {
#ifdef TUX
      userlog ("TPC-C server %d: more than 1 invalid item?\n", proc_no);
#else
      fprintf (stderr, "TPC-C server %d: more than 1 invalid item?\n", proc_no);
#endif
    return (rpc3);
  }
  if (status == 0) return (rpc3);

  /* find the invalid item, transfer the rowid information */

  for (i = 0; i < o_ol_cnt; i++) {
    if (nctx->cons[i] != i) break;   /* this item is invalid */
  }


#ifdef TUX
          userlog ("TPC-C server %d: reordering items and stocks\n",
                  proc_no);
#else
/*
          fprintf (stderr, "TPC-C server %d: reordering items and stocks\n",
                  proc_no);   */
#endif

  /* not the last item - shift up */

  for (j = i; j < o_ol_cnt-1; j++)
  {
    shiftitemstock (j, j+1);
  }
  /* zero the last item */
    i = o_ol_cnt-1;
    nctx->nol_i_id_ind[i] = NA;
    nctx->nol_supply_w_id_ind[i] = NA;
    nctx->nol_quantity_ind[i] = NA;
    nctx->nol_amount_ind[i] = NA;
    nctx->ol_w_id_ind[i] = NA;
    nctx->ol_d_id_ind[i] = NA;
    nctx->ol_o_id_ind[i] = NA;
    nctx->null_date_ind[i]= NA;
    nctx->ol_number_ind[i] = NA;
    nctx->ol_dist_info_ind[i] = NA;
    nctx->s_remote_ind[i] = NA;
    nctx->s_quant_ind[i] = NA;

    nctx->nol_i_id_len[i] = 0;
    nctx->nol_supply_w_id_len[i] = 0;
    nctx->nol_quantity_len[i] = 0;
    nctx->nol_amount_len[i] = 0;
    nctx->ol_w_id_len[i] = 0;
    nctx->ol_d_id_len[i] = 0;
    nctx->ol_o_id_len[i] = 0;
    nctx->ol_number_len[i] = 0;
    nctx->ol_dist_info_len[i] = 0;
    nctx->null_date_ind[i] = 0;
    nctx->s_remote_len[i] = 0;
    nctx->s_quant_len[i] = 0;

  return (rpc3);

}


#ifdef OPS
UpdStk ()

{
  int rcount;
  /* array update of stock table */

  execstatus = OCIStmtExecute(tpcsvc,nctx->curn2,errhp,o_ol_cnt,
                        0,0,0,OCI_DEFAULT);
  if(execstatus != OCI_SUCCESS) {
    OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
    errcode = OCIERROR(errhp,execstatus);
    if(errcode == NOT_SERIALIZABLE) {
      retries++;
                return (-2);
    } else if (errcode == RECOVERR) {
      retries++;
                return (-2);
    } else {
      return -1;
    }
  }
  OCIAttrGet(nctx->curn2,OCI_HTYPE_STMT,&rcount,NULL,
          OCI_ATTR_ROW_COUNT, errhp);
  return (rcount);
}
#endif



#ifndef OPS
UpdStk2 ()

{

  int rpc, rowoff, iters,rcount;

  /* array update of stock table */

  execstatus = OCIStmtExecute(tpcsvc,nctx->curn2,errhp,o_ol_cnt-status,0,0,0,
                                                OCI_DEFAULT);
  if(execstatus != OCI_SUCCESS) {
    OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
    errcode = OCIERROR(errhp,execstatus);
    if(errcode == NOT_SERIALIZABLE) {
      retries++;
                return (-2);
    } else if (errcode == RECOVERR) {
      retries++;
                return (-2);
    } else {
      return -1;
    }
  }
  OCIAttrGet(nctx->curn2,OCI_HTYPE_STMT,&rcount,NULL,
OCI_ATTR_ROW_COUNT, errhp);
  rpc = rcount;

  if (rpc != (o_ol_cnt - status)) {
#ifdef TUX
      userlog ("Error in TPC-C server %d: array update failed\n",
              proc_no);
#else
      fprintf (stderr, "Error in TPC-C server %d: array update failed\n",
              proc_no);
#endif
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      return (-1);
  }

  return (rpc);

}
#endif
```

plpay_ora.c

```
/*================================================================
=========+
|       Copyright (c) 1995  Oracle Corp, Redwood Shores, CA      |
```

```
|                OPEN SYSTEMS PERFORMANCE GROUP                |
|                   All Rights Reserved                        |

+============================================================
==========+
| FILENAME
|   plpay.c
| DESCRIPTION
|   OCI version (using PL/SQL stored procedure) of
|   PAYMENT transaction in TPC-C benchmark.

+============================================================
==========*/

#include "tpcc_ora.h"
#include "tpccpl_ora.h"
#include "plpay_ora.h"

#define SQLTXT_INIT "BEGIN pay.pay_init; END;"

payctx *pctx;

plpayinit ()

{

  text stmbuf[SQL_BUF_SIZE];
  pctx = (payctx *)malloc(sizeof(payctx));
  memset(pctx,(char)0,sizeof(payctx));

/* cursor for init */
  OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)(&(pctx->curpi)),
                OCI_HTYPE_STMT,0,(dvoid**)0));

  OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)(&(pctx->curp0)),
                OCI_HTYPE_STMT,0,(dvoid**)0));
  OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)(&(pctx->curp1)),
                OCI_HTYPE_STMT,0,(dvoid**)0));

  /*  build the init statement  and execute it */

  sprintf ((char*)stmbuf, SQLTXT_INIT);
  OCIERROR(errhp,OCIStmtPrepare(pctx->curpi, errhp, stmbuf,
                strlen((char *)stmbuf), OCI_NTV_SYNTAX,
OCI_DEFAULT));
  OCIERROR(errhp,
    OCIStmtExecute(tpcsvc,pctx->curpi,errhp,1,0,0,0,OCI_DEFAULT));

  /* customer id !=  0, go by last name */
#ifdef ATOMA
  if (sqlfile("payid_abort.sql",stmbuf))
    return(1);
#else
  if (sqlfile("pay_id.sql", stmbuf))
    return(1);
#endif
  OCIERROR(errhp,OCIStmtPrepare(pctx->curp0, errhp, stmbuf,
                strlen((char *)stmbuf), OCI_NTV_SYNTAX,
OCI_DEFAULT));

  /* customer id ==  0, go by last name */
#ifdef ATOMA
  if (sqlfile("payln_abort.sql",stmbuf))
    return(1);
#else
   if(sqlfile("pay_ln.sql", stmbuf))
    return(1);
#endif
  OCIERROR(errhp,OCIStmtPrepare(pctx->curp1, errhp, stmbuf,
                strlen((char *)stmbuf), OCI_NTV_SYNTAX,
OCI_DEFAULT));

  pctx->w_id_ind = TRUE;
  pctx->w_id_len = SIZ(w_id);
  pctx->d_id_ind = TRUE;
  pctx->d_id_len = SIZ(d_id);
  pctx->c_w_id_ind = TRUE;
  pctx->c_w_id_len = SIZ(c_w_id);
  pctx->c_d_id_ind = TRUE;
  pctx->c_d_id_len = SIZ(c_d_id);
  pctx->c_id_ind = TRUE;
  pctx->c_id_len = 0;
  pctx->h_amount_len = SIZ(h_amount);
  pctx->h_amount_ind = TRUE;
  pctx->c_last_ind = TRUE;
  pctx->c_last_len = 0;
  pctx->w_street_1_ind = TRUE;
  pctx->w_street_1_len = 0;
  pctx->w_street_2_ind = TRUE;
  pctx->w_street_2_len = 0;
  pctx->w_city_ind = TRUE;
  pctx->w_city_len = 0;
  pctx->w_state_ind = TRUE;
  pctx->w_state_len = 0;
  pctx->w_zip_ind = TRUE;
  pctx->w_zip_len = 0;
  pctx->d_street_1_ind = TRUE;
  pctx->d_street_1_len = 0;
  pctx->d_street_2_ind = TRUE;
  pctx->d_street_2_len = 0;
  pctx->d_city_ind = TRUE;
  pctx->d_city_len = 0;
  pctx->d_state_ind = TRUE;
  pctx->d_state_len = 0;
  pctx->d_zip_ind = TRUE;
  pctx->d_zip_len = 0;
  pctx->c_first_ind = TRUE;
  pctx->c_first_len = 0;
  pctx->c_middle_ind = TRUE;
  pctx->c_middle_len = 0;
  pctx->c_street_1_ind = TRUE;
  pctx->c_street_1_len = 0;
  pctx->c_street_2_ind = TRUE;
  pctx->c_street_2_len = 0;
  pctx->c_city_ind = TRUE;
  pctx->c_city_len = 0;
  pctx->c_state_ind = TRUE;
  pctx->c_state_len = 0;
  pctx->c_zip_ind = TRUE;
  pctx->c_zip_len = 0;
  pctx->c_phone_ind = TRUE;
  pctx->c_phone_len = 0;
  pctx->c_since_ind = TRUE;
  pctx->c_since_len = 0;
  pctx->c_credit_ind = TRUE;
  pctx->c_credit_len = 0;
  pctx->c_credit_lim_ind = TRUE;
  pctx->c_credit_lim_len = 0;
  pctx->c_discount_ind = TRUE;
  pctx->c_discount_len = 0;
  pctx->c_balance_ind = TRUE;
  pctx->c_balance_len = sizeof(double);
  pctx->c_data_ind = TRUE;
  pctx->c_data_len = 0;
  pctx->h_date_ind = TRUE;
  pctx->h_date_len = 0;
  pctx->retries_ind = TRUE;
  pctx->retries_len = 0;
  pctx->cr_date_ind = TRUE;
  pctx->cr_date_len = 7;



  /* bind variables */


  OCIBNDR(pctx->curp0, pctx->w_id_bp, errhp,":w_id",ADR(w_id),SIZ(int),
                SQLT_INT, &pctx->w_id_ind, NULL, NULL);
  OCIBNDR(pctx->curp0, pctx->d_id_bp, errhp,":d_id",ADR(d_id),SIZ(int),
                SQLT_INT, &pctx->d_id_ind, NULL, NULL);
  OCIBND(pctx->curp0, pctx->c_w_id_bp,
errhp,":c_w_id",ADR(c_w_id),SIZ(int),
                SQLT_INT);
  OCIBND(pctx->curp0, pctx->c_d_id_bp,
errhp,":c_d_id",ADR(c_d_id),SIZ(int),
                SQLT_INT);
  OCIBND(pctx->curp0, pctx->c_id_bp, errhp,":c_id",ADR(c_id),SIZ(int),
                SQLT_INT);
  OCIBNDR(pctx->curp0, pctx->h_amount_bp,
errhp,":h_amount",ADR(h_amount),
                SIZ(int),SQLT_INT, &pctx->h_amount_ind, &pctx-
>h_amount_len,
                &pctx->h_amount_rc);
  OCIBNDR(pctx->curp0, pctx->c_last_bp, errhp,":c_last",c_last,SIZ(c_last),
                SQLT_STR, &pctx->c_last_ind, &pctx->c_last_len, &pctx-
>c_last_rc);
  OCIBNDR(pctx->curp0, pctx->w_street_1_bp, errhp,":w_street_1",w_street_1,
                SIZ(w_street_1),SQLT_STR, &pctx->w_street_1_ind,
                &pctx->w_street_1_len, &pctx->w_street_1_rc);
  OCIBNDR(pctx->curp0, pctx->w_street_2_bp, errhp,":w_street_2",w_street_2,
                SIZ(w_street_2),SQLT_STR, &pctx->w_street_2_ind,
                &pctx->w_street_2_len, &pctx->w_street_2_rc);
  OCIBNDR(pctx->curp0, pctx->w_city_bp, errhp,":w_city",w_city,SIZ(w_city),
                SQLT_STR, &pctx->w_city_ind, &pctx->w_city_len, &pctx-
>w_city_rc);
  OCIBNDR(pctx->curp0, pctx->w_state_bp,
errhp,":w_state",w_state,SIZ(w_state),
```

```
                SQLT_STR, &pctx->w_state_ind, &pctx->w_state_len, &pctx-
>w_state_rc);
  OCIBNDR(pctx->curp0, pctx->w_zip_bp, errhp,":w_zip",w_zip,SIZ(w_zip),
                SQLT_STR, &pctx->w_zip_ind, &pctx->w_zip_len, &pctx-
>w_zip_rc);
  OCIBNDR(pctx->curp0, pctx->d_street_1_bp, errhp,":d_street_1",d_street_1,
                SIZ(d_street_1),SQLT_STR, &pctx->d_street_1_ind,
                &pctx->d_street_1_len, &pctx->d_street_1_rc);
  OCIBNDR(pctx->curp0, pctx->d_street_2_bp, errhp,":d_street_2",d_street_2,
                SIZ(d_street_2),SQLT_STR, &pctx->d_street_2_ind,
                &pctx->d_street_2_len, &pctx->d_street_2_rc);
  OCIBNDR(pctx->curp0, pctx->d_city_bp, errhp,":d_city",d_city,SIZ(d_city),
                SQLT_STR, &pctx->d_city_ind, &pctx->d_city_len, &pctx-
>d_city_rc);
  OCIBNDR(pctx->curp0, pctx->d_state_bp,
errhp,":d_state",d_state,SIZ(d_state),
                SQLT_STR, &pctx->d_state_ind, &pctx->d_state_len, &pctx-
>d_state_rc);
  OCIBNDR(pctx->curp0, pctx->d_zip_bp, errhp,":d_zip",d_zip,SIZ(d_zip),
                SQLT_STR, &pctx->d_zip_ind, &pctx->d_zip_len, &pctx-
>d_zip_rc);
  OCIBNDR(pctx->curp0, pctx->c_first_bp, errhp,":c_first",c_first,SIZ(c_first),
                SQLT_STR, &pctx->c_first_ind, &pctx->c_first_len, &pctx-
>c_first_rc);
  OCIBNDR(pctx->curp0, pctx->c_middle_bp, errhp,":c_middle",c_middle,2,
                SQLT_AFC, &pctx->c_middle_ind, &pctx->c_middle_len,
                &pctx->c_middle_rc);
  OCIBNDR(pctx->curp0, pctx->c_street_1_bp, errhp,":c_street_1",c_street_1,
                SIZ(c_street_1),SQLT_STR, &pctx->c_street_1_ind,
                &pctx->c_street_1_len, &pctx->c_street_1_rc);
  OCIBNDR(pctx->curp0, pctx->c_street_2_bp, errhp,":c_street_2",c_street_2,
                SIZ(c_street_2),SQLT_STR, &pctx->c_street_2_ind,
                &pctx->c_street_2_len, &pctx->c_street_2_rc);
  OCIBNDR(pctx->curp0, pctx->c_city_bp, errhp,":c_city",c_city,SIZ(c_city),
                SQLT_STR, &pctx->c_city_ind, &pctx->c_city_len, &pctx-
>c_city_rc);
  OCIBNDR(pctx->curp0, pctx->c_state_bp,
errhp,":c_state",c_state,SIZ(c_state),
                SQLT_STR, &pctx->c_state_ind, &pctx->c_state_len, &pctx-
>c_state_rc);
  OCIBNDR(pctx->curp0, pctx->c_zip_bp, errhp,":c_zip",c_zip,SIZ(c_zip),
                SQLT_STR, &pctx->c_zip_ind, &pctx->c_zip_len, &pctx-
>c_zip_rc);
  OCIBNDR(pctx->curp0, pctx->c_phone_bp,
errhp,":c_phone",c_phone,SIZ(c_phone),          SQLT_STR, &pctx-
>c_phone_ind, &pctx->c_phone_len, &pctx->c_phone_rc);
  OCIBNDR(pctx->curp0, pctx->c_since_bp, errhp,":c_since",c_since,7,
                SQLT_DAT, &pctx->c_since_ind, &pctx->c_since_len, &pctx-
>c_since_rc);
  OCIBNDR(pctx->curp0, pctx->c_credit_bp, errhp,":c_credit",c_credit,
                SIZ(c_credit),SQLT_CHR, &pctx->c_credit_ind, &pctx-
>c_credit_len,
                &pctx->c_credit_rc);
  OCIBNDR(pctx->curp0, pctx->c_credit_lim_bp, errhp,":c_credit_lim",
                ADR(c_credit_lim),SIZ(int), SQLT_INT, &pctx-
>c_credit_lim_ind,
                &pctx->c_credit_lim_len, &pctx->c_credit_lim_rc);
  OCIBNDR(pctx->curp0, pctx->c_discount_bp, errhp,":c_discount",
                ADR(c_discount),SIZ(int), SQLT_INT, &pctx->c_discount_ind,
                &pctx->c_discount_len, &pctx->c_discount_rc);
  OCIBNDR(pctx->curp0, pctx->c_balance_bp,
errhp,":c_balance",ADR(c_balance),
                SIZ(double),SQLT_FLT, &pctx->c_balance_ind, &pctx-
>c_balance_len,
                &pctx->c_balance_rc);
  OCIBNDR(pctx->curp0, pctx->c_data_bp, errhp,":c_data",c_data,SIZ(c_data),
                SQLT_STR, &pctx->c_data_ind, &pctx->c_data_len, &pctx-
>c_data_rc);
  OCIBNDR(pctx->curp0, pctx->h_date_bp, errhp,":h_date",h_date,SIZ(h_date),
                SQLT_STR, &pctx->h_date_ind, &pctx->h_date_len, &pctx-
>h_date_rc);
  OCIBNDR(pctx->curp0, pctx->retries_bp, errhp,":retry",ADR(retries),SIZ(int),
                SQLT_INT, &pctx->retries_ind, &pctx->retries_len, &pctx-
>retries_rc);
  OCIBNDR(pctx->curp0, pctx->cr_date_bp, errhp,":cr_date",ADR(cr_date),
                SIZ(cr_date),SQLT_DAT, &pctx->cr_date_ind, &pctx-
>cr_date_len,
                &pctx->cr_date_rc);

/* ---- Binds for  the second cursor   */


  OCIBNDR(pctx->curp1, pctx->w_id_bp1, errhp,":w_id",ADR(w_id),SIZ(int),
                SQLT_INT, &pctx->w_id_ind, &pctx->w_id_len, &pctx-
>w_id_rc);
  OCIBNDR(pctx->curp1, pctx->d_id_bp1, errhp,":d_id",ADR(d_id),SIZ(int),
                SQLT_INT, &pctx->d_id_ind, &pctx->d_id_len, &pctx-
>d_id_rc);

  OCIBND(pctx->curp1, pctx->c_w_id_bp1,
errhp,":c_w_id",ADR(c_w_id),SIZ(int),
                SQLT_INT);
  OCIBND(pctx->curp1, pctx->c_d_id_bp1,
errhp,":c_d_id",ADR(c_d_id),SIZ(int),
                SQLT_INT);
  OCIBNDR(pctx->curp1, pctx->c_id_bp1, errhp,":c_id",ADR(c_id),SIZ(int),
                SQLT_INT, &pctx->c_id_ind, &pctx->c_id_len, &pctx-
>c_id_rc);
  OCIBNDR(pctx->curp1, pctx->h_amount_bp1,
errhp,":h_amount",ADR(h_amount),
                SIZ(int),SQLT_INT, &pctx->h_amount_ind, &pctx-
>h_amount_len,
                &pctx->h_amount_rc);
  OCIBND(pctx->curp1, pctx->c_last_bp1, errhp,":c_last",c_last,SIZ(c_last),
                SQLT_STR);
  OCIBNDR(pctx->curp1, pctx->w_street_1_bp1,
errhp,":w_street_1",w_street_1,
                SIZ(w_street_1),SQLT_STR, &pctx->w_street_1_ind,
                &pctx->w_street_1_len, &pctx->w_street_1_rc);
  OCIBNDR(pctx->curp1, pctx->w_street_2_bp1,
errhp,":w_street_2",w_street_2,
                SIZ(w_street_2),SQLT_STR, &pctx->w_street_2_ind,
                &pctx->w_street_2_len, &pctx->w_street_2_rc);
  OCIBNDR(pctx->curp1, pctx->w_city_bp1,
errhp,":w_city",w_city,SIZ(w_city),
                SQLT_STR, &pctx->w_city_ind, &pctx->w_city_len, &pctx-
>w_city_rc);
  OCIBNDR(pctx->curp1, pctx->w_state_bp1,
errhp,":w_state",w_state,SIZ(w_state),
                SQLT_STR, &pctx->w_state_ind, &pctx->w_state_len, &pctx-
>w_state_rc);
  OCIBNDR(pctx->curp1, pctx->w_zip_bp1, errhp,":w_zip",w_zip,SIZ(w_zip),
                SQLT_STR, &pctx->w_zip_ind, &pctx->w_zip_len, &pctx-
>w_zip_rc);
  OCIBNDR(pctx->curp1, pctx->d_street_1_bp1, errhp,":d_street_1",d_street_1,
                SIZ(d_street_1),SQLT_STR, &pctx->d_street_1_ind,
                &pctx->d_street_1_len, &pctx->d_street_1_rc);
  OCIBNDR(pctx->curp1, pctx->d_street_2_bp1, errhp,":d_street_2",d_street_2,
                SIZ(d_street_2),SQLT_STR, &pctx->d_street_2_ind,
                &pctx->d_street_2_len, &pctx->d_street_2_rc);
  OCIBNDR(pctx->curp1, pctx->d_city_bp1, errhp,":d_city",d_city,SIZ(d_city),
                SQLT_STR, &pctx->d_city_ind, &pctx->d_city_len, &pctx-
>d_city_rc);
  OCIBNDR(pctx->curp1, pctx->d_state_bp1, errhp,":d_state",d_state,
                SIZ(d_state), SQLT_STR, &pctx->d_state_ind, &pctx-
>d_state_len,
                &pctx->d_state_rc);
  OCIBNDR(pctx->curp1, pctx->d_zip_bp1, errhp,":d_zip",d_zip,SIZ(d_zip),
                SQLT_STR, &pctx->d_zip_ind, &pctx->d_zip_len, &pctx-
>d_zip_rc);
  OCIBNDR(pctx->curp1, pctx->c_first_bp1, errhp,":c_first",c_first,
                SIZ(c_first), SQLT_STR, &pctx->c_first_ind, &pctx-
>c_first_len,
                &pctx->c_first_rc);
  OCIBNDR(pctx->curp1, pctx->c_middle_bp1, errhp,":c_middle",c_middle,2,
                SQLT_AFC, &pctx->c_middle_ind, &pctx->c_middle_len,
                &pctx->c_middle_rc);


  OCIBNDR(pctx->curp1, pctx->c_street_1_bp1, errhp,":c_street_1",c_street_1,
                SIZ(c_street_1),SQLT_STR, &pctx->c_street_1_ind,
                &pctx->c_street_1_len, &pctx->c_street_1_rc);
  OCIBNDR(pctx->curp1, pctx->c_street_2_bp1, errhp,":c_street_2",c_street_2,
                SIZ(c_street_2),SQLT_STR, &pctx->c_street_2_ind,
                &pctx->c_street_2_len, &pctx->c_street_2_rc);
  OCIBNDR(pctx->curp1, pctx->c_city_bp1, errhp,":c_city",c_city,
        SIZ(c_city),SQLT_STR,
                &pctx->c_city_ind, &pctx->c_city_len, &pctx->c_city_rc);
  OCIBNDR(pctx->curp1, pctx->c_state_bp1, errhp,":c_state",c_state,
        SIZ(c_state), SQLT_STR, &pctx->c_state_ind, &pctx->c_state_len,
        &pctx->c_state_rc);
  OCIBNDR(pctx->curp1, pctx->c_zip_bp1, errhp,":c_zip",c_zip,SIZ(c_zip),
                SQLT_STR, &pctx->c_zip_ind, &pctx->c_zip_len, &pctx-
>c_zip_rc);
  OCIBNDR(pctx->curp1, pctx->c_phone_bp1, errhp,":c_phone",c_phone,
        SIZ(c_phone), SQLT_STR, &pctx->c_phone_ind, &pctx->c_phone_len,
        &pctx->c_phone_rc);
  OCIBNDR(pctx->curp1, pctx->c_since_bp1, errhp,":c_since",c_since,7,
                SQLT_DAT, &pctx->c_since_ind, &pctx->c_since_len, &pctx-
>c_since_rc);
  OCIBNDR(pctx->curp1, pctx->c_credit_bp1, errhp,":c_credit",c_credit,
        SIZ(c_credit),SQLT_CHR, &pctx->c_credit_ind, &pctx->c_credit_len,
        &pctx->c_credit_rc);
  OCIBNDR(pctx->curp1, pctx->c_credit_lim_bp1, errhp,":c_credit_lim",
                ADR(c_credit_lim),SIZ(int), SQLT_INT, &pctx-
>c_credit_lim_ind,
                &pctx->c_credit_lim_len, &pctx->c_credit_lim_rc);
  OCIBNDR(pctx->curp1, pctx->c_discount_bp1, errhp,":c_discount",
                ADR(c_discount),SIZ(int), SQLT_INT, &pctx->c_discount_ind,
```

```
                &pctx->c_discount_len, &pctx->c_discount_rc);
  OCIBNDR(pctx->curp1, pctx->c_balance_bp1,
errhp,":c_balance",ADR(c_balance),
                SIZ(double),SQLT_FLT, &pctx->c_balance_ind, &pctx-
>c_balance_len,
                &pctx->c_balance_rc);
  OCIBNDR(pctx->curp1, pctx->c_data_bp1,
errhp,":c_data",c_data,SIZ(c_data),
                SQLT_STR, &pctx->c_data_ind, &pctx->c_data_len, &pctx-
>c_data_rc);
  OCIBNDR(pctx->curp1, pctx->h_date_bp1,
errhp,":h_date",h_date,SIZ(h_date),
                SQLT_STR, &pctx->h_date_ind, &pctx->h_date_len, &pctx-
>h_date_rc);
  OCIBNDR(pctx->curp1, pctx->retries_bp1,
errhp,":retry",ADR(retries),SIZ(int),
                SQLT_INT, &pctx->retries_ind, &pctx->retries_len, &pctx-
>retries_rc);
  OCIBNDR(pctx->curp1, pctx->cr_date_bp1, errhp,":cr_date",ADR(cr_date),
                SIZ(cr_date),SQLT_DAT, &pctx->cr_date_ind, &pctx-
>cr_date_len,
                &pctx->cr_date_rc);

  return (0);

}



plpay ()

{

retry:
  pctx->w_id_ind = TRUE;
  pctx->w_id_len = SIZ(w_id);
  pctx->d_id_ind = TRUE;
  pctx->d_id_len = SIZ(d_id);
  pctx->c_w_id_ind = TRUE;
  pctx->c_w_id_len = 0;
  pctx->c_d_id_ind = TRUE;
  pctx->c_d_id_len = 0;
  pctx->c_id_ind = TRUE;
  pctx->c_id_len = 0;
  pctx->h_amount_len = SIZ(h_amount);
  pctx->h_amount_ind = TRUE;
  pctx->c_last_ind = TRUE;
  pctx->c_last_len = SIZ(c_last);
  pctx->w_street_1_ind = TRUE;
  pctx->w_street_1_len = 0;
  pctx->w_street_2_ind = TRUE;
  pctx->w_street_2_len = 0;
  pctx->w_city_ind = TRUE;
  pctx->w_city_len = 0;
  pctx->w_state_ind = TRUE;
  pctx->w_state_len = 0;
  pctx->w_zip_ind = TRUE;
  pctx->w_zip_len = 0;
  pctx->d_street_1_ind = TRUE;
  pctx->d_street_1_len = 0;
  pctx->d_street_2_ind = TRUE;
  pctx->d_street_2_len = 0;
  pctx->d_city_ind = TRUE;
  pctx->d_city_len = 0;
  pctx->d_state_ind = TRUE;
  pctx->d_state_len = 0;
  pctx->d_zip_ind = TRUE;
  pctx->d_zip_len = 0;
  pctx->c_first_ind = TRUE;
  pctx->c_first_len = 0;
  pctx->c_middle_ind = TRUE;
  pctx->c_middle_len = 0;
  pctx->c_street_1_ind = TRUE;
  pctx->c_street_1_len = 0;
  pctx->c_street_2_ind = TRUE;
  pctx->c_street_2_len = 0;
  pctx->c_city_ind = TRUE;
  pctx->c_city_len = 0;
  pctx->c_state_ind = TRUE;
  pctx->c_state_len = 0;
  pctx->c_zip_ind = TRUE;
  pctx->c_zip_len = 0;
  pctx->c_phone_ind = TRUE;
  pctx->c_phone_len = 0;
  pctx->c_since_ind = TRUE;
  pctx->c_since_len = 0;
  pctx->c_credit_ind = TRUE;
  pctx->c_credit_len = 0;
  pctx->c_credit_lim_ind = TRUE;
  pctx->c_credit_lim_len = 0;
  pctx->c_discount_ind = TRUE;
  pctx->c_discount_len = 0;
  pctx->c_balance_ind = TRUE;
  pctx->c_balance_len = sizeof(double);
  pctx->c_data_ind = TRUE;
  pctx->c_data_len = 0;
  pctx->h_date_ind = TRUE;
  pctx->h_date_len = 0;
  pctx->retries_ind = TRUE;
  pctx->retries_len = 0;
  pctx->cr_date_ind = TRUE;
  pctx->cr_date_len = 7;

  if(bylastname) {

    execstatus=OCIStmtExecute(tpcsvc,pctx-
>curp1,errhp,1,0,0,0,OCI_DEFAULT);
  } else {
    execstatus=OCIStmtExecute(tpcsvc,pctx-
>curp0,errhp,1,0,0,0,OCI_DEFAULT);
  }
  if(execstatus != OCI_SUCCESS) {
    OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     errcode = OCIERROR(errhp,execstatus);
     if(errcode == NOT_SERIALIZABLE) {
       retries++;
       goto retry;
     } else if (errcode == RECOVERR) {
       retries++;
       goto retry;
     } else {
       return -1;
     }
  }
  return (0);

}



void plpaydone ()

{
  if(pctx) {
    free(pctx);
  }
}
```

plsto_ora.c

```
/*=============================================================
=========+
˜
|       Copyright (c) 1994  Oracle Corp, Redwood Shores, CA       |
˜
|                  OPEN SYSTEMS PERFORMANCE GROUP                 |
˜
|                       All Rights Reserved                       |

+=============================================================
=========+
| FILENAME
|    plsto.c
| DESCRIPTION
|    OCI version of STOCK LEVEL transaction in TPC-C benchmark.

+=============================================================
=========*/

#include "tpcc_ora.h"
#include "tpccpl_ora.h"


#define SQLTXTA "SELECT count (DISTINCT s_i_id) \
FROM order_line, stock, district WHERE d_id = :d_id AND d_w_id = :w_id
AND \
"

#define SQLTXTB "d_id = ol_d_id AND d_w_id = ol_w_id AND ol_i_id = s_i_id
AND \
ol_w_id = s_w_id AND s_quantity < :threshold AND \
ol_o_id BETWEEN (d_next_o_id - 20) AND (d_next_o_id - 1)"


#define SQLTXTTEST "BEGIN stocklevel.getstocklevel (:w_id, :d_id, \
:threshold); END;"
```

```
struct stoctx {
  OCIStmt *curs;
  OCIBind *w_id_bp;
  OCIBind *d_id_bp;
  OCIBind *threshold_bp;
  OCIDefine *low_stock_bp;

  int norow;
};

typedef struct stoctx stoctx;

stoctx *sctx;


plstoinit ()

{

  text stmbuf[SQL_BUF_SIZE];
  sctx = (stoctx *)malloc(sizeof(stoctx));
  memset(sctx,(char)0,sizeof(stoctx));

  sctx->norow=0;

  OCIERROR(errhp,
    OCIHandleAlloc(tpcenv,(dvoid**)&sctx-
>curs,OCI_HTYPE_STMT,0,(dvoid**)0));
  sprintf ((char *) stmbuf, "%s%s", SQLTXTA, SQLTXTB);
  OCIERROR(errhp,OCIStmtPrepare(sctx->curs,errhp,stmbuf,strlen((char
*)stmbuf),
                                OCI_NTV_SYNTAX,OCI_DEFAULT));
  OCIERROR(errhp,
    OCIAttrSet(sctx->curs,OCI_HTYPE_STMT,(dvoid*)&sctx->norow,0,
           OCI_ATTR_PREFETCH_ROWS,errhp));

  /* bind variables */

  OCIBND(sctx->curs,sctx->w_id_bp,errhp, ":w_id", ADR(w_id),sizeof(int),
       SQLT_INT);
  OCIBND(sctx->curs,sctx->d_id_bp,errhp, ":d_id", ADR(d_id),sizeof(int),
    SQLT_INT);
  OCIBND(sctx->curs,sctx->threshold_bp,errhp, ":threshold", ADR(threshold),
    sizeof(int),SQLT_INT);
  OCIDEFINE(sctx->curs,sctx->low_stock_bp,errhp, 1, ADR(low_stock),
    sizeof(int), SQLT_INT);

  return (0);

}



plsto ()

{
retry:
  execstatus=
    OCIStmtExecute(tpcsvc,sctx->curs,errhp,1,0,0,0,
                   OCI_COMMIT_ON_SUCCESS | OCI_DEFAULT);
  if(execstatus != OCI_SUCCESS) {
    OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
    errcode = OCIERROR(errhp,execstatus);
    if(errcode == NOT_SERIALIZABLE) {
      retries++;
      goto retry;
    } else if (errcode == RECOVERR) {
      retries++;
      goto retry;
    } else {
      return -1;
    }
  }
  return (0);
}



void plstodone ()

{
  if(sctx) free(sctx);
}
```

plord_ora.c

```
/*=========================================================
=========+
~
|       Copyright (c) 1995  Oracle Corp, Redwood Shores, CA      |
~
|                OPEN SYSTEMS PERFORMANCE GROUP                  |
~
|                     All Rights Reserved                        |

+=========================================================
=========+
| FILENAME
|    plord.c
| DESCRIPTION
|    OCI version (using PL/SQL anynomous block) of
|    ORDER STATUS transaction in TPC-C benchmark.

+=========================================================
=========*/

#include "tpcc_ora.h"
#include "tpccpl_ora.h"

#if defined(ISO1) || defined(ISO2) || defined(ISO8)
#define SQLTXT0 "SELECT substr(value,1,5) FROM v$parameter \
  WHERE name = 'instance_number'"
#endif

#ifdef ISO8
#define SQLTXT "BEGIN aorderstatus.agetstatus (:w_id, :d_id, :c_id, :byln, \
  :c_last, :c_first, :c_middle, :c_balance, :o_id, :o_entry_d, :o_cr_id, \
  :o_ol_cnt, :ol_s_w_id, :ol_i_id, :ol_quantity, :ol_amount, :ol_d_d); END;"
#endif

#define SQLCUR0 "SELECT rowid FROM customer \
                                WHERE c_d_id = :d_id AND c_w_id = :w_id
AND c_last = :c_last \
                                ORDER BY c_w_id, c_d_id, c_last, c_first"

#define SQLCUR1 "SELECT c_id,c_balance,c_first,c_middle, \
o_id,o_entry_d,o_carrier_id,o_ol_cnt,c_last FROM customer,orders \
WHERE customer.rowid = :cust_rowid AND o_d_id=c_d_id AND
o_w_id=c_w_id \
AND o_c_id = c_id ORDER BY o_w_id, o_d_id, o_c_id, o_id DESC"


#define SQLCUR2 "SELECT c_balance, c_first, c_middle, c_last, \
o_id,o_entry_d,o_carrier_id,o_ol_cnt FROM customer,orders \
                                WHERE c_id = :c_id AND c_d_id = :d_id AND
c_w_id = :w_id \
                                AND o_d_id = c_d_id AND o_w_id = c_w_id
AND o_c_id = c_id \
                                ORDER BY o_w_id, o_d_id, o_c_id, o_id DESC"


#define SQLCUR3 "SELECT ol_i_id,ol_supply_w_id,ol_quantity,ol_amount, \
                                ol_delivery_d\
                                FROM order_line \
                                WHERE ol_d_id = :d_id AND ol_w_id = :w_id
AND ol_o_id = :o_id"

struct ordctx {
  sb2 c_rowid_ind[100];
  sb2 ol_supply_w_id_ind[NITEMS];
  sb2 ol_i_id_ind[NITEMS];
  sb2 ol_quantity_ind[NITEMS];
  sb2 ol_amount_ind[NITEMS];
  sb2 ol_delivery_d_ind[NITEMS];
  sb2 ol_w_id_ind;
  sb2 ol_d_id_ind;
  sb2 ol_o_id_ind;
  sb2 c_id_ind;
  sb2 c_first_ind;
  sb2 c_middle_ind;
  sb2 c_balance_ind;
  sb2 c_last_ind;
  sb2 o_id_ind;
  sb2 o_entry_d_ind;
  sb2 o_carrier_id_ind;
  sb2 o_ol_cnt_ind;
#if defined(ISO1) || defined(ISO2) || defined(ISO8)
  sb2 inum_ind;
#endif


  ub2 c_rowid_len[100];
  ub2 ol_supply_w_id_len[NITEMS];
```

```
  ub2 ol_i_id_len[NITEMS];
  ub2 ol_quantity_len[NITEMS];
  ub2 ol_amount_len[NITEMS];
  ub2 ol_delivery_d_len[NITEMS];
  ub2 ol_w_id_len;
  ub2 ol_d_id_len;
  ub2 ol_o_id_len;
#if defined(ISO1) || defined(ISO2) || defined(ISO8)
  ub2 inum_len;
#endif


  ub2 c_rowid_rcode[100];
  ub2 ol_supply_w_id_rcode[NITEMS];
  ub2 ol_i_id_rcode[NITEMS];
  ub2 ol_quantity_rcode[NITEMS];
  ub2 ol_amount_rcode[NITEMS];
  ub2 ol_delivery_d_rcode[NITEMS];
  ub2 ol_w_id_rcode;
  ub2 ol_d_id_rcode;
  ub2 ol_o_id_rcode;
#if defined(ISO1) || defined(ISO2) || defined(ISO8)
  ub2 inum_rcode;
#endif

  ub4 ol_supply_w_id_csize;
  ub4 ol_i_id_csize;
  ub4 ol_quantity_csize;
  ub4 ol_amount_csize;
  ub4 ol_delivery_d_csize;
  ub4 ol_w_id_csize;
  ub4 ol_d_id_csize;
  ub4 ol_o_id_csize;
#if defined(ISO1) || defined(ISO2) || defined(ISO8)
  char inum[10];
#endif

  OCIStmt *curio0;
  OCIStmt *curo0;
  OCIStmt *curo1;
  OCIStmt *curo2;
  OCIStmt *curo3;
  OCIBind *w_id_bp0;
  OCIBind *w_id_bp2;
  OCIBind *w_id_bp3;
  OCIBind *d_id_bp0;
  OCIBind *d_id_bp2;
  OCIBind *d_id_bp3;
  OCIBind *c_id_bp;
  OCIBind *byln_bp;
  OCIBind *c_last_bp;
  OCIBind *c_first_bp;
  OCIBind *c_middle_bp;
  OCIBind *c_balance_bp;
  OCIBind *o_id_bp;
  OCIBind *o_entry_d_bp;
  OCIBind *o_cr_id_bp;
  OCIBind *o_ol_cnt_bp;
  OCIBind *ol_s_w_id_bp;
  OCIBind *ol_i_id_bp;
  OCIBind *ol_quantity_bp;
  OCIBind *ol_amount_bp;
  OCIBind *ol_d_d_bp;
  OCIBind *c_rowid_bp;
  OCIDefine *c_rowid_dp;
  OCIDefine *inum_dp;
  OCIDefine *c_last_dp;
  OCIDefine *c_last_dp1;
  OCIDefine *c_id_dp;
  OCIDefine *c_first_dp1;
  OCIDefine *c_first_dp2;
  OCIDefine *c_middle_dp1;
  OCIDefine *c_middle_dp2;
  OCIDefine *c_balance_dp1;
  OCIDefine *c_balance_dp2;
  OCIDefine *o_id_dp1;
  OCIDefine *o_id_dp2;
  OCIDefine *o_entry_d_dp1;
  OCIDefine *o_entry_d_dp2;
  OCIDefine *o_cr_id_dp1;
  OCIDefine *o_cr_id_dp2;
  OCIDefine *o_ol_cnt_dp1;
  OCIDefine *o_ol_cnt_dp2;
  OCIDefine *ol_d_d_dp;
  OCIDefine *ol_i_id_dp;
  OCIDefine *ol_supply_w_id_dp;
  OCIDefine *ol_quantity_dp;
  OCIDefine *ol_amount_dp;
  OCIDefine *ol_d_base_dp;
  OCIRowid *c_rowid_ptr[100];
  int cs;
  int cust_idx;
  int norow;
};

typedef struct ordctx ordctx;

ordctx *octx;


plordinit ()

{

  int i;
  text stmbuf[SQL_BUF_SIZE];

  octx = (ordctx *) malloc (sizeof(ordctx));
  memset(octx,(char)0,sizeof(ordctx));
  octx->cs = 1;
  octx->norow = 0;
/* get the rowid handles */
  for(i=0;i<100;i++) {
   OCIERROR(errhp, OCIDescriptorAlloc(tpcenv,(dvoid**)&octx-
>c_rowid_ptr[i],
      OCI_DTYPE_ROWID,0,(dvoid**)0));
  }


#if defined(ISO1) || defined(ISO2) || defined(ISO8)
  OCIHandleAlloc(tpcenv, (dvoid **)(&octx->curio0), OCI_HTYPE_STMT, 0,
(dvoid**)
0);
  sprintf ((char *) stmbuf, SQLTXT0);
  OCIStmtPrepare(octx->curio0, errhp, stmbuf, strlen((char
*)stmbuf),OCI_NTV_SYNTAX, OCI_DEFAULT);

  OCIDFNRA(octx->curio0, octx->inum_dp,errhp,1,octx->inum,SIZ(octx-
>inum),SQLT_STR,&(octx->inum_ind),&(octx->inum_len),&(octx-
>inum_rcode));
#endif



#ifdef ISO8
  OCIERROR(errhp,
    OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo0,OCI_HTYPE_STMT,0,(dvoid**)0));
#else
  OCIERROR(errhp,
    OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo0,OCI_HTYPE_STMT,0,(dvoid**)0));
  OCIERROR(errhp,
    OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo1,OCI_HTYPE_STMT,0,(dvoid**)0));
  OCIERROR(errhp,
    OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo2,OCI_HTYPE_STMT,0,(dvoid**)0));
  OCIERROR(errhp,
    OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo3,OCI_HTYPE_STMT,0,(dvoid**)0));
#endif

#ifdef ISO8
  sprintf((char *) stmbuf, SQLTXT);
  OCIERROR(errhp,
    OCIStmtPrepare(octx->curo0,errhp,stmbuf,strlen((char *)stmbuf),
                   OCI_NTV_SYNTAX,OCI_DEFAULT));
#else

/*  c_id = 0, use find customer by lastname. Get an array or rowid's back*/
  sprintf((char *) stmbuf, SQLCUR0);
  OCIERROR(errhp,
    OCIStmtPrepare(octx->curo0,errhp,stmbuf,strlen((char *)stmbuf),
                   OCI_NTV_SYNTAX,OCI_DEFAULT));
  OCIERROR(errhp,
    OCIAttrSet(octx->curo0,OCI_HTYPE_STMT,(dvoid*)&octx->norow,0,
                   OCI_ATTR_PREFETCH_ROWS,errhp));

/* get order/customer info back based on rowid */
  sprintf((char *) stmbuf, SQLCUR1);
  OCIERROR(errhp,
    OCIStmtPrepare(octx->curo1,errhp,stmbuf,strlen((char *)stmbuf),
                   OCI_NTV_SYNTAX,OCI_DEFAULT));
  OCIERROR(errhp,
    OCIAttrSet(octx->curo1,OCI_HTYPE_STMT,(dvoid*)&octx->norow,0,
                   OCI_ATTR_PREFETCH_ROWS,errhp));
```

```
/*  c_id == 0, use lastname to find customer */
  sprintf((char *) stmbuf, SQLCUR2);
  OCIERROR(errhp,
    OCIStmtPrepare(octx->curo2,errhp,stmbuf,strlen((char *)stmbuf),
                                OCI_NTV_SYNTAX,OCI_DEFAULT));
  OCIERROR(errhp,
    OCIAttrSet(octx->curo2,OCI_HTYPE_STMT,(dvoid*)&octx->norow,0,
                                OCI_ATTR_PREFETCH_ROWS,errhp));

  sprintf((char *) stmbuf, SQLCUR3);
  OCIERROR(errhp,
    OCIStmtPrepare(octx->curo3,errhp,stmbuf,strlen((char *)stmbuf),
                                OCI_NTV_SYNTAX,OCI_DEFAULT));
  OCIERROR(errhp,
    OCIAttrSet(octx->curo3,OCI_HTYPE_STMT,(dvoid*)&octx->norow,0,
                                OCI_ATTR_PREFETCH_ROWS,errhp));
#endif

  for (i = 0; i < NITEMS; i++) {
    octx->ol_supply_w_id_ind[i] = TRUE;
    octx->ol_i_id_ind[i] = TRUE;
    octx->ol_quantity_ind[i] = TRUE;
    octx->ol_amount_ind[i] = TRUE;
    octx->ol_delivery_d_ind[i] = TRUE;
    octx->ol_supply_w_id_len[i] = sizeof(int);
    octx->ol_i_id_len[i] = sizeof(int);
    octx->ol_quantity_len[i] = sizeof(int);
    octx->ol_amount_len[i] = sizeof(int);
    octx->ol_delivery_d_len[i] = sizeof(ol_d_base[0]);
  }
  octx->ol_supply_w_id_csize = NITEMS;
  octx->ol_i_id_csize = NITEMS;
  octx->ol_quantity_csize = NITEMS;
  octx->ol_amount_csize = NITEMS;
  octx->ol_delivery_d_csize = NITEMS;
  octx->ol_w_id_csize = NITEMS;
  octx->ol_o_id_csize = NITEMS;
  octx->ol_d_id_csize = NITEMS;
  octx->ol_w_id_ind = TRUE;
  octx->ol_d_id_ind = TRUE;
  octx->ol_o_id_ind = TRUE;
  octx->ol_w_id_len = sizeof(int);
  octx->ol_d_id_len = sizeof(int);
  octx->ol_o_id_len = sizeof(int);

 /* bind variables */
#ifdef ISO8

 OCIBND(octx->curo0,octx->w_id_bp0,errhp,":w_id",ADR(w_id),SIZ(w_id),
                SQLT_INT);
  OCIBND(octx->curo0, octx->d_id_bp0, errhp,":d_id",ADR(d_id),SIZ(d_id),
          SQLT_INT);
  OCIBND(octx->curo0, octx->c_id_bp, errhp,":c_id",ADR(c_id),SIZ(c_id),
          SQLT_INT);
  OCIBND(octx->curo0, octx->byln_bp, errhp,":byln",ADR(bylastname),
          SIZ(bylastname),SQLT_INT);
  OCIBND(octx->curo0, octx->c_last_bp, errhp,":c_last",c_last,
          SIZ(c_last),SQLT_STR);
  OCIBND(octx->curo0, octx->c_first_bp, errhp,":c_first",c_first,
          SIZ(c_first),SQLT_STR);
  OCIBND(octx->curo0, octx->c_middle_bp, errhp,":c_middle",c_middle,
          SIZ(c_middle),SQLT_STR);
  OCIBND(octx->curo0, octx->c_balance_bp,
errhp,":c_balance",ADR(c_balance),
          SIZ(c_balance),SQLT_FLT);
  OCIBND(octx->curo0, octx->o_id_bp, errhp,":o_id",ADR(o_id),
          SIZ(o_id),SQLT_INT);
  OCIBND(octx->curo0, octx->o_entry_d_bp, errhp,":o_entry_d",
          o_entry_d_base,SIZ(o_entry_d_base),SQLT_DAT);
  OCIBND(octx->curo0, octx->o_cr_id_bp, errhp,":o_cr_id",ADR(o_carrier_id),
          SIZ(o_carrier_id), SQLT_INT);
  OCIBND(octx->curo0, octx->o_ol_cnt_bp, errhp,":o_ol_cnt",ADR(o_ol_cnt),
          SIZ(o_ol_cnt),SQLT_INT);
  OCIBNDRAA(octx->curo0, octx->ol_s_w_id_bp, errhp,":ol_s_w_id",
                          ol_supply_w_id,SIZ(int),SQLT_INT,
                octx->ol_supply_w_id_ind,octx->ol_supply_w_id_len,
                octx->ol_supply_w_id_rcode,NITEMS,
                          ADR(octx->ol_supply_w_id_csize));
  OCIBNDRAA(octx->curo0, octx->ol_i_id_bp, errhp,":ol_i_id",ol_i_id,
                          SIZ(int),SQLT_INT,
                octx->ol_i_id_ind,octx->ol_i_id_len,octx-
>ol_i_id_rcode,NITEMS,
                ADR(octx->ol_i_id_csize));
  OCIBNDRAA(octx->curo0, octx->ol_quantity_bp, errhp,":ol_quantity",
                          ol_quantity,SIZ(int),SQLT_INT,
                octx->ol_quantity_ind,octx->ol_quantity_len,
                          octx->ol_quantity_rcode, NITEMS,ADR(octx-
>ol_quantity_csize));
  OCIBNDRAA(octx->curo0, octx->ol_amount_bp,
errhp,":ol_amount",ol_amount,
                          SIZ(float),SQLT_FLT,
                octx->ol_amount_ind,octx->ol_amount_len,octx-
>ol_amount_rcode,
                NITEMS,ADR(octx->ol_amount_csize));
  OCIBNDRAA(octx->curo0, octx->ol_d_d_bp, errhp,":ol_d_d",ol_delivery_d,
                          SIZ(ol_delivery_d[0]),SQLT_STR,
                octx->ol_delivery_d_ind,octx->ol_delivery_d_len,
                octx->ol_delivery_d_rcode,NITEMS,
                          ADR(octx->ol_delivery_d_csize));
#else

  /* c_id (customer id) is not known */
  OCIBND(octx->curo0,octx-
>w_id_bp0,errhp,":w_id",ADR(w_id),SIZ(int),SQLT_INT);
  OCIBND(octx->curo0,octx-
>d_id_bp0,errhp,":d_id",ADR(d_id),SIZ(int),SQLT_INT);
  OCIBND(octx->curo0,octx->c_last_bp,errhp,":c_last",c_last,SIZ(c_last),
                SQLT_STR);
  OCIDFNRA(octx->curo0,octx->c_rowid_dp,errhp,1,octx->c_rowid_ptr,
                  sizeof(octx->c_rowid_ptr[0]),SQLT_RDD,octx->c_rowid_ind,
                  octx->c_rowid_len,octx->c_rowid_rcode);

  OCIBND(octx->curo1,octx->c_rowid_bp,errhp,":cust_rowid",
                &octx->c_rowid_ptr[octx->cust_idx],
                sizeof( octx->c_rowid_ptr[0]),SQLT_RDD);
  OCIDEF(octx->curo1,octx->c_id_dp,errhp,1,ADR(c_id),SIZ(int),SQLT_INT);
  OCIDEF(octx->curo1,octx->c_balance_dp1,errhp,2,ADR(c_balance),
                SIZ(double),SQLT_FLT);
  OCIDEF(octx->curo1,octx->c_first_dp1,errhp,3,c_first,SIZ(c_first),
                SQLT_STR);
  OCIDEF(octx->curo1,octx->c_middle_dp1,errhp,4,c_middle,
                SIZ(c_middle),SQLT_STR);
  OCIDEF(octx->curo1,octx-
>o_id_dp1,errhp,5,ADR(o_id),SIZ(int),SQLT_INT);
  OCIDEF(octx->curo1,octx->o_entry_d_dp1,errhp,6,
                o_entry_d_base,SIZ(o_entry_d_base),SQLT_DAT);
  OCIDEF(octx->curo1,octx->o_cr_id_dp1,errhp,7,ADR(o_carrier_id),
                SIZ(int),SQLT_INT);
  OCIDEF(octx->curo1,octx->o_ol_cnt_dp1,errhp,8,ADR(o_ol_cnt),
                SIZ(int),SQLT_INT);
  OCIDEF(octx->curo1,octx->c_last_dp1,errhp,9,c_last,SIZ(c_last),
SQLT_STR);

/* Bind for third cursor , no-zero customer id */
  OCIBND(octx->curo2,octx-
>w_id_bp2,errhp,":w_id",ADR(w_id),SIZ(int),SQLT_INT);
  OCIBND(octx->curo2,octx-
>d_id_bp2,errhp,":d_id",ADR(d_id),SIZ(int),SQLT_INT);
  OCIBND(octx->curo2,octx-
>c_id_bp,errhp,":c_id",ADR(c_id),SIZ(int),SQLT_INT);
  OCIDEF(octx->curo2,octx->c_balance_dp2,errhp,1,ADR(c_balance),
                SIZ(double),SQLT_FLT);
  OCIDEF(octx->curo2,octx->c_first_dp2,errhp,2,c_first,SIZ(c_first),
                SQLT_STR);
  OCIDEF(octx->curo2,octx->c_middle_dp2,errhp,3,c_middle,
                SIZ(c_middle),SQLT_STR);
  OCIDEF(octx->curo2,octx->c_last_dp,errhp,4,c_last,SIZ(c_last), SQLT_STR);
  OCIDEF(octx->curo2,octx-
>o_id_dp2,errhp,5,ADR(o_id),SIZ(int),SQLT_INT);
  OCIDEF(octx->curo2,octx->o_entry_d_dp2,errhp,6,
                o_entry_d_base,SIZ(o_entry_d_base),SQLT_DAT);
  OCIDEF(octx->curo2, octx->o_cr_id_dp2,errhp,7,ADR(o_carrier_id),
                SIZ(int), SQLT_INT);
  OCIDEF(octx->curo2,octx->o_ol_cnt_dp2,errhp,8,ADR(o_ol_cnt),
                SIZ(int),SQLT_INT);

/* Bind for last cursor */

  OCIBND(octx->curo3,octx-
>w_id_bp3,errhp,":w_id",ADR(w_id),SIZ(int),SQLT_INT);
  OCIBND(octx->curo3,octx-
>d_id_bp3,errhp,":d_id",ADR(d_id),SIZ(int),SQLT_INT);
  OCIBND(octx->curo3,octx-
>o_id_bp,errhp,":o_id",ADR(o_id),SIZ(int),SQLT_INT);

  OCIDFNRA(octx->curo3, octx->ol_i_id_dp, errhp, 1,
ol_i_id,SIZ(int),SQLT_INT,
                octx->ol_i_id_ind,octx->ol_i_id_len, octx->ol_i_id_rcode);
  OCIDFNRA(octx->curo3,octx->ol_supply_w_id_dp,errhp,2, ol_supply_w_id,
                SIZ(int),SQLT_INT, octx->ol_supply_w_id_ind,
                octx->ol_supply_w_id_len, octx->ol_supply_w_id_rcode);
  OCIDFNRA(octx->curo3, octx->ol_quantity_dp,errhp,3, ol_quantity,SIZ(int),
                SQLT_INT, octx->ol_quantity_ind,octx->ol_quantity_len,
                octx->ol_quantity_rcode);
  OCIDFNRA(octx->curo3,octx->ol_amount_dp,errhp,4,ol_amount, SIZ(int),
                SQLT_INT,octx->ol_amount_ind, octx->ol_amount_len,
                octx->ol_amount_rcode);
```

```
  OCIDFNRA(octx->curo3,octx->ol_d_base_dp,errhp,5,ol_d_base,7,
SQLT_DAT,
                    octx->ol_delivery_d_ind,octx->ol_delivery_d_len,
                    octx->ol_delivery_d_rcode);
#endif /* ISO8 */
  return (0);

}



plord ()

{

  int i;
  int rcount;

#if defined(ISO1) || defined(ISO2) || defined(ISO8)
  int hasno;
  char sdate[30];

  OCIStmtExecute(tpcsvc,octx->curio0,errhp,1,0,0,0,OCI_DEFAULT);
  sysdate (sdate);
  printf ("Orderstatus started at %s on node %s\n", sdate, octx->inum);
#endif


retry:


  for (i = 0; i < NITEMS; i++) {
    octx->ol_supply_w_id_ind[i] = TRUE;
    octx->ol_i_id_ind[i] = TRUE;
    octx->ol_quantity_ind[i] = TRUE;
    octx->ol_amount_ind[i] = TRUE;
    octx->ol_delivery_d_ind[i] = TRUE;
    octx->ol_supply_w_id_len[i] = sizeof(int);
    octx->ol_i_id_len[i] = sizeof(int);
    octx->ol_quantity_len[i] = sizeof(int);
    octx->ol_amount_len[i] = sizeof(int);
    octx->ol_delivery_d_len[i] = sizeof(ol_d_base[0]);
  }
  octx->ol_supply_w_id_csize = NITEMS;
  octx->ol_i_id_csize = NITEMS;
  octx->ol_quantity_csize = NITEMS;
  octx->ol_amount_csize = NITEMS;
  octx->ol_delivery_d_csize = NITEMS;
#ifdef ISO8
  OCIERROR(errhp,
    OCIStmtExecute(tpcsvc,octx->curo0,errhp,1,0,0,0,OCI_DEFAULT));
#else
  if(bylastname) {
    execstatus=OCIStmtExecute(tpcsvc,octx-
>curo0,errhp,100,0,0,0,OCI_DEFAULT);
    if (execstatus != OCI_NO_DATA)    /* will get OCI_NO_DATA if <100
found */
    {
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      errcode = OCIERROR(errhp,execstatus);
      if(errcode == NOT_SERIALIZABLE) {
        retries++;
        goto retry;
      } else if (errcode == RECOVERR) {
        retries++;
        goto retry;
      } else {
       return -1;
      }
    }
    /* get rowcount, find middle one */
    OCIAttrGet(octx-
>curo0,OCI_HTYPE_STMT,&rcount,NULL,OCI_ATTR_ROW_COUNT,errhp)
;
    octx->cust_idx=(rcount+1)/2 ;
    execstatus = OCIStmtExecute(tpcsvc,octx-
>curo1,errhp,1,0,0,0,OCI_DEFAULT);
    if (execstatus != OCI_SUCCESS)
    {
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      errcode = OCIERROR(errhp,execstatus);
      if(errcode == NOT_SERIALIZABLE) {
        retries++;
        goto retry;
      } else if (errcode == RECOVERR) {
        retries++;
        goto retry;
      } else {
       return -1;
      }
    }

  } else {
   execstatus = OCIStmtExecute(tpcsvc,octx-
>curo2,errhp,1,0,0,0,OCI_DEFAULT);
    if (execstatus != OCI_SUCCESS)
    {
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      errcode = OCIERROR(errhp,execstatus);
      if(errcode == NOT_SERIALIZABLE) {
        retries++;
        goto retry;
      } else if (errcode == RECOVERR) {
        retries++;
        goto retry;
      } else {
       return -1;
      }
    }
  }
  octx->ol_w_id_ind = TRUE;
  octx->ol_d_id_ind = TRUE;
  octx->ol_o_id_ind = TRUE;
  octx->ol_w_id_len = sizeof(int);
  octx->ol_d_id_len = sizeof(int);
  octx->ol_o_id_len = sizeof(int);

  execstatus =   OCIStmtExecute(tpcsvc,octx->curo3,errhp,o_ol_cnt,0,0,0,
                 OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
  if (execstatus != OCI_SUCCESS)
    {
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      errcode = OCIERROR(errhp,execstatus);
      if(errcode == NOT_SERIALIZABLE) {
        retries++;
        goto retry;
      } else if (errcode == RECOVERR) {
        retries++;
        goto retry;
      } else {
       return -1;
      }
    }
#ifdef NOTMORE
  OCIERROR(errhp,
    OCITransCommit(tpcsvc,errhp,OCI_DEFAULT));
#endif

  /* clean up and convert the delivery dates */
  for (i = 0; i < o_ol_cnt; i++) {
    if (octx->ol_delivery_d_ind[i] == -1) /* null date in field */
                strncpy(ol_delivery_d[i],"1-1-1811",10);
    else
                cvtdmy(ol_d_base[i],ol_delivery_d[i]);
  }
#endif
#if defined(ISO1) || defined(ISO2) || defined(ISO8)
  printf ("Orderstatus ended at %s on node %s\n", sdate, octx->inum);
#endif
  return (0);

}



void plorddone ()

{


  if (octx) {
#if defined(ISO1) || defined(ISO2) || defined(ISO8)
    OCIHandleFree((dvoid *)octx->curio0,OCI_HTYPE_STMT);
#endif
     free (octx);
  }
}
```

tmserver_dell.c

```
//----------------------tmserver.c: Dell TPC-C Transaction Monitor Server----------
-
~
//
~
// Copyright (c) 1997 Dell Computer Corporation, All Rights Reserved
~
```

```
//

// Author: James Jordan, Dave Jaffe                                    Last
modified: 01/15/98
//
// Audited: Richard Gimarc, Performance Metrics Inc. 10/9/97
//
// Transaction Monitor server code for tpcc benchmark
// jpj. 1-18-98. added ORACLE switch to call oracle backend db server
//
//

#include "tmserver_dell.h"

//------------------------------------tpsvrinit-----------------------------------------
//              main entry point for application. called by tmboot at process startup
time.
//              returns zero if able to init database and open database connection.
otherwise,
//              returns negative one(-1) which will cause tmboot to shutdown
process.

int tpsvrinit(int argc, char *argv[])
                {
                int spid = 0; // database connection process id

                // so compiler does not complain about not using variables
                argc = argc;
                argv = argv;

                // Open Error Log
                if(!OpenErrorLog()) return FALSE;
                WriteErrorLog("Error Log Opened...");

                // Read database parameters from registry
                ReadTPCCRegParams();

#ifdef _TUX_D
                if(!OpenDeliveryLog()) return FALSE;
                QueryPerformanceFrequency(&freq);
                freqd = (double) freq.QuadPart;
#endif

#ifdef ORACLE
                if (TPCinit ()) {
        fprintf(stderr, "Failed in TPCinit (probably connecting).");
        exit (1);
    }


#else
                DBInit();
                DBOpenConnection(&hDB, server, database_name, database_user,
database_passwd,
                                "Client", &spid, (long*) 4096);
#endif// end ORACLE

#ifdef _TUX_NO
                userlog("Starting NewOrder service.....");
#endif
#ifdef _TUX_PY
                userlog("Starting Payment service.....");
#endif
#ifdef _TUX_OS
                userlog("Starting OrderStatus service.....");
#endif
#ifdef _TUX_D
                userlog("Starting Delivery service.....");
#endif
#ifdef _TUX_SL
                userlog("Starting StockLevel service.....");
#endif

                userlog("TPC-C Tuxedo Server started");
                return(0);
                } // End tpsvrinit()
//
//
//------------------------------------tpsvrdone-----------------------------------------
// Exits transaction monitor server function - called by TMShutdown

void tpsvrdone( void )
                {

#ifdef ORACLE
                 TPCexit (0);
#else
                DBCloseConnection(hDB);
                DBExit();
#endif// end ORACLE

                fclose(fp_errorlog);
#ifdef _TUX_D
                fclose(fp_delivlog);
#endif
                }
//
#ifdef _TUX_NO
//------------------------------------TMNO----------------------------------------------
// Transaction Monitor NewOrder server function: calls DBNewOrder
//
void TMNO (TPSVCINFO *rqst)
                {

                int rc;

#ifdef ORACLE
   tpreturn (TPSUCCESS,rc = (TPCnew ((NEW_ORDER_DATA *) rqst->data))
,
                                rqst->data , rqst->len, 0);
#else
                tpreturn(TPSUCCESS,rc = (DBNewOrder(hDB,
(NEW_ORDER_DATA *) rqst->data)) ,
                                rqst->data, rqst->len, 0);
#endif
                }
//
#endif

#ifdef _TUX_PY
//
//------------------------------------TMPY----------------------------------------------
// Transaction Monitor Payment server function: calls DBPayment
//
void TMPY (TPSVCINFO *rqst)
                {
                int rc;
#ifdef ORACLE
                tpreturn(TPSUCCESS,rc = (TPCpay((PAYMENT_DATA *) rqst-
>data)) ,
                                rqst->data, rqst->len , 0);
#else
                tpreturn(TPSUCCESS,rc = (DBPayment(hDB,
(PAYMENT_DATA *) rqst->data)) ,
                                rqst->data, rqst->len , 0);
#endif
                }
//
#endif

#ifdef _TUX_OS
//
//------------------------------------TMOS----------------------------------------------
// Transaction Monitor OrderStatus server function: calls DBOrderStatus
//
void TMOS (TPSVCINFO *rqst)
                {
                int rc;
#ifdef ORACLE
                tpreturn(TPSUCCESS,
                                rc = (TPCord((ORDER_STATUS_DATA *) rqst-
>data)),
                                rqst->data, rqst->len, 0);
#else
                tpreturn(TPSUCCESS,
                                rc = (DBOrderStatus(hDB,
(ORDER_STATUS_DATA *) rqst->data)),
                                rqst->data, rqst->len, 0);
#endif
                }
//
#endif

#ifdef _TUX_D
//
//------------------------------------TMDL----------------------------------------------
// Transaction Monitor Delivery server function: calls DBDelivery
//
void TMDL (TPSVCINFO *rqst)
                {
                int rc;
#ifdef ORACLE
                rc = (TPCdel((DELIVERY_DATA *) rqst->data));
                tpreturn(TPSUCCESS, 0 ,
                                NULL, 0, 0);
#else
                rc = (DBDelivery(hDB, (DELIVERY_DATA *) rqst->data));
                tpreturn(TPSUCCESS, 0 ,
```

```
                                NULL, 0, 0);
#endif
        }
//
#endif

//
//-----------------------------------TMSL-----------------------------------------
// Transaction Monitor StockLevel server function: calls DBStockLevel
//
#ifdef _TUX_SL
void TMSL (TPSVCINFO *rqst)
        {
   int rc;
#ifdef ORACLE
        tpreturn(TPSUCCESS,
                rc = (TPCsto((STOCK_LEVEL_DATA *) rqst->data)) ,
                rqst->data, rqst->len, 0);
#else
        tpreturn(TPSUCCESS,
                rc = (DBStockLevel(hDB,
(STOCK_LEVEL_DATA *) rqst->data)) ,
                rqst->data, rqst->len, 0);
#endif
        }
#endif
//
//
//--------------------------------OpenErrorLog---------------------------------------
// create and open error log w/ filename error_server_XX.log in directory log_path
//
BOOL OpenErrorLog()
        {
        char Value[] = "LogPath";
        char SubKey[100];
        HKEY Key;
        DWORD Type;
        DWORD lenData = 200;
        unsigned char Text[200];
        char errorlog_fn[250];

        strcpy(SubKey, "Software\\TPCC");

        RegOpenKeyEx(HKEY_LOCAL_MACHINE, SubKey, 0,
KEY_QUERY_VALUE, &Key);
        RegQueryValueEx(Key, Value, 0, &Type, Text, &lenData);
        ExpandEnvironmentStrings((const char*) Text, log_path, 250);

        //strcpy(log_path,"c:\\jpj\\logfiles");
        // Create file pathname, open file, return FALSE if error
#ifdef _ALL_TRANS
        sprintf(errorlog_fn, "%s\\error_server_%d.log", log_path,getpid
());
#elif _TUX_NO
        sprintf(errorlog_fn, "%s\\error_server_no_%d.log", log_path,getpid
());
#elif _TUX_PY
        sprintf(errorlog_fn, "%s\\error_server_py_%d.log", log_path,getpid
());
#elif _TUX_OS
        sprintf(errorlog_fn, "%s\\error_server_os_%d.log", log_path,getpid
());
#elif _TUX_D
        sprintf(errorlog_fn, "%s\\error_server_dl_%d.log", log_path,getpid
());
#elif _TUX_SL
        sprintf(errorlog_fn, "%s\\error_server_sl_%d.log", log_path,getpid
());
#endif

        if((fp_errorlog = fopen(errorlog_fn, "w")) == NULL)
                {
                userlog("Can't open error log %s\n", errorlog_fn);
                return FALSE;
                }
        else return TRUE;

        } // OpenErrorLog
//
//
///-----------------------------ReadTPCCRegParams-----------------------------------
// read database params from registry
//
void ReadTPCCRegParams()
        {
        char SubKey[100];
        char Value[100];
        HKEY Key;
        DWORD Type;
        DWORD lenData = 200;
        unsigned char Text[200];
        char message[100];

        strcpy(SubKey, "Software\\TPCC");

        lenData = 200;
        strcpy(Value, "DatabaseServer");
        RegOpenKeyEx(HKEY_LOCAL_MACHINE, SubKey, 0,
KEY_QUERY_VALUE, &Key);
        RegQueryValueEx(Key, Value, 0, &Type, Text, &lenData);
        strncpy(server, (const char*)Text, lenData);
        sprintf(message, "server= %s", server);
        WriteErrorLog(message);

        lenData = 200;
        strcpy(Value, "DatabaseName");
        RegOpenKeyEx(HKEY_LOCAL_MACHINE, SubKey, 0,
KEY_QUERY_VALUE, &Key);
        RegQueryValueEx(Key, Value, 0, &Type, Text, &lenData);
        strncpy(database_name, (const char*)Text, lenData);
        sprintf(message, "database_name= %s", database_name);
        WriteErrorLog(message);

        lenData = 200;
        strcpy(Value, "DatabaseUser");
        RegOpenKeyEx(HKEY_LOCAL_MACHINE, SubKey, 0,
KEY_QUERY_VALUE, &Key);
        RegQueryValueEx(Key, Value, 0, &Type, Text, &lenData);
        strncpy(database_user, (const char*)Text, lenData);
        sprintf(message, "database_user= %s", database_user);
        WriteErrorLog(message);

        lenData = 200;
        strcpy(Value, "DatabasePassword");
        RegOpenKeyEx(HKEY_LOCAL_MACHINE, SubKey, 0,
KEY_QUERY_VALUE, &Key);
        RegQueryValueEx(Key, Value, 0, &Type, Text, &lenData);
        strncpy(database_passwd, (const char*)Text, lenData);
        sprintf(message, "database_passwd= %s", database_passwd);
        WriteErrorLog(message);

        } // ReadTPCCRegParams
//
//
//--------------------------------OpenDeliveryLog-------------------------------------
// create and open delivery log w/ filename log_path\dyymmddhhmm_processid.log
//
#ifdef _TUX_D
BOOL OpenDeliveryLog()
        {
        char d[10];
        char t[10];
        char delivlog_fn[64];

        InitializeCriticalSection(&deliv_write_crit_sec);

        _strdate(d);
        _strtime(t);
        sprintf(delivlog_fn,
"%s\\d%c%c%c%c%c%c%c%c%c%c_%d.log",
                log_path, d[6], d[7], d[0], d[1], d[3], d[4], t[0], t[1],
t[3], t[4],
                GetCurrentProcessId());
        if((fp_delivlog = fopen(delivlog_fn, "w")) == NULL)
                {
                return FALSE;
                }
        else return TRUE;
        } // OpenDeliveryLog
#endif
//
//
//--------------------------------WriteErrorLog---------------------------------------
// create and open error log w/ filename error.log in directory log_path
//
void WriteErrorLog(char* message)
        {
        char d[10];
        char t[10];
        struct _timeb tb;

        _strdate(d);
        _strtime(t);
        _ftime(&tb);
        fprintf(fp_errorlog, "%s %s.%03u Thread: 0x%03X %s\n",
                d, t, tb.millitm, GetCurrentThreadId(), message);
        fflush(fp_errorlog);
```

```
		} // WriteErrorLog
//
//
//
```

//----------End of tmserver.c----------

tmserver_stub_dell.c

```
//----------tmserver_stub.c: Dell TPC-C Transaction Monitor Server----------
//
// Copyright (c) 1997 Dell Computer Corporation, All Rights Reserved
//
// Author: James Jordan                                          Last modified: 9/24/97
//
// Audited: Richard Gimarc, Performance Metrics Inc. 9/24/97
//
// Transaction Monitor server code for tpcc benchmark
//
//

#include <stdio.h>
#include <xa.h>
#include <atmi.h>

#if defined(__cplusplus)
extern "C" {
#endif
extern int _tmrunserver _((int));
#ifdef _TUX_D
extern void TMDL _((TPSVCINFO *));
#endif
#ifdef _TUX_NO
extern void TMNO _((TPSVCINFO *));
#endif
#ifdef _TUX_OS
extern void TMOS _((TPSVCINFO *));
#endif
#ifdef _TUX_PY
extern void TMPY _((TPSVCINFO *));
#endif
#ifdef _TUX_SL
extern void TMSL _((TPSVCINFO *));
#endif
#if defined(__cplusplus)
}
#endif

#ifdef _ALL_TRANS
static struct tmdsptchtbl_t _tmdsptchtbl[] = {
#ifdef _TUX_D
		{ "TMDL", "TMDL", (void (*) _((TPSVCINFO *))) TMDL, 0, 0
},
#endif
#ifdef _TUX_NO
		{ "TMNO", "TMNO", (void (*) _((TPSVCINFO *))) TMNO, 1, 0
},
#endif
#ifdef _TUX_OS
		{ "TMOS", "TMOS", (void (*) _((TPSVCINFO *))) TMOS, 2, 0
},
#endif
#ifdef _TUX_PY
		{ "TMPY", "TMPY", (void (*) _((TPSVCINFO *))) TMPY, 3, 0
},
#endif
#ifdef _TUX_SL
		{ "TMSL", "TMSL", (void (*) _((TPSVCINFO *))) TMSL, 4, 0 },
#endif
		{ NULL, NULL, NULL, 0, 0 }
};
#else
static struct tmdsptchtbl_t _tmdsptchtbl[] = {
#ifdef _TUX_D
		{ "TMDL", "TMDL", (void (*) _((TPSVCINFO *))) TMDL, 0, 0
},
#endif
#ifdef _TUX_NO
		{ "TMNO", "TMNO", (void (*) _((TPSVCINFO *))) TMNO, 0, 0
},
#endif
#ifdef _TUX_OS
		{ "TMOS", "TMOS", (void (*) _((TPSVCINFO *))) TMOS, 0, 0
},
#endif
#ifdef _TUX_PY
		{ "TMPY", "TMPY", (void (*) _((TPSVCINFO *))) TMPY, 0, 0
},
#endif
#ifdef _TUX_SL
		{ "TMSL", "TMSL", (void (*) _((TPSVCINFO *))) TMSL, 0, 0 },
#endif
		{ NULL, NULL, NULL, 0, 0 }
};
#endif

#ifndef _TMDLLIMPORT
#define _TMDLLIMPORT
#endif

_TMDLLIMPORT extern struct xa_switch_t tmnull_switch;

struct tmsvrargs_t tmsvrargs = {
		NULL,
		&_tmdsptchtbl[0],
		0,
		tpsvrinit,
		tpsvrdone,
		_tmrunserver,  /* PRIVATE  */
		NULL,		/* RESERVED */
		NULL,		/* RESERVED */
		NULL,		/* RESERVED */
		NULL		/* RESERVED */
};

struct tmsvrargs_t *
#ifdef _TMPROTOTYPES
_tmgetsvrargs(void)
#else
_tmgetsvrargs()
#endif
{
		tmsvrargs.xa_switch = &tmnull_switch;
		return(&tmsvrargs);
}

int
#ifdef _TMPROTOTYPES
main(int argc, char **argv)
#else
main(argc,argv)
int argc;
char **argv;
#endif
{
#ifdef TMMAINEXIT
#include "mainexit.h"
#endif

		return( _tmstartserver( argc, argv, _tmgetsvrargs()));
}
```

orafuncs.c

```
/*==========================================================
=========+
"
|		jpj. 1-18-98. modified to work with DELL tpc-c web client/tuxedo
"
|		server.
"
| last updated: 2-24-98. jpj.

+==========================================================
=========*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <process.h>
#include <time.h>
#include <windows.h>
#include "tpcc_ora.h"
#undef boolean
#include "tpcc_info_ora.h"
#include "tpccpl_ora.h"
#include "dpbcore_ora.h"
#include <userlog.h>
#include "tpcc_dell.h"
#include "trans_dell.h"

#define SQLTXT "alter session set isolation_level = serializable"
```

Appendix A – Application Source Code

```
#define SQL_TRACE_ON  "alter session set sql_trace = true"
#define SQL_TRACE_OFF "alter session set sql_trace = false"


char errmsg[80];
int proc_no = 0;
int logon = 0;
int new_init = 0;
int pay_init = 0;
int ord_init = 0;
int del_init = 0;
int sto_init = 0;

int execstatus;
int errcode;

OCIEnv *tpcenv;
OCIServer *tpcsrv;
OCIError *errhp;
OCISvcCtx *tpcsvc;
OCISession *tpcusr;
OCIStmt *curi;


ldadef tpclda;
csrdef curs;
csrdef curd;
csrdef curo0;
csrdef curo1;
csrdef curo2;
csrdef curp0;
csrdef curp1;

unsigned long tpchda[256];


/* for stock-level transaction */

int w_id;
int d_id;
int c_id;
int threshold;
int low_stock;

/* for delivery transaction */
int del_o_id[10];
int retries;

/* for order-status transaction */

int bylastname;
char c_last[17];
char c_first[17];
char c_middle[3];
double c_balance;
int o_id;
char o_entry_d[20];
int o_carrier_id;
int o_ol_cnt;
int ol_supply_w_id[15];
int ol_i_id[15];
int ol_quantity[15];
int ol_amount[15];
char ol_delivery_d[15][11];

/* for payment transaction */

int c_w_id;
int c_d_id;
int h_amount;
char w_street_1[21];
char w_street_2[21];
char w_city[21];
char w_state[3];
char w_zip[10];
char d_street_1[21];
char d_street_2[21];
char d_city[21];
char d_state[3];
char d_zip[10];
char c_street_1[21];
char c_street_2[21];
char c_city[21];
char c_state[3];
char c_zip[10];
char c_phone[17];
char c_since_d[11];
int c_discount;
char c_credit[3];
int c_credit_lim;
char c_data[201];
char h_date[20];

/* for new order transaction */

int nol_i_id[15];
int nol_supply_w_id[15];
int nol_quantity[15];
int nol_amount[15];
int o_all_local;
int w_tax;
int d_tax;
float total_amount;
char i_name[15][25];
int s_quantity[15];
char brand_gen[15];
int i_price[15];
int status;

unsigned char cr_date[7];
unsigned char c_since[7];
unsigned char o_entry_d_base[7];
unsigned char ol_d_base[15][7];
dvoid *xmem;

//              write all oci errors to user-defined tuxedo error log
extern void WriteErrorLog(char * tempstr);
extern CRITICAL_SECTION deliv_write_crit_sec;
extern char server[32];                                 // Name of database
server machine
extern char database_name[32];                          // Name of database
(REG_SZ: DatabaseName)
extern char database_user[32];                          // Database user login
name (REG_SZ: DatabaseUser)
extern char database_passwd[32];            // Database user login password


int ocierror(fname, lineno, errhp, status)
char *fname;
int lineno;
OCIError *errhp;
sword status;
{
 text tmpbuf[1024];
 text errbuf[512];
 sb4 errcode;

 switch (status) {
 case OCI_SUCCESS:
   break;
 case OCI_SUCCESS_WITH_INFO:

              (void) sprintf(tmpbuf,"Module %s Line %d\n", fname, lineno);
              WriteErrorLog((char *)tmpbuf);
   (void) sprintf(tmpbuf,"Error - OCI_SUCCESS_WITH_INFO\n");
              WriteErrorLog((char *)tmpbuf);
 break;
 case OCI_NEED_DATA:
   (void) sprintf(tmpbuf,"Module %s Line %d\n", fname, lineno);
              WriteErrorLog((char *)tmpbuf);
   (void) sprintf(tmpbuf,"Error - OCI_NEED_DATA\n");
              WriteErrorLog((char *)tmpbuf);
 break;
 case OCI_NO_DATA:
   (void) sprintf(tmpbuf,"Module %s Line %d\n", fname, lineno);
              WriteErrorLog((char *)tmpbuf);
   (void) sprintf(tmpbuf,"Error - OCI_NO_DATA\n");
              WriteErrorLog((char *)tmpbuf);
   return IRRECERR;
 break;
 case OCI_ERROR:
   (void) OCIErrorGet (errhp, (ub4) 1,
                                 (text *) NULL, &errcode, errbuf,
                                          (ub4) sizeof(errbuf),
OCI_HTYPE_ERROR);
    if (errcode == NOT_SERIALIZABLE) return (errcode);
   (void) sprintf(tmpbuf,"Module %s Line %d\n", fname, lineno);
              WriteErrorLog((char *)tmpbuf);
   (void) sprintf(tmpbuf,"Error - %s\n", errbuf);
              WriteErrorLog((char *)tmpbuf);
   return (errcode);
 break;
 case OCI_INVALID_HANDLE:
   (void) sprintf(tmpbuf,"Module %s Line %d\n", fname, lineno);
              WriteErrorLog((char *)tmpbuf);
   (void) sprintf(tmpbuf,"Error - OCI_INVALID_HANDLE\n");
              WriteErrorLog((char *)tmpbuf);
   TPCexit(1);
```

```
    exit(-1);
  break;
  case OCI_STILL_EXECUTING:
    (void) sprintf(tmpbuf,"Module %s Line %d\n", fname, lineno);
                WriteErrorLog((char *)tmpbuf);
    (void) sprintf(tmpbuf,"Error - OCI_STILL_EXECUTE\n");
                WriteErrorLog((char *)tmpbuf);
  break;
  case OCI_CONTINUE:
    (void) sprintf(tmpbuf,"Module %s Line %d\n", fname, lineno);
                WriteErrorLog((char *)tmpbuf);
    (void) sprintf(tmpbuf,"Error - OCI_CONTINUE\n");
                WriteErrorLog((char *)tmpbuf);
  break;
  default:
                WriteErrorLog("OCIERROR Default: Houston we have a
problem!!!");
                  break;
  }
  return RECOVERR;
}

int sqlfile(char *fnam, text *linebuf)
{
FILE *fd;
int nulpt=0;

HKEY        hKey;
DWORD       size;
DWORD       type;
char        szTmp[64];
char    szSQLPath[64];
BOOL    bReg = TRUE;

   strcpy  (szSQLPath, "c:\\tpcc\\sql\\");
   if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\\ORACLE\\tpcc", 0, KEY_READ,&hKey) !=
ERROR_SUCCESS )
        bReg = FALSE;

   size = sizeof(szTmp);
   if (bReg == TRUE)
      {
      if ( RegQueryValueEx(hKey, "SQL_PATH", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
                strcpy(szSQLPath, szTmp);
      }
   strcat(szSQLPath, fnam);

#ifdef  DEBUG
  fprintf(stderr, "sqlfile() fnam: %s,  linebuf: %#x\n", szSQLPath, linebuf);
#endif

    fd = fopen(szSQLPath,"r");
    if(fd == NULL)
    {
      sprintf(errmsg, "sqlfile: could not open %s\n", szSQLPath);
      WriteErrorLog(errmsg);
      return(1);
    }

    while (fgets((char *)(linebuf+nulpt), SQL_BUF_SIZE, fd))
    {
                nulpt = strlen((char *)linebuf);
    }
    return(0);
}

void vgetdate (unsigned char *oradt)
{
                struct tm *loctime;
                time_t  int_time;

                struct ORADATE {
                        unsigned char   century;
                        unsigned char   year;
                        unsigned char   month;
                        unsigned char   day;
                        unsigned char   hour;
                        unsigned char   minute;
                        unsigned char   second;
                } Date;
                int century;
                int cnvrtOK;

                /* assume convert is successful */
                cnvrtOK = 1;

                /* get the current date and time as an integer */
                time( &int_time);

                /* Convert the current date and time into local time */
                loctime = localtime( &int_time);

                century = (1900+loctime->tm_year) / 100;

                Date.century = (unsigned char)(century + 100);
                if (Date.century < 119 || Date.century > 120) cnvrtOK = 0;
                Date.year  = (unsigned char)(loctime->tm_year+100);
                if (Date.year < 100 || Date.year > 199) cnvrtOK = 0;
                Date.month = (unsigned char)(loctime->tm_mon + 1);
                if (Date.month < 1 || Date.month > 12) cnvrtOK = 0;
                Date.day   = (unsigned char)loctime->tm_mday;
                if (Date.day < 1 || Date.day > 31) cnvrtOK = 0;
                Date.hour  = (unsigned char)(loctime->tm_hour + 1);
                if (Date.hour < 1 || Date.hour > 24) cnvrtOK = 0;
                Date.minute= (unsigned char)(loctime->tm_min + 1);
                if (Date.minute < 1 || Date.minute > 60) cnvrtOK = 0;
                Date.second= (unsigned char)(loctime->tm_sec + 1);
                if (Date.second < 1 || Date.second > 60) cnvrtOK = 0;


                if (cnvrtOK)
                  memcpy(oradt,&Date,7);
                else
                  *oradt = '\0';

                return;

}
void cvtdmy (unsigned char *oradt, char *outdate)
{
                struct ORADATE {
                        unsigned char   century;
                        unsigned char   year;
                        unsigned char   month;
                        unsigned char   day;
                        unsigned char   hour;
                        unsigned char   minute;
                        unsigned char   second;
                } Date;

                int day,month,year;

                memcpy(&Date,oradt,7);
                year = (Date.century-100)*100 + Date.year-100;
                month = Date.month;
                day = Date.day;
                sprintf(outdate,"%02d-%02d-%4d\0",day,month,year);

                return;
}

void cvtdmyhms (unsigned char *oradt, char *outdate)
{
                struct ORADATE {
                        unsigned char   century;
                        unsigned char   year;
                        unsigned char   month;
                        unsigned char   day;
                        unsigned char   hour;
                        unsigned char   minute;
                        unsigned char   second;
                } Date;
                int day,month,year;
                int hour,min,sec;

                memcpy(&Date,oradt,7);
                year = (Date.century-100)*100 + Date.year-100;
                month = Date.month;
                day = Date.day;
                hour = Date.hour - 1;
                min = Date.minute - 1;
                sec = Date.second - 1;

                sprintf(outdate,"%02d-%02d-%4d %02d:%02d:%02d\0",
                                        day,month,year,hour,min,sec);

                return;
}

/* flag is non zero for batch driver case only */
int TPCexit (int flag)
{
  if (new_init) {
    plnewdone();
    new_init = 0;
  }
```

```
  if (pay_init) {
    plpaydone();
    pay_init = 0;
  }
  if (ord_init) {
    plorddone();
    ord_init = 0;
  }
  if (del_init) {
    pldeldone();
    del_init = 0;
  }
  if (sto_init) {
    plstodone();
    sto_init = 0;
  }

  if(flag) return(0);
  OCIHandleFree((dvoid *)tpcusr, OCI_HTYPE_SESSION);
  OCIHandleFree((dvoid *)tpcsvc, OCI_HTYPE_SVCCTX);
  OCIHandleFree((dvoid *)errhp, OCI_HTYPE_ERROR);
  OCIHandleFree((dvoid *)tpcsrv, OCI_HTYPE_SERVER);
  OCIHandleFree((dvoid *)tpcenv, OCI_HTYPE_ENV);

    logon = 0;


  return(0);
}


int TPCinit ()
{

  text stmbuf[100];

  OCIInitialize(OCI_DEFAULT,(dvoid *)0,0,0,0);
  OCIEnvInit(&tpcenv, OCI_DEFAULT, 0, (dvoid **)0);
  OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&tpcsrv, OCI_HTYPE_SERVER,
0 , (dvoid **)0);
  OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&errhp, OCI_HTYPE_ERROR, 0
, (dvoid **)0);
  OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&tpcsvc, OCI_HTYPE_SVCCTX,
0 , (dvoid **)0);
  OCIServerAttach(tpcsrv, errhp, (text *) server,0,OCI_DEFAULT);
  OCIAttrSet((dvoid *)tpcsvc, OCI_HTYPE_SVCCTX, (dvoid *)tpcsrv,
(ub4)0,OCI_ATTR_SERVER, errhp);
  OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&tpcusr, OCI_HTYPE_SESSION,
0 , (dvoid **)0);
  OCIAttrSet((dvoid *)tpcusr, OCI_HTYPE_SESSION, (dvoid *) database_user,
(ub4)strlen(database_user),OCI_ATTR_USERNAME, errhp);
  OCIAttrSet((dvoid *)tpcusr, OCI_HTYPE_SESSION, (dvoid *)
database_passwd, (ub4)strlen(database_passwd),
                    OCI_ATTR_PASSWORD, errhp);
  OCIERROR(errhp, OCISessionBegin(tpcsvc, errhp, tpcusr,
OCI_CRED_RDBMS, OCI_DEFAULT));
  OCIAttrSet(tpcsvc, OCI_HTYPE_SVCCTX, tpcusr, 0, OCI_ATTR_SESSION,
errhp);
  OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0,
(dvoid**)0);
  sprintf ((char *) stmbuf, SQLTXT);
  OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf),
OCI_NTV_SYNTAX, OCI_DEFAULT);
  OCIERROR(errhp,OCIStmtExecute(tpcsvc, curi,
errhp,1,0,0,0,OCI_DEFAULT));
  OCIHandleFree(curi, OCI_HTYPE_STMT);

  logon = 1;
  vgetdate(cr_date);


//                initialize neworder transaction
#ifdef _TUX_NO
  if (plnewinit () ) {
     TPCexit (0);
     userlog("TPCinit: New Order init failed");
     return (-1);
  }
  else
     new_init = 1;
#endif

//                initialize payment transaction
#ifdef _TUX_PY
  if (plpayinit ()) {
    TPCexit (0);
    userlog("TPCinit: Payment init failed");
    return (-1);
  }
  else
    pay_init = 1;
#endif

//                initialize order-status transaction
#ifdef _TUX_OS
  if (plordinit ()) {
    TPCexit (0);
    userlog("TPCinit: Order status init failed");
    return (-1);
  }
  else
    ord_init = 1;
#endif

//                initialize delivery transaction
#ifdef _TUX_D
  if (pldelinit ()) {
    TPCexit (0);
    userlog("TPCinit: Delivery init failed");
    return (-1);
  };
   del_init = 1;
   // Obtain frequency and initial count of 64-bit counter
   QueryPerformanceFrequency(&freq);
   freqd = (double) freq.QuadPart;
#endif

//                initialize stock transaction
#ifdef _TUX_SL
  if (plstoinit ()) {
    TPCexit (0);
    userlog("TPCinit: Stock level init failed");
    return (-1);
  }
  else
     sto_init = 1;
#endif

  // return to tuxedo successfull
  return (0);
}// end of TPCinit



#ifdef _TUX_NO
int TPCnew ( NEW_ORDER_DATA* str)
{
  int i;
  short rc;

  w_id = str->w_id;
  d_id = str->d_id;
  c_id = str->c_id;
  for (i = 0; i < 15; i++) {
    nol_i_id[i] = str->Ol[i].ol_i_id;
    nol_supply_w_id[i] = str->Ol[i].ol_supply_w_id;
    nol_quantity[i] = str->Ol[i].ol_quantity;
  }
  retries = 0;
  vgetdate(cr_date);

  if (rc = plnew ()) {
    userlog("plnew failed in orafuncs.c");
                if (rc != RECOVERR)
                  rc = IRRECERR;
    return (SQL_ERROR);
  }

  /* fill in date for o_entry_d from time in beginning of txn*/
  cvtdmyhms(cr_date,o_entry_d);

  rc = NOERR;
  str->o_id = o_id;
  str->o_ol_cnt = o_ol_cnt;
  strncpy (str->c_last, c_last, LAST_NAME_LEN+1);
  strncpy (str->c_credit, c_credit, CREDIT_LEN+1);
  str->c_discount = (float)c_discount/10000;
  str->w_tax = (float)w_tax/10000;
  str->d_tax = (float)d_tax/10000;
  strncpy (str->o_entry_d, o_entry_d, DATETIME_LEN+1);
  str->total_amount = total_amount;
  for (i = 0; i < o_ol_cnt; i++) {
    strncpy (str->Ol[i].ol_i_name, i_name[i], ITEM_NAME_LEN+1);
    str->Ol[i].ol_stock = s_quantity[i];
    str->Ol[i].ol_brand_generic = brand_gen[i];
    str->Ol[i].ol_i_price = (float)i_price[i]/100;
    str->Ol[i].ol_amount = (float)nol_amount[i]/100;
  }
```

```
  if (status){
    rc = INVALID_ITEM;
                userlog("plnew returned INVALID_ITEM in orafuncs.c");
                WriteErrorLog("plnew returned INVALID_ITEM in orafuncs.c");
  }
  else
     rc = SUCCESS;
  str->num_deadlocks = retries;
  return (rc);
}
#endif// end _TUX_NO

#ifdef _TUX_PY
int TPCpay ( PAYMENT_DATA* str)
{
  short rc;

  w_id = str->w_id;
  d_id = str->d_id;
  c_w_id = str->c_w_id;
  c_d_id = str->c_d_id;
  h_amount = (int) (str->h_amount * 100.00);
  vgetdate(cr_date);
  if (str->c_id !=0){
    c_id = str->c_id;
    strncpy (c_last, " ", LAST_NAME_LEN+1);
                bylastname = 0;
  }
  else {
    strncpy (c_last, str->c_last, LAST_NAME_LEN+1);
                bylastname = 1;
                c_id = 0;
  }
  retries = 0;

  if (rc = plpay ()) {
    userlog("plpay failed in orafuncs.c");
                if (rc != RECOVERR)
                   rc = IRRECERR;
    return (SQL_ERROR);
  }

  /* post process dates */
  cvtdmyhms(cr_date,h_date);
  cvtdmy(c_since,c_since_d);

  rc = NOERR;
  strncpy (str->w_street_1, w_street_1, ADDRESS_LEN+1);
  strncpy (str->w_street_2, w_street_2, ADDRESS_LEN+1);
  strncpy (str->w_city, w_city, ADDRESS_LEN+1);
  strncpy (str->w_state, w_state, STATE_LEN+1);
  strncpy (str->w_zip, w_zip, ZIP_LEN+1);
  strncpy (str->d_street_1, d_street_1, ADDRESS_LEN+1);
  strncpy (str->d_street_2, d_street_2, ADDRESS_LEN+1);
  strncpy (str->d_city, d_city, ADDRESS_LEN+1);
  strncpy (str->d_state, d_state, STATE_LEN+1);
  strncpy (str->d_zip, d_zip, ZIP_LEN+1);
  str->c_id = c_id;
  strncpy (str->c_first, c_first, FIRST_NAME_LEN+1);
  strncpy (str->c_middle, c_middle, MIDDLE_NAME_LEN+1);
  strncpy (str->c_last, c_last, LAST_NAME_LEN+1);
  strncpy (str->c_street_1, c_street_1, ADDRESS_LEN+1);
  strncpy (str->c_street_2, c_street_2, ADDRESS_LEN+1);
  strncpy (str->c_city, c_city, ADDRESS_LEN+1);
  strncpy (str->c_state, c_state, STATE_LEN+1);
  strncpy (str->c_zip, c_zip, ZIP_LEN+1);
  strncpy (str->c_phone, c_phone, PHONE_LEN+1);
  strncpy (str->c_since, c_since_d, DATE_LEN+1);
  strncpy (str->c_credit, c_credit, CREDIT_LEN+1);
  str->c_credit_lim = (float)c_credit_lim/100;
  str->c_discount = (float)c_discount/10000;
  str->c_balance = (float)c_balance/100;
  strncpy (str->c_data, c_data, CUST_DATA_LEN+1);
  strncpy (str->h_date, h_date, DATETIME_LEN+1);
  str->num_deadlocks = retries;
  return (SUCCESS);

}
#endif


#ifdef _TUX_OS
int TPCord ( ORDER_STATUS_DATA* str)
{
  int i;
  short rc;

  w_id = str->w_id;
  d_id = str->d_id;
  if (str->c_id != 0){
                c_id = str->c_id;
    strncpy (c_last, " ", LAST_NAME_LEN+1);
                bylastname = 0;
  }
  else {
    strncpy (c_last, str->c_last, LAST_NAME_LEN+1);
                c_id = 0;
    bylastname = 1;
  }


  retries = 0;

  if (rc = plord ()) {
                userlog("plord failed in orafuncs.c");
    if (rc != RECOVERR)
                   rc= IRRECERR;
    return (SQL_ERROR);
  }

  /* post process dates */
  cvtdmyhms(o_entry_d_base,o_entry_d);

  rc = NOERR;
  str->c_id = c_id;
  strncpy (str->c_last, c_last, LAST_NAME_LEN+1);
  strncpy (str->c_first, c_first, FIRST_NAME_LEN+1);
  strncpy (str->c_middle, c_middle, MIDDLE_NAME_LEN+1);
  str->c_balance = c_balance/100;
  str->o_id = o_id;
  strncpy (str->o_entry_d, o_entry_d, DATETIME_LEN+1);
   if ( o_carrier_id == 11 )
     str->o_carrier_id = 0;
  else
     str->o_carrier_id = o_carrier_id;
  str->o_ol_cnt = o_ol_cnt;
  for (i = 0; i < o_ol_cnt; i++) {
    ol_delivery_d[i][10] = '\0';
    if ( !strcmp(ol_delivery_d[i],"15-09-1911") )
                   strncpy(ol_delivery_d[i],"NOT DELIVR",10);
    str->OlOrderStatusData[i].ol_supply_w_id = ol_supply_w_id[i];
    str->OlOrderStatusData[i].ol_i_id = ol_i_id[i];
    str->OlOrderStatusData[i].ol_quantity = ol_quantity[i];
    str->OlOrderStatusData[i].ol_amount = (float)ol_amount[i]/100;
    strncpy (str->OlOrderStatusData[i].ol_delivery_d, ol_delivery_d[i], 11);
  }
  str->num_deadlocks = retries;
  return (SUCCESS);

}
#endif


#ifdef _TUX_D
int TPCdel ( DELIVERY_DATA* str)
{

                short rc;
                char
                printbuf[255];
                char                                                    d[10];
                char                                                    t[10];
                struct _timeb                   tb;
                LARGE_INTEGER                           tick_count;
                int
                current_time, elapsed_time;  // milliseconds since bootup


  w_id = (int) str->w_id;
  o_carrier_id = (int) str->o_carrier_id;
  retries = 0;
  vgetdate(cr_date);

  if (rc = pldel ()) {
    sprintf(errmsg, "DEL error return is %d\n", rc);
    WriteErrorLog(errmsg);// dell error log
                userlog(errmsg);// tuxedo error log
                userlog("pldel failed in orafuncs.c");
    if (rc != RECOVERR)
                   rc= IRRECERR;
    return (SQL_ERROR);
  }

                // Read current time in a few formats and write record to delivery
log
                // Each record:
                // today's date (mm/dd/yy), time now (hh:mm:ss.sss), queued_time
(msec since
```

```
                //    system bootup), time now (msec since system bootup), elapsed
time (msec),
                //    home warehouse, carrier, last o_id for district 1, ... last o_id for
                //    district 10
                _strdate(d);
                _strtime(t);
                _ftime(&tb);

                QueryPerformanceCounter(&tick_count);
                current_time = (int) floor(1000.0 *
((double)(tick_count.QuadPart))/freqd);
                elapsed_time = current_time - str->queued_time;

                sprintf(printbuf, "%s %s.%03u %d %d %d %d %d %d %d %d %d
%d %d %d %d %d\n",
                                d, t, tb.millitm, str->queued_time, current_time,
elapsed_time,
                                str->w_id, str->o_carrier_id,del_o_id[0],
del_o_id[1], del_o_id[2],
                                del_o_id[3], del_o_id[4],del_o_id[5], del_o_id[6],
del_o_id[7],
                                del_o_id[8], del_o_id[9]);

                EnterCriticalSection(&deliv_write_crit_sec);
                fprintf(fp_delivlog,printbuf);
                fflush(fp_delivlog);
                LeaveCriticalSection(&deliv_write_crit_sec);



  rc = NOERR;
  return (SUCCESS);

}
#endif// end _TUX_D


#ifdef _TUX_SL
int TPCsto ( STOCK_LEVEL_DATA* str)
{
  short rc;
  w_id = str->w_id;
  d_id = str->d_id;
  threshold = str->threshold;
  retries = 0;

  //            jpj. 1-19-98.
  //   passed return code back to calling funcition instead of in structure
  if (rc = plsto ()) {
    userlog("plsto failed in orafuncs.c");
                if (rc != RECOVERR)
                  rc = IRRECERR;
    return (SQL_ERROR);
  }

  rc = NOERR;
  str->low_stock = low_stock;
  str->num_deadlocks = retries;
  return (SUCCESS);

}
#endif//         end _TUX_SL
```

Commands For Compiling and Link Tuxedo Server

```
                cl.exe /nologo /ML /W3 /GX /Zi /O2  /I "d:\tuxedo\include" /I
"d:\orant\oci80\include" /I "c:\tpcc\tmserver"  /D "_ALL_TRANS" /D
"_TUX_NO" /D "_TUX_PY" /D "_TUX_OS" /D "_TUX_D" /D "_TUX_SL" /D
"ORACLE" /D "OCI" /D "ORA_NT" /D "WIN32" /D "_CONSOLE" /D
"_MBCS" /D "_TMSTHREADS" /D "_TUX" /Fp"c:\tpcc\tmserver\tmserver.pch"
/YX /Fo"c:\tpcc\tmserver\\" /Fd"c:\tpcc\tmserver\\" /FD /c
c:\tpcc\tmserver\orafuncs.c
orafuncs.c
                cl.exe /nologo /ML /W3 /GX /Zi /O2  /I "d:\tuxedo\include" /I
"d:\orant\oci80\include" /I "c:\tpcc\tmserver"  /D "_ALL_TRANS" /D
"_TUX_NO" /D "_TUX_PY" /D "_TUX_OS" /D "_TUX_D" /D "_TUX_SL" /D
"ORACLE" /D "OCI" /D "ORA_NT" /D "WIN32" /D "_CONSOLE" /D
"_MBCS" /D "_TMSTHREADS" /D "_TUX" /Fp"c:\tpcc\tmserver\tmserver.pch"
/YX /Fo"c:\tpcc\tmserver\\" /Fd"c:\tpcc\tmserver\\" /FD /c
c:\tpcc\tmserver\pldel_ora.c
pldel_ora.c
                cl.exe /nologo /ML /W3 /GX /Zi /O2  /I "d:\tuxedo\include" /I
"d:\orant\oci80\include" /I "c:\tpcc\tmserver"  /D "_ALL_TRANS" /D
"_TUX_NO" /D "_TUX_PY" /D "_TUX_OS" /D "_TUX_D" /D "_TUX_SL" /D
"ORACLE" /D "OCI" /D "ORA_NT" /D "WIN32" /D "_CONSOLE" /D
"_MBCS" /D "_TMSTHREADS" /D "_TUX" /Fp"c:\tpcc\tmserver\tmserver.pch"
/YX /Fo"c:\tpcc\tmserver\\" /Fd"c:\tpcc\tmserver\\" /FD /c
c:\tpcc\tmserver\plnew_ora.c
plnew_ora.c
                cl.exe /nologo /ML /W3 /GX /Zi /O2  /I "d:\tuxedo\include" /I
"d:\orant\oci80\include" /I "c:\tpcc\tmserver"  /D "_ALL_TRANS" /D
"_TUX_NO" /D "_TUX_PY" /D "_TUX_OS" /D "_TUX_D" /D "_TUX_SL" /D
"ORACLE" /D "OCI" /D "ORA_NT" /D "WIN32" /D "_CONSOLE" /D
"_MBCS" /D "_TMSTHREADS" /D "_TUX" /Fp"c:\tpcc\tmserver\tmserver.pch"
/YX /Fo"c:\tpcc\tmserver\\" /Fd"c:\tpcc\tmserver\\" /FD /c
c:\tpcc\tmserver\plord_ora.c
plord_ora.c
                cl.exe /nologo /ML /W3 /GX /Zi /O2  /I "d:\tuxedo\include" /I
"d:\orant\oci80\include" /I "c:\tpcc\tmserver"  /D "_ALL_TRANS" /D
"_TUX_NO" /D "_TUX_PY" /D "_TUX_OS" /D "_TUX_D" /D "_TUX_SL" /D
"ORACLE" /D "OCI" /D "ORA_NT" /D "WIN32" /D "_CONSOLE" /D
"_MBCS" /D "_TMSTHREADS" /D "_TUX" /Fp"c:\tpcc\tmserver\tmserver.pch"
/YX /Fo"c:\tpcc\tmserver\\" /Fd"c:\tpcc\tmserver\\" /FD /c
c:\tpcc\tmserver\plpay_ora.c
plpay_ora.c
                cl.exe /nologo /ML /W3 /GX /Zi /O2  /I "d:\tuxedo\include" /I
"d:\orant\oci80\include" /I "c:\tpcc\tmserver"  /D "_ALL_TRANS" /D
"_TUX_NO" /D "_TUX_PY" /D "_TUX_OS" /D "_TUX_D" /D "_TUX_SL" /D
"ORACLE" /D "OCI" /D "ORA_NT" /D "WIN32" /D "_CONSOLE" /D
"_MBCS" /D "_TMSTHREADS" /D "_TUX" /Fp"c:\tpcc\tmserver\tmserver.pch"
/YX /Fo"c:\tpcc\tmserver\\" /Fd"c:\tpcc\tmserver\\" /FD /c
c:\tpcc\tmserver\plsto_ora.c
plsto_ora.c
                cl.exe /nologo /ML /W3 /GX /Zi /O2  /I "d:\tuxedo\include" /I
"d:\orant\oci80\include" /I "c:\tpcc\tmserver"  /D "_ALL_TRANS" /D
"_TUX_NO" /D "_TUX_PY" /D "_TUX_OS" /D "_TUX_D" /D "_TUX_SL" /D
"ORACLE" /D "OCI" /D "ORA_NT" /D "WIN32" /D "_CONSOLE" /D
"_MBCS" /D "_TMSTHREADS" /D "_TUX" /Fp"c:\tpcc\tmserver\tmserver.pch"
/YX /Fo"c:\tpcc\tmserver\\" /Fd"c:\tpcc\tmserver\\" /FD /c
c:\tpcc\tmserver\tmserver_dell.c
tmserver_dell.c
                cl.exe /nologo /ML /W3 /GX /Zi /O2  /I "d:\tuxedo\include" /I
"d:\orant\oci80\include" /I "c:\tpcc\tmserver"  /D "_ALL_TRANS" /D
"_TUX_NO" /D "_TUX_PY" /D "_TUX_OS" /D "_TUX_D" /D "_TUX_SL" /D
"ORACLE" /D "OCI" /D "ORA_NT" /D "WIN32" /D "_CONSOLE" /D
"_MBCS" /D "_TMSTHREADS" /D "_TUX" /Fp"c:\tpcc\tmserver\tmserver.pch"
/YX /Fo"c:\tpcc\tmserver\\" /Fd"c:\tpcc\tmserver\\" /FD /c
c:\tpcc\tmserver\tmserver_stub_dell.c
tmserver_stub_dell.c
                link.exe kernel32.lib user32.lib advapi32.lib oci.lib ociw32.lib
libtux.lib ws2_32.lib mswsock.lib libbuft.lib libtux2.lib libgp.lib dpbnt.lib /nologo
/subsystem:console /pdb:none /machine:I386 /out:"TMALL.exe"
/libpath:"d:\tuxedo\lib" /libpath:"d:\orant\oci80\lib\msvc"
/libpath:"c:\tpcc\tmserver" "c:\tpcc\tmserver\orafuncs.obj"
"c:\tpcc\tmserver\pldel_ora.obj"  "c:\tpcc\tmserver\plnew_ora.obj"
"c:\tpcc\tmserver\plord_ora.obj"  "c:\tpcc\tmserver\plpay_ora.obj"
"c:\tpcc\tmserver\plsto_ora.obj"  "c:\tpcc\tmserver\tmserver_dell.obj"
"c:\tpcc\tmserver\tmserver_stub_dell.obj"
```

Appendix B – Database Design

Build Scripts and Loader Source Code

Addfile.sh

```
#
# $Header: addfile.sh 7030100.1 96/05/02 10:30:04 plai Generic<base> $ Copyr
(c) 1995 Oracle
#

#============================================================
========+
#       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#                   OPEN SYSTEMS PERFORMANCE GROUP                |
#                       All Rights Reserved                       |
#============================================================
========+
# FILENAME
#     addfile.sh
# DESCRIPTION
#     Add datafile to a tablespace.
# USAGE
#     addfile.sh <tablespace> <data file> <size>
#============================================================
========*/

FILE=`basename $2`

if [ -d ./outdir ]
then
  echo `date` > ./outdir/${FILE}.addf
fi

svrmgrl <<!
  set echo on
  connect internal
  alter tablespace $1 add datafile '$2' size $3 reuse;
  exit;
!

if [ -d ./outdir ]
then
  echo `date` >> ./outdir/${FILE}.addf
fi
```

Addlog.sh

```
#!/bin/ksh

typeset -Z2 q=1

while [ $q -le 16 ]
do
(
sleep 2
(
cat <<EOF1 | svrmgrl
connect internal
alter database add logfile thread $q
'?/dbs/tpcc_16sq_disks/log${q}_f1_m1' size 8000M reuse;
alter database add logfile member
'?/dbs/tpcc_16sq_disks/log${q}_f1_m2' reuse to
'?/dbs/tpcc_16sq_disks/log${q}_f1_m1';
exit;
EOF1
) &
sleep 5
(
cat <<EOF2 | svrmgrl
connect internal
alter database add logfile thread $q
'?/dbs/tpcc_16sq_disks/log${q}_f2_m1' size 8000M reuse;
alter database add logfile member
'?/dbs/tpcc_16sq_disks/log${q}_f2_m2' reuse to
'?/dbs/tpcc_16sq_disks/log${q}_f2_m1';
exit;
EOF2
) &
wait
cat <<EOF3 | svrmgrl
connect internal
alter database enable public thread $q;
exit
EOF3
) > /tmp/log$q.out 2>&1 &
/usr/local/bin/surun rqadmin -assign $! $q

(( q += 1 ))
done
```

Crts.sh

```
#
# $Header: addfile.sh 7030100.1 96/05/02 10:30:04 plai Generic<base> $ Copyr
(c) 1995 Oracle
#

#============================================================
========+
#       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#                   OPEN SYSTEMS PERFORMANCE GROUP                |
#                       All Rights Reserved                       |
#============================================================
========+
# FILENAME
#     crts.sh
# DESCRIPTION
#     Create the tablespace with the initial datafile
# USAGE
#     crts.sh <tablespace> <data file> <size>
#============================================================
========*/

FILE=`basename $2`

sleep 1

if [ -d ./outdir ]
then
  echo `date` > ./outdir/${FILE}.addf
fi

svrmgrl <<!
  set echo on
  connect internal
  create tablespace $1 datafile '$2' size $3 reuse;
  exit;
!

if [ -d ./outdir ]
then
  echo `date` >> ./outdir/${FILE}.addf
fi
```

Benchsetup.sh

```
#
# $Header: benchsetup.sh 7030100.2 96/05/16 17:59:17 plai Generic<base> $
Copyr (c) 1995 Oracle
#

#
#============================================================
========+
#       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#                   OPEN SYSTEMS PERFORMANCE GROUP                |
#                       All Rights Reserved                       |
#============================================================
========+
# FILENAME
#     benchsetup.sh
# DESCRIPTION
```

```
#     Usage: benchsetup.sh [options]
#           -mu <multiplier>     (# of warehouses)
#           -nd                  do not run benchdb.sh
#           -nt                  do not create tpcc tables
#           -nx                  do not create index for tpcc tables
#============================================================
========
#

export ORACLE_SID=TPCC1
surun rqadmin -assign $$ 81

print '\n\n\n\n\nStarting BENCHSETUP.SH\n\n\n\n\n'

BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BENCH_GEN/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BENCH_HOME/tpcc/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$TPCC_SCRIPTS/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
AUDIT_SCRIPTS=$BENCH_HOME/tpcc/audit/scripts
BUILD_SQL=sql
OUTDIR=outdir
MULT=8704

PATH=${PATH}:${TPCC_SOURCE}:${TPCC_UTILS}
export PATH

if echo "\c" | grep c >/dev/null 2>&1; then
   N='-n'
else
   C='\c'
fi
export N C

while [ "$#" != "0" ]
do
  case $1 in
     -mu) shift
          if [ "$1" != "" ]
          then
            MULT=$1
            shift
          fi
          ;;
     -nd) shift
          NO_DB="y"
          ;;
     -nt) shift
          NO_TAB="y"
          ;;
     -nx) shift
          NO_IND="y"
           ;;
      *) echo "Bag arg: $1"
          exit 1;
          ;;
  esac
done

if [ "$MULT" = "" ]
then
  echo $N "Database multiplier (# of warehouses)? [1]" $C
  read MULT
  if [ "$MULT" = "" ]
  then
     MULT=1
  fi
fi

if [ ! -d $OUTDIR ]
then
  mkdir $OUTDIR
fi

#
# Create database.
#

if [ "$NO_DB" = "" ]
then
  print '\n\n\n\n\nCreating database by calling BENCHDB.SH . . .\n\n\n\n\n'
  benchdb.sh  -n
  print '\n\n\n\n\nBENCHDB.SH script complete. Back in BENCHSETUP.SH
script.\n\n\n\n\n'
fi

switchlog.sh

#
#  Create tables.
#
 print '\n\n\n\n\nCreating tables and rollback segments . . .\n\n\n\n\n'

 if [ "$NO_TAB" = "" ]
 then
    print '\n\n\n\n\n Creating tables . . . \n\n\n\n\n'
    sqlplus system/manager @$BUILD_SQL/tpcc_tab
    print '\n\n\n\n\n Creating rollback segments . . .\n\n\n\n\n'
    sqlplus system/manager @$BUILD_SQL/tpcc_rol
 fi



#  Create customer and stock tables.
#

if [ "$NO_TAB" = "" ]
then
   print '\n\n\n   Creating CUSTOMER table . . .\n\n\n\n\n'
   sqlplus tpcc/tpcc @$BUILD_SQL/tpcc_tab2 > ${OUTDIR}/tab2.out 2>&1 &
   print '\n\n\n   Creating STOCK table . . .\n\n\n\n\n'
   sqlplus tpcc/tpcc @$BUILD_SQL/tpcc_tab3 > ${OUTDIR}/tab3.out 2>&1 &
fi

#
#  Load history, new-order, order, order-line tables
#

print '\n\n\n\n\n Calling PLOAD . . .\n\n\n\n\n'

pload.sh > ${OUTDIR}/pload.out 2>&1

switchlog.sh

wait


#
#  Load warehouse, district, item tables
#

print '\n\n\n\n\n Loading WAREHOUSE, DISTRICT and ITEM table data . .
.\n\n\n\n\n'
tpccload -M $MULT -w
tpccload -M $MULT -d
tpccload -M $MULT -i


#
#  Load customer table (in parallel with loading stock table)
#

ploadcust.sh > $OUTDIR/ploadcust.out 2>&1 &


#
#  Load stock table (in parallel with loading customer table)
#

ploadstock.sh > $OUTDIR/ploadstock.out 2>&1

wait

#
#
#

switchlog.sh

#
#  Create indexes
#

if [ "$NO_IND" = "" ]
then

sqlplus system/manager <<!
   set echo on;
   alter user tpcc temporary tablespace temp;
   quit;
!

svrmgrl <<!
   set echo on;
   connect internal
   alter tablespace temp
```

```
    default storage (initial 300M next 300M pctincrease 0
    maxextents unlimited );
  exit;
!


print '\n\n\n\n\n Creating INDEXES  . . .\n\n\n\n\n'

sqlplus tpcc/tpcc @$BUILD_SQL/tpcc_ix1
sqlplus tpcc/tpcc @$BUILD_SQL/tpcc_ix2

svrmgrl <<!
  set echo on;
  connect internal
  alter tablespace temp
    default storage (initial 20K next 20K pctincrease 50);
  exit;
!

fi

#
#  Analyze tables and indexes
#

print '\n\n\n\n\nAnalyzing tables and indexes . . .\n\n\n\n\n'
sqlplus tpcc/tpcc @$TPCC_SQL/tpcc_ana

#
#  Create table for processing benchmark results
#

print '\n\n\n\n\nCreating benchmark result tables . . .\n\n\n\n\n'
sqlplus sys/change_on_install @$GEN_SQL/orst_cre
sqlplus sys/change_on_install @$TPCC_SQL/c_stat
sqlplus sys/change_on_install @$GEN_SQL/pst_c

#
#  Create stored procedures
#

print '\n\n\n\n\nCreating stored procedures . . .\n\n\n\n\n'
sqlplus tpcc/tpcc @$TPCC_BLOCKS/views
sqlplus tpcc/tpcc @$TPCC_BLOCKS/pay

#
#  Create cache views
#

$TPCC_SCRIPTS/create_cache_views.sh

#
# Update the save_dfile table.
if [ -f dfile.map ]; then
      $TPCC_SCRIPTS/utils/dfile_init.sh
fi


#
#  Get some statistics
#

$TPCC_SCRIPTS/utils/ext_all.sh > ${OUTDIR}/ext_all.out 2>&1

$TPCC_SCRIPTS/utils/space_init.sh
$TPCC_SCRIPTS/utils/space_get.sh 95000 $MULT
$TPCC_SCRIPTS/utils/space_rpt.sh ${OUTDIR}/space.rpt

sqlplus sys/change_on_install <<!
  grant execute on dbms_lock to public;
  grant execute on dbms_pipe to public;
  grant select on v_\$parameter to public;
  quit;
!

sqlplus tpcc/tpcc @$AUDIT_SQL/plsql_mon
sqlplus tpcc/tpcc @$AUDIT_SQL/cre_tab

alter.sh

svrmgrl <<!
  set echo off;
  connect sys/change_on_install;
  @?/rdbms/admin/catparr;
  exit;
!

#$AUDIT_SCRIPTS/dbcheck.sh $MULT dbcheck.out.`date +%m%d`
#$AUDIT_SCRIPTS/dbtables.sh dbtables.out.`date +%m%d`
#$AUDIT_SCRIPTS/consist.sh -o consist.out.`date +%m%d` -m $MULT

#
#  Shutdown database
#

svrmgrl <<!
  connect internal;
  alter system switch logfile;
  alter system switch logfile;
  shutdown;
  exit;
!
```

benchdb.sh

```
#
# $Header: benchdb.sh 7030100.1 96/05/02 19:05:22 plai Generic<base> $ Copyr
(c) 1995 Oracle
#

#
#=========================================================
========+
#       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#                  OPEN SYSTEMS PERFORMANCE GROUP                |
#                       All Rights Reserved                      |
#=========================================================
========+
# FILENAME
#      benchdb.sh
# DESCRIPTION
#      Usage: benchdb.sh [options]
#            -n          do not create new tpcc database
#            -c          do not run catalog scripts
#=========================================================
========
#

/usr/local/bin/surun rqadmin -assign $$ 81
export ORACLE_SID=TPCC1
export SGA_QUAD_SET=0

print '\n\n\n\n\nStarting BENCHDB.SH\n\n\n\n\n'

BENCH_HOME=$ORACLE_HOME/bench/tpc
TPCC_ADMIN=admin

BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BENCH_GEN/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BENCH_HOME/tpcc/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$TPCC_SCRIPTS/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
AUDIT_SCRIPTS=$BENCH_HOME/tpcc/audit/scripts
BUILD_SQL=sql
OUTDIR=outdir
MULT=8704

PATH=${PATH}:${TPCC_SOURCE}:${TPCC_UTILS}:${TPCC_SCRIPTS}
export PATH
while [ "$#" != "0" ]
do
  case $1 in
     -n) shift
        NO_CREATE="y"
        ;;
     -c) shift
        NO_CAT="y"
        ;;
      *) echo "Bag arg: $1"
        exit 1;
        ;;
  esac
done


#Create database if NO_CREATE unset


if [ "$NO_CREATE" = "" ]
  then

  print '\n\n\n\n\nCreating database . . .\n\n\n\n\n'
```

```
svrmgrl <<!
                set echo on
                connect internal
                startup  pfile=$TPCC_ADMIN/p_create.ora  nomount
                create database tp16 controlfile reuse maxdatafiles 3000
maxinstances 16
                   maxlogfiles 40
                   datafile '?/dbs/tpcc_16sq_disks/system_1' size 750M reuse
                   logfile  '?/dbs/tpcc_16sq_disks/log_1' size 10100M reuse,
                                    '?/dbs/tpcc_16sq_disks/log_2' size 10100M
reuse;
                exit;
!

#
# Create more rollback segments
#

  sleep 5

  print '\nCreating additional rollback segments in SYSTEM . . .\n'

  svrmgrl <<!
    set echo on;
    connect system/manager
    alter tablespace system default storage ( maxextents unlimited );
    create rollback segment s1  storage (initial 200k minextents 2 next 200k);
    create rollback segment s2  storage (initial 200k minextents 2 next 200k);
    create rollback segment s3  storage (initial 200k minextents 2 next 200k);
    create rollback segment s4  storage (initial 200k minextents 2 next 200k);
    create rollback segment s5  storage (initial 200k minextents 2 next 200k);
    create rollback segment s6  storage (initial 200k minextents 2 next 200k);
    create rollback segment s7  storage (initial 200k minextents 2 next 200k);
    create rollback segment s8  storage (initial 200k minextents 2 next 200k);
    create rollback segment s9  storage (initial 200k minextents 2 next 200k);
    create rollback segment s10 storage (initial 200k minextents 2 next 200k);
    create rollback segment s11 storage (initial 200k minextents 2 next 200k);
    create rollback segment s12 storage (initial 200k minextents 2 next 200k);
    create rollback segment s13 storage (initial 200k minextents 2 next 200k);
    create rollback segment s14 storage (initial 200k minextents 2 next 200k);
    create rollback segment s15 storage (initial 200k minextents 2 next 200k);
    create rollback segment s16 storage (initial 200k minextents 2 next 200k);
    create rollback segment s17 storage (initial 200k minextents 2 next 200k);
    create rollback segment s18 storage (initial 200k minextents 2 next 200k);
    create rollback segment s19 storage (initial 200k minextents 2 next 200k);
    create rollback segment s20 storage (initial 200k minextents 2 next 200k);
    create rollback segment s21 storage (initial 200k minextents 2 next 200k);
    create rollback segment s22 storage (initial 200k minextents 2 next 200k);
    create rollback segment s23 storage (initial 200k minextents 2 next 200k);
    create rollback segment s24 storage (initial 200k minextents 2 next 200k);
    create rollback segment s25 storage (initial 200k minextents 2 next 200k);
    create rollback segment s26 storage (initial 200k minextents 2 next 200k);
    create rollback segment s27 storage (initial 200k minextents 2 next 200k);
    create rollback segment s28 storage (initial 200k minextents 2 next 200k);
    create rollback segment s29 storage (initial 200k minextents 2 next 200k);
    create rollback segment s30 storage (initial 200k minextents 2 next 200k);
    disconnect;
    connect internal;
    shutdown;
    exit;
!
 fi

#
#  Startup database with params file that includes new rollback segments
#

/etc/nopreempt svrmgrl <<!
                set echo on
        connect internal
        startup pfile=$TPCC_ADMIN/p_build.ora;
        connect system/manager
        exit;
!


#
#
# Create the tablespaces in parallel

                addfile.sh system  ?/dbs/tpcc_16sq_disks/system_2 750M &

                wait

print '\nInitalizing first datafiles in all tablespaces . . .\n'


                crts.sh ordl_1  ?/dbs/tpcc_16sq_disks/ordl_1 7674M &
                crts.sh ordl_2  ?/dbs/tpcc_16sq_disks/ordl_2 7674M &
                crts.sh ordl_3  ?/dbs/tpcc_16sq_disks/ordl_3 7674M &
                crts.sh ordl_4  ?/dbs/tpcc_16sq_disks/ordl_4 7674M &
                crts.sh ordl_5  ?/dbs/tpcc_16sq_disks/ordl_5 7674M &
                crts.sh ordl_6  ?/dbs/tpcc_16sq_disks/ordl_6 7674M &
                crts.sh ordl_7  ?/dbs/tpcc_16sq_disks/ordl_7 7674M &
                crts.sh ordl_8  ?/dbs/tpcc_16sq_disks/ordl_8 7674M &
                crts.sh ordl_9  ?/dbs/tpcc_16sq_disks/ordl_9 7674M &
                crts.sh ordl_10  ?/dbs/tpcc_16sq_disks/ordl_10 7674M &
                crts.sh ordl_11  ?/dbs/tpcc_16sq_disks/ordl_11 7674M &
                crts.sh ordl_12  ?/dbs/tpcc_16sq_disks/ordl_12 7674M &
                crts.sh ordl_13  ?/dbs/tpcc_16sq_disks/ordl_13 7674M &
                crts.sh ordl_14  ?/dbs/tpcc_16sq_disks/ordl_14 7674M &
                crts.sh ordl_15  ?/dbs/tpcc_16sq_disks/ordl_15 7674M &
                crts.sh ordl_16  ?/dbs/tpcc_16sq_disks/ordl_16 7674M &
                crts.sh ordl_17  ?/dbs/tpcc_16sq_disks/ordl_17 7674M &
                crts.sh ordl_18  ?/dbs/tpcc_16sq_disks/ordl_18 7674M &
                crts.sh ordl_19  ?/dbs/tpcc_16sq_disks/ordl_19 7674M &
                crts.sh ordl_20  ?/dbs/tpcc_16sq_disks/ordl_20 7674M &
                crts.sh ordl_21  ?/dbs/tpcc_16sq_disks/ordl_21 7674M &
                crts.sh ordl_22  ?/dbs/tpcc_16sq_disks/ordl_22 7674M &
                crts.sh ordl_23  ?/dbs/tpcc_16sq_disks/ordl_23 7674M &
                crts.sh ordl_24  ?/dbs/tpcc_16sq_disks/ordl_24 7674M &
                crts.sh ordl_25  ?/dbs/tpcc_16sq_disks/ordl_25 7674M &
                crts.sh ordl_26  ?/dbs/tpcc_16sq_disks/ordl_26 7674M &
                crts.sh ordl_27  ?/dbs/tpcc_16sq_disks/ordl_27 7674M &
                crts.sh ordl_28  ?/dbs/tpcc_16sq_disks/ordl_28 7674M &
                crts.sh ordl_29  ?/dbs/tpcc_16sq_disks/ordl_29 7674M &
                crts.sh ordl_30  ?/dbs/tpcc_16sq_disks/ordl_30 7674M &
                crts.sh ordl_31  ?/dbs/tpcc_16sq_disks/ordl_31 7674M &
                crts.sh ordl_32  ?/dbs/tpcc_16sq_disks/ordl_32 7674M &
                crts.sh ordl_33  ?/dbs/tpcc_16sq_disks/ordl_33 7674M &
                crts.sh ordl_34  ?/dbs/tpcc_16sq_disks/ordl_34 7674M &
                crts.sh ordl_35  ?/dbs/tpcc_16sq_disks/ordl_35 7674M &
                crts.sh ordl_36  ?/dbs/tpcc_16sq_disks/ordl_36 7674M &
                crts.sh ordl_37  ?/dbs/tpcc_16sq_disks/ordl_37 7674M &
                crts.sh ordl_38  ?/dbs/tpcc_16sq_disks/ordl_38 7674M &
                crts.sh ordl_39  ?/dbs/tpcc_16sq_disks/ordl_39 7674M &
                crts.sh ordl_40  ?/dbs/tpcc_16sq_disks/ordl_40 7674M &
                crts.sh ordl_41  ?/dbs/tpcc_16sq_disks/ordl_41 7674M &
                crts.sh ordl_42  ?/dbs/tpcc_16sq_disks/ordl_42 7674M &
                crts.sh ordl_43  ?/dbs/tpcc_16sq_disks/ordl_43 7674M &
                crts.sh ordl_44  ?/dbs/tpcc_16sq_disks/ordl_44 7674M &
                crts.sh ordl_45  ?/dbs/tpcc_16sq_disks/ordl_45 7674M &
                crts.sh ordl_46  ?/dbs/tpcc_16sq_disks/ordl_46 7674M &
                crts.sh ordl_47  ?/dbs/tpcc_16sq_disks/ordl_47 7674M &
                crts.sh ordl_48  ?/dbs/tpcc_16sq_disks/ordl_48 7674M &

                wait

                crts.sh iordl_1  ?/dbs/tpcc_16sq_disks/iordl_1 2165M &
                crts.sh iordl_2  ?/dbs/tpcc_16sq_disks/iordl_2 2165M &
                crts.sh iordl_3  ?/dbs/tpcc_16sq_disks/iordl_3 2165M &
                crts.sh iordl_4  ?/dbs/tpcc_16sq_disks/iordl_4 2165M &
                crts.sh iordl_5  ?/dbs/tpcc_16sq_disks/iordl_5 2165M &
                crts.sh iordl_6  ?/dbs/tpcc_16sq_disks/iordl_6 2165M &
                crts.sh iordl_7  ?/dbs/tpcc_16sq_disks/iordl_7 2165M &
                crts.sh iordl_8  ?/dbs/tpcc_16sq_disks/iordl_8 2165M &
                crts.sh iordl_9  ?/dbs/tpcc_16sq_disks/iordl_9 2165M &
                crts.sh iordl_10  ?/dbs/tpcc_16sq_disks/iordl_10 2165M &
                crts.sh iordl_11  ?/dbs/tpcc_16sq_disks/iordl_11 2165M &
                crts.sh iordl_12  ?/dbs/tpcc_16sq_disks/iordl_12 2165M &
                crts.sh iordl_13  ?/dbs/tpcc_16sq_disks/iordl_13 2165M &
                crts.sh iordl_14  ?/dbs/tpcc_16sq_disks/iordl_14 2165M &
                crts.sh iordl_15  ?/dbs/tpcc_16sq_disks/iordl_15 2165M &
                crts.sh iordl_16  ?/dbs/tpcc_16sq_disks/iordl_16 2165M &
                crts.sh iordl_17  ?/dbs/tpcc_16sq_disks/iordl_17 2165M &
                crts.sh iordl_18  ?/dbs/tpcc_16sq_disks/iordl_18 2165M &
                crts.sh iordl_19  ?/dbs/tpcc_16sq_disks/iordl_19 2165M &
                crts.sh iordl_20  ?/dbs/tpcc_16sq_disks/iordl_20 2165M &
                crts.sh iordl_21  ?/dbs/tpcc_16sq_disks/iordl_21 2165M &
                crts.sh iordl_22  ?/dbs/tpcc_16sq_disks/iordl_22 2165M &
                crts.sh iordl_23  ?/dbs/tpcc_16sq_disks/iordl_23 2165M &
                crts.sh iordl_24  ?/dbs/tpcc_16sq_disks/iordl_24 2165M &
                crts.sh iordl_25  ?/dbs/tpcc_16sq_disks/iordl_25 2165M &
                crts.sh iordl_26  ?/dbs/tpcc_16sq_disks/iordl_26 2165M &
                crts.sh iordl_27  ?/dbs/tpcc_16sq_disks/iordl_27 2165M &
                crts.sh iordl_28  ?/dbs/tpcc_16sq_disks/iordl_28 2165M &
                crts.sh iordl_29  ?/dbs/tpcc_16sq_disks/iordl_29 2165M &
                crts.sh iordl_30  ?/dbs/tpcc_16sq_disks/iordl_30 2165M &
                crts.sh iordl_31  ?/dbs/tpcc_16sq_disks/iordl_31 2165M &
                crts.sh iordl_32  ?/dbs/tpcc_16sq_disks/iordl_32 2165M &
                crts.sh iordl_33  ?/dbs/tpcc_16sq_disks/iordl_33 2165M &
                crts.sh iordl_34  ?/dbs/tpcc_16sq_disks/iordl_34 2165M &
                crts.sh iordl_35  ?/dbs/tpcc_16sq_disks/iordl_35 2165M &
                crts.sh iordl_36  ?/dbs/tpcc_16sq_disks/iordl_36 2165M &
                crts.sh iordl_37  ?/dbs/tpcc_16sq_disks/iordl_37 2165M &
                crts.sh iordl_38  ?/dbs/tpcc_16sq_disks/iordl_38 2165M &
                crts.sh iordl_39  ?/dbs/tpcc_16sq_disks/iordl_39 2165M &
                crts.sh iordl_40  ?/dbs/tpcc_16sq_disks/iordl_40 2165M &
                crts.sh iordl_41  ?/dbs/tpcc_16sq_disks/iordl_41 2165M &
                crts.sh iordl_42  ?/dbs/tpcc_16sq_disks/iordl_42 2165M &
                crts.sh iordl_43  ?/dbs/tpcc_16sq_disks/iordl_43 2165M &
```

```
crts.sh iordl_44  ?/dbs/tpcc_16sq_disks/iordl_44 2165M &
crts.sh iordl_45  ?/dbs/tpcc_16sq_disks/iordl_45 2165M &
crts.sh iordl_46  ?/dbs/tpcc_16sq_disks/iordl_46 2165M &
crts.sh iordl_47  ?/dbs/tpcc_16sq_disks/iordl_47 2165M &
crts.sh iordl_48  ?/dbs/tpcc_16sq_disks/iordl_48 2165M &

wait

crts.sh ord_1  ?/dbs/tpcc_16sq_disks/ord_1 597M &
crts.sh ord_2  ?/dbs/tpcc_16sq_disks/ord_2 597M &
crts.sh ord_3  ?/dbs/tpcc_16sq_disks/ord_3 597M &
crts.sh ord_4  ?/dbs/tpcc_16sq_disks/ord_4 597M &
crts.sh ord_5  ?/dbs/tpcc_16sq_disks/ord_5 597M &
crts.sh ord_6  ?/dbs/tpcc_16sq_disks/ord_6 597M &
crts.sh ord_7  ?/dbs/tpcc_16sq_disks/ord_7 597M &
crts.sh ord_8  ?/dbs/tpcc_16sq_disks/ord_8 597M &
crts.sh ord_9  ?/dbs/tpcc_16sq_disks/ord_9 597M &
crts.sh ord_10  ?/dbs/tpcc_16sq_disks/ord_10 597M &
crts.sh ord_11  ?/dbs/tpcc_16sq_disks/ord_11 597M &
crts.sh ord_12  ?/dbs/tpcc_16sq_disks/ord_12 597M &
crts.sh ord_13  ?/dbs/tpcc_16sq_disks/ord_13 597M &
crts.sh ord_14  ?/dbs/tpcc_16sq_disks/ord_14 597M &
crts.sh ord_15  ?/dbs/tpcc_16sq_disks/ord_15 597M &
crts.sh ord_16  ?/dbs/tpcc_16sq_disks/ord_16 597M &
crts.sh ord_17  ?/dbs/tpcc_16sq_disks/ord_17 597M &
crts.sh ord_18  ?/dbs/tpcc_16sq_disks/ord_18 597M &
crts.sh ord_19  ?/dbs/tpcc_16sq_disks/ord_19 597M &
crts.sh ord_20  ?/dbs/tpcc_16sq_disks/ord_20 597M &
crts.sh ord_21  ?/dbs/tpcc_16sq_disks/ord_21 597M &
crts.sh ord_22  ?/dbs/tpcc_16sq_disks/ord_22 597M &
crts.sh ord_23  ?/dbs/tpcc_16sq_disks/ord_23 597M &
crts.sh ord_24  ?/dbs/tpcc_16sq_disks/ord_24 597M &
crts.sh ord_25  ?/dbs/tpcc_16sq_disks/ord_25 597M &
crts.sh ord_26  ?/dbs/tpcc_16sq_disks/ord_26 597M &
crts.sh ord_27  ?/dbs/tpcc_16sq_disks/ord_27 597M &
crts.sh ord_28  ?/dbs/tpcc_16sq_disks/ord_28 597M &
crts.sh ord_29  ?/dbs/tpcc_16sq_disks/ord_29 597M &
crts.sh ord_30  ?/dbs/tpcc_16sq_disks/ord_30 597M &
crts.sh ord_31  ?/dbs/tpcc_16sq_disks/ord_31 597M &
crts.sh ord_32  ?/dbs/tpcc_16sq_disks/ord_32 597M &
crts.sh ord_33  ?/dbs/tpcc_16sq_disks/ord_33 597M &
crts.sh ord_34  ?/dbs/tpcc_16sq_disks/ord_34 597M &
crts.sh ord_35  ?/dbs/tpcc_16sq_disks/ord_35 597M &
crts.sh ord_36  ?/dbs/tpcc_16sq_disks/ord_36 597M &
crts.sh ord_37  ?/dbs/tpcc_16sq_disks/ord_37 597M &
crts.sh ord_38  ?/dbs/tpcc_16sq_disks/ord_38 597M &
crts.sh ord_39  ?/dbs/tpcc_16sq_disks/ord_39 597M &
crts.sh ord_40  ?/dbs/tpcc_16sq_disks/ord_40 597M &
crts.sh ord_41  ?/dbs/tpcc_16sq_disks/ord_41 597M &
crts.sh ord_42  ?/dbs/tpcc_16sq_disks/ord_42 597M &
crts.sh ord_43  ?/dbs/tpcc_16sq_disks/ord_43 597M &
crts.sh ord_44  ?/dbs/tpcc_16sq_disks/ord_44 597M &
crts.sh ord_45  ?/dbs/tpcc_16sq_disks/ord_45 597M &
crts.sh ord_46  ?/dbs/tpcc_16sq_disks/ord_46 597M &
crts.sh ord_47  ?/dbs/tpcc_16sq_disks/ord_47 597M &
crts.sh ord_48  ?/dbs/tpcc_16sq_disks/ord_48 597M &

wait

crts.sh iord2_1  ?/dbs/tpcc_16sq_disks/iord2_1 334M &
crts.sh iord2_2  ?/dbs/tpcc_16sq_disks/iord2_2 334M &
crts.sh iord2_3  ?/dbs/tpcc_16sq_disks/iord2_3 334M &
crts.sh iord2_4  ?/dbs/tpcc_16sq_disks/iord2_4 334M &
crts.sh iord2_5  ?/dbs/tpcc_16sq_disks/iord2_5 334M &
crts.sh iord2_6  ?/dbs/tpcc_16sq_disks/iord2_6 334M &
crts.sh iord2_7  ?/dbs/tpcc_16sq_disks/iord2_7 334M &
crts.sh iord2_8  ?/dbs/tpcc_16sq_disks/iord2_8 334M &
crts.sh iord2_9  ?/dbs/tpcc_16sq_disks/iord2_9 334M &
crts.sh iord2_10  ?/dbs/tpcc_16sq_disks/iord2_10 334M &
crts.sh iord2_11  ?/dbs/tpcc_16sq_disks/iord2_11 334M &
crts.sh iord2_12  ?/dbs/tpcc_16sq_disks/iord2_12 334M &
crts.sh iord2_13  ?/dbs/tpcc_16sq_disks/iord2_13 334M &
crts.sh iord2_14  ?/dbs/tpcc_16sq_disks/iord2_14 334M &
crts.sh iord2_15  ?/dbs/tpcc_16sq_disks/iord2_15 334M &
crts.sh iord2_16  ?/dbs/tpcc_16sq_disks/iord2_16 334M &
crts.sh iord2_17  ?/dbs/tpcc_16sq_disks/iord2_17 334M &
crts.sh iord2_18  ?/dbs/tpcc_16sq_disks/iord2_18 334M &
crts.sh iord2_19  ?/dbs/tpcc_16sq_disks/iord2_19 334M &
crts.sh iord2_20  ?/dbs/tpcc_16sq_disks/iord2_20 334M &
crts.sh iord2_21  ?/dbs/tpcc_16sq_disks/iord2_21 334M &
crts.sh iord2_22  ?/dbs/tpcc_16sq_disks/iord2_22 334M &
crts.sh iord2_23  ?/dbs/tpcc_16sq_disks/iord2_23 334M &
crts.sh iord2_24  ?/dbs/tpcc_16sq_disks/iord2_24 334M &
crts.sh iord2_25  ?/dbs/tpcc_16sq_disks/iord2_25 334M &
crts.sh iord2_26  ?/dbs/tpcc_16sq_disks/iord2_26 334M &
crts.sh iord2_27  ?/dbs/tpcc_16sq_disks/iord2_27 334M &
crts.sh iord2_28  ?/dbs/tpcc_16sq_disks/iord2_28 334M &
crts.sh iord2_29  ?/dbs/tpcc_16sq_disks/iord2_29 334M &
crts.sh iord2_30  ?/dbs/tpcc_16sq_disks/iord2_30 334M &
crts.sh iord2_31  ?/dbs/tpcc_16sq_disks/iord2_31 334M &
crts.sh iord2_32  ?/dbs/tpcc_16sq_disks/iord2_32 334M &
crts.sh iord2_33  ?/dbs/tpcc_16sq_disks/iord2_33 334M &
crts.sh iord2_34  ?/dbs/tpcc_16sq_disks/iord2_34 334M &
crts.sh iord2_35  ?/dbs/tpcc_16sq_disks/iord2_35 334M &
crts.sh iord2_36  ?/dbs/tpcc_16sq_disks/iord2_36 334M &
crts.sh iord2_37  ?/dbs/tpcc_16sq_disks/iord2_37 334M &
crts.sh iord2_38  ?/dbs/tpcc_16sq_disks/iord2_38 334M &
crts.sh iord2_39  ?/dbs/tpcc_16sq_disks/iord2_39 334M &
crts.sh iord2_40  ?/dbs/tpcc_16sq_disks/iord2_40 334M &
crts.sh iord2_41  ?/dbs/tpcc_16sq_disks/iord2_41 334M &
crts.sh iord2_42  ?/dbs/tpcc_16sq_disks/iord2_42 334M &
crts.sh iord2_43  ?/dbs/tpcc_16sq_disks/iord2_43 334M &
crts.sh iord2_44  ?/dbs/tpcc_16sq_disks/iord2_44 334M &
crts.sh iord2_45  ?/dbs/tpcc_16sq_disks/iord2_45 334M &
crts.sh iord2_46  ?/dbs/tpcc_16sq_disks/iord2_46 334M &
crts.sh iord2_47  ?/dbs/tpcc_16sq_disks/iord2_47 334M &
crts.sh iord2_48  ?/dbs/tpcc_16sq_disks/iord2_48 334M &

wait

crts.sh iord1_1  ?/dbs/tpcc_16sq_disks/iord1_1 215M &
crts.sh iord1_2  ?/dbs/tpcc_16sq_disks/iord1_2 215M &
crts.sh iord1_3  ?/dbs/tpcc_16sq_disks/iord1_3 215M &
crts.sh iord1_4  ?/dbs/tpcc_16sq_disks/iord1_4 215M &
crts.sh iord1_5  ?/dbs/tpcc_16sq_disks/iord1_5 215M &
crts.sh iord1_6  ?/dbs/tpcc_16sq_disks/iord1_6 215M &
crts.sh iord1_7  ?/dbs/tpcc_16sq_disks/iord1_7 215M &
crts.sh iord1_8  ?/dbs/tpcc_16sq_disks/iord1_8 215M &
crts.sh iord1_9  ?/dbs/tpcc_16sq_disks/iord1_9 215M &
crts.sh iord1_10  ?/dbs/tpcc_16sq_disks/iord1_10 215M &
crts.sh iord1_11  ?/dbs/tpcc_16sq_disks/iord1_11 215M &
crts.sh iord1_12  ?/dbs/tpcc_16sq_disks/iord1_12 215M &
crts.sh iord1_13  ?/dbs/tpcc_16sq_disks/iord1_13 215M &
crts.sh iord1_14  ?/dbs/tpcc_16sq_disks/iord1_14 215M &
crts.sh iord1_15  ?/dbs/tpcc_16sq_disks/iord1_15 215M &
crts.sh iord1_16  ?/dbs/tpcc_16sq_disks/iord1_16 215M &
crts.sh iord1_17  ?/dbs/tpcc_16sq_disks/iord1_17 215M &
crts.sh iord1_18  ?/dbs/tpcc_16sq_disks/iord1_18 215M &
crts.sh iord1_19  ?/dbs/tpcc_16sq_disks/iord1_19 215M &
crts.sh iord1_20  ?/dbs/tpcc_16sq_disks/iord1_20 215M &
crts.sh iord1_21  ?/dbs/tpcc_16sq_disks/iord1_21 215M &
crts.sh iord1_22  ?/dbs/tpcc_16sq_disks/iord1_22 215M &
crts.sh iord1_23  ?/dbs/tpcc_16sq_disks/iord1_23 215M &
crts.sh iord1_24  ?/dbs/tpcc_16sq_disks/iord1_24 215M &
crts.sh iord1_25  ?/dbs/tpcc_16sq_disks/iord1_25 215M &
crts.sh iord1_26  ?/dbs/tpcc_16sq_disks/iord1_26 215M &
crts.sh iord1_27  ?/dbs/tpcc_16sq_disks/iord1_27 215M &
crts.sh iord1_28  ?/dbs/tpcc_16sq_disks/iord1_28 215M &
crts.sh iord1_29  ?/dbs/tpcc_16sq_disks/iord1_29 215M &
crts.sh iord1_30  ?/dbs/tpcc_16sq_disks/iord1_30 215M &
crts.sh iord1_31  ?/dbs/tpcc_16sq_disks/iord1_31 215M &
crts.sh iord1_32  ?/dbs/tpcc_16sq_disks/iord1_32 215M &
crts.sh iord1_33  ?/dbs/tpcc_16sq_disks/iord1_33 215M &
crts.sh iord1_34  ?/dbs/tpcc_16sq_disks/iord1_34 215M &
crts.sh iord1_35  ?/dbs/tpcc_16sq_disks/iord1_35 215M &
crts.sh iord1_36  ?/dbs/tpcc_16sq_disks/iord1_36 215M &
crts.sh iord1_37  ?/dbs/tpcc_16sq_disks/iord1_37 215M &
crts.sh iord1_38  ?/dbs/tpcc_16sq_disks/iord1_38 215M &
crts.sh iord1_39  ?/dbs/tpcc_16sq_disks/iord1_39 215M &
crts.sh iord1_40  ?/dbs/tpcc_16sq_disks/iord1_40 215M &
crts.sh iord1_41  ?/dbs/tpcc_16sq_disks/iord1_41 215M &
crts.sh iord1_42  ?/dbs/tpcc_16sq_disks/iord1_42 215M &
crts.sh iord1_43  ?/dbs/tpcc_16sq_disks/iord1_43 215M &
crts.sh iord1_44  ?/dbs/tpcc_16sq_disks/iord1_44 215M &
crts.sh iord1_45  ?/dbs/tpcc_16sq_disks/iord1_45 215M &
crts.sh iord1_46  ?/dbs/tpcc_16sq_disks/iord1_46 215M &
crts.sh iord1_47  ?/dbs/tpcc_16sq_disks/iord1_47 215M &
crts.sh iord1_48  ?/dbs/tpcc_16sq_disks/iord1_48 215M &

wait

crts.sh nord_1  ?/dbs/tpcc_16sq_disks/nord_1 658M &
crts.sh nord_2  ?/dbs/tpcc_16sq_disks/nord_2 658M &
crts.sh nord_3  ?/dbs/tpcc_16sq_disks/nord_3 658M &
crts.sh nord_4  ?/dbs/tpcc_16sq_disks/nord_4 658M &
crts.sh nord_5  ?/dbs/tpcc_16sq_disks/nord_5 658M &
crts.sh nord_6  ?/dbs/tpcc_16sq_disks/nord_6 658M &
crts.sh nord_7  ?/dbs/tpcc_16sq_disks/nord_7 658M &
crts.sh nord_8  ?/dbs/tpcc_16sq_disks/nord_8 658M &
crts.sh nord_9  ?/dbs/tpcc_16sq_disks/nord_9 658M &
crts.sh nord_10  ?/dbs/tpcc_16sq_disks/nord_10 658M &
crts.sh nord_11  ?/dbs/tpcc_16sq_disks/nord_11 658M &
crts.sh nord_12  ?/dbs/tpcc_16sq_disks/nord_12 658M &
crts.sh nord_13  ?/dbs/tpcc_16sq_disks/nord_13 658M &
crts.sh nord_14  ?/dbs/tpcc_16sq_disks/nord_14 658M &
crts.sh nord_15  ?/dbs/tpcc_16sq_disks/nord_15 658M &
crts.sh nord_16  ?/dbs/tpcc_16sq_disks/nord_16 658M &


crts.sh inord_1  ?/dbs/tpcc_16sq_disks/inord_1 422M &
crts.sh inord_2  ?/dbs/tpcc_16sq_disks/inord_2 422M &
crts.sh inord_3  ?/dbs/tpcc_16sq_disks/inord_3 422M &
```

```
crts.sh inord_4  ?/dbs/tpcc_16sq_disks/inord_4 422M &
crts.sh inord_5  ?/dbs/tpcc_16sq_disks/inord_5 422M &
crts.sh inord_6  ?/dbs/tpcc_16sq_disks/inord_6 422M &
crts.sh inord_7  ?/dbs/tpcc_16sq_disks/inord_7 422M &
crts.sh inord_8  ?/dbs/tpcc_16sq_disks/inord_8 422M &
crts.sh inord_9  ?/dbs/tpcc_16sq_disks/inord_9 422M &
crts.sh inord_10  ?/dbs/tpcc_16sq_disks/inord_10 422M &
crts.sh inord_11  ?/dbs/tpcc_16sq_disks/inord_11 422M &
crts.sh inord_12  ?/dbs/tpcc_16sq_disks/inord_12 422M &
crts.sh inord_13  ?/dbs/tpcc_16sq_disks/inord_13 422M &
crts.sh inord_14  ?/dbs/tpcc_16sq_disks/inord_14 422M &
crts.sh inord_15  ?/dbs/tpcc_16sq_disks/inord_15 422M &
crts.sh inord_16  ?/dbs/tpcc_16sq_disks/inord_16 422M &


crts.sh hist_1  ?/dbs/tpcc_16sq_disks/hist_1 1417M &
crts.sh hist_2  ?/dbs/tpcc_16sq_disks/hist_2 1417M &
crts.sh hist_3  ?/dbs/tpcc_16sq_disks/hist_3 1417M &
crts.sh hist_4  ?/dbs/tpcc_16sq_disks/hist_4 1417M &
crts.sh hist_5  ?/dbs/tpcc_16sq_disks/hist_5 1417M &
crts.sh hist_6  ?/dbs/tpcc_16sq_disks/hist_6 1417M &
crts.sh hist_7  ?/dbs/tpcc_16sq_disks/hist_7 1417M &
crts.sh hist_8  ?/dbs/tpcc_16sq_disks/hist_8 1417M &
crts.sh hist_9  ?/dbs/tpcc_16sq_disks/hist_9 1417M &
crts.sh hist_10  ?/dbs/tpcc_16sq_disks/hist_10 1417M &
crts.sh hist_11  ?/dbs/tpcc_16sq_disks/hist_11 1417M &
crts.sh hist_12  ?/dbs/tpcc_16sq_disks/hist_12 1417M &
crts.sh hist_13  ?/dbs/tpcc_16sq_disks/hist_13 1417M &
crts.sh hist_14  ?/dbs/tpcc_16sq_disks/hist_14 1417M &
crts.sh hist_15  ?/dbs/tpcc_16sq_disks/hist_15 1417M &
crts.sh hist_16  ?/dbs/tpcc_16sq_disks/hist_16 1417M &

wait

crts.sh roll_1  ?/dbs/tpcc_16sq_disks/roll_1 290M &
crts.sh roll_2  ?/dbs/tpcc_16sq_disks/roll_2 290M &
crts.sh roll_3  ?/dbs/tpcc_16sq_disks/roll_3 290M &
crts.sh roll_4  ?/dbs/tpcc_16sq_disks/roll_4 290M &
crts.sh roll_5  ?/dbs/tpcc_16sq_disks/roll_5 290M &
crts.sh roll_6  ?/dbs/tpcc_16sq_disks/roll_6 290M &
crts.sh roll_7  ?/dbs/tpcc_16sq_disks/roll_7 290M &
crts.sh roll_8  ?/dbs/tpcc_16sq_disks/roll_8 290M &
crts.sh roll_9  ?/dbs/tpcc_16sq_disks/roll_9 290M &
crts.sh roll_10  ?/dbs/tpcc_16sq_disks/roll_10 290M &
crts.sh roll_11  ?/dbs/tpcc_16sq_disks/roll_11 290M &
crts.sh roll_12  ?/dbs/tpcc_16sq_disks/roll_12 290M &
crts.sh roll_13  ?/dbs/tpcc_16sq_disks/roll_13 290M &
crts.sh roll_14  ?/dbs/tpcc_16sq_disks/roll_14 290M &
crts.sh roll_15  ?/dbs/tpcc_16sq_disks/roll_15 290M &
crts.sh roll_16  ?/dbs/tpcc_16sq_disks/roll_16 290M &


crts.sh ware_1  ?/dbs/tpcc_16sq_disks/ware_1 39M &
crts.sh ware_2  ?/dbs/tpcc_16sq_disks/ware_2 39M &
crts.sh ware_3  ?/dbs/tpcc_16sq_disks/ware_3 39M &
crts.sh ware_4  ?/dbs/tpcc_16sq_disks/ware_4 39M &
crts.sh ware_5  ?/dbs/tpcc_16sq_disks/ware_5 39M &
crts.sh ware_6  ?/dbs/tpcc_16sq_disks/ware_6 39M &
crts.sh ware_7  ?/dbs/tpcc_16sq_disks/ware_7 39M &
crts.sh ware_8  ?/dbs/tpcc_16sq_disks/ware_8 39M &
crts.sh ware_9  ?/dbs/tpcc_16sq_disks/ware_9 39M &
crts.sh ware_10  ?/dbs/tpcc_16sq_disks/ware_10 39M &
crts.sh ware_11  ?/dbs/tpcc_16sq_disks/ware_11 39M &
crts.sh ware_12  ?/dbs/tpcc_16sq_disks/ware_12 39M &
crts.sh ware_13  ?/dbs/tpcc_16sq_disks/ware_13 39M &
crts.sh ware_14  ?/dbs/tpcc_16sq_disks/ware_14 39M &
crts.sh ware_15  ?/dbs/tpcc_16sq_disks/ware_15 39M &
crts.sh ware_16  ?/dbs/tpcc_16sq_disks/ware_16 39M &

wait

crts.sh icust2  ?/dbs/tpcc_16sq_disks/icust2_1 4723M &
crts.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_1 515M &
crts.sh istk  ?/dbs/tpcc_16sq_disks/istk_1 1415M &
crts.sh items  ?/dbs/tpcc_16sq_disks/items_1 9M &

wait

addfile.sh icust2  ?/dbs/tpcc_16sq_disks/icust2_2 4723M &
addfile.sh icust2  ?/dbs/tpcc_16sq_disks/icust2_3 4723M &
addfile.sh icust2  ?/dbs/tpcc_16sq_disks/icust2_4 4723M &


addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_2 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_3 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_4 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_5 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_6 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_7 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_8 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_9 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_10 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_11 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_12 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_13 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_14 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_15 515M &
addfile.sh icust1  ?/dbs/tpcc_16sq_disks/icust1_16 515M &


addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_2 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_3 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_4 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_5 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_6 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_7 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_8 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_9 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_10 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_11 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_12 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_13 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_14 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_15 1415M &
addfile.sh istk  ?/dbs/tpcc_16sq_disks/istk_16 1415M &


addfile.sh items  ?/dbs/tpcc_16sq_disks/items_2 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_3 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_4 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_5 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_6 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_7 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_8 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_9 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_10 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_11 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_12 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_13 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_14 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_15 9M &
addfile.sh items  ?/dbs/tpcc_16sq_disks/items_16 9M &

wait

crts.sh cust  ?/dbs/tpcc_16sq_disks/cust_1 1121M

addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_2 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_3 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_4 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_5 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_6 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_7 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_8 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_9 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_10 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_11 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_12 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_13 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_14 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_15 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_16 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_17 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_18 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_19 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_20 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_21 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_22 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_23 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_24 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_25 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_26 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_27 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_28 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_29 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_30 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_31 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_32 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_33 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_34 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_35 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_36 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_37 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_38 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_39 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_40 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_41 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_42 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_43 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_44 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_45 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_46 1121M &
```

```
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_47 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_48 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_49 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_50 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_51 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_52 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_53 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_54 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_55 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_56 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_57 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_58 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_59 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_60 1121M &

wait

addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_61 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_62 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_63 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_64 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_65 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_66 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_67 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_68 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_69 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_70 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_71 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_72 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_73 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_74 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_75 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_76 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_77 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_78 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_79 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_80 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_81 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_82 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_83 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_84 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_85 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_86 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_87 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_88 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_89 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_90 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_91 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_92 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_93 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_94 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_95 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_96 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_97 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_98 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_99 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_100 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_101 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_102 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_103 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_104 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_105 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_106 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_107 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_108 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_109 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_110 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_111 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_112 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_113 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_114 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_115 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_116 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_117 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_118 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_119 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_120 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_121 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_122 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_123 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_124 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_125 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_126 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_127 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_128 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_129 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_130 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_131 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_132 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_133 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_134 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_135 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_136 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_137 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_138 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_139 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_140 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_141 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_142 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_143 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_144 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_145 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_146 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_147 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_148 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_149 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_150 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_151 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_152 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_153 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_154 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_155 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_156 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_157 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_158 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_159 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_160 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_161 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_162 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_163 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_164 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_165 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_166 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_167 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_168 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_169 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_170 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_171 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_172 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_173 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_174 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_175 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_176 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_177 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_178 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_179 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_180 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_181 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_182 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_183 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_184 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_185 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_186 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_187 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_188 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_189 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_190 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_191 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_192 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_193 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_194 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_195 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_196 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_197 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_198 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_199 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_200 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_201 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_202 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_203 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_204 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_205 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_206 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_207 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_208 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_209 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_210 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_211 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_212 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_213 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_214 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_215 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_216 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_217 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_218 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_219 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_220 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_221 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_222 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_223 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_224 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_225 1121M &
```

```
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_226 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_227 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_228 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_229 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_230 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_231 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_232 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_233 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_234 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_235 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_236 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_237 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_238 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_239 1121M &
addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_240 1121M &

wait

addfile.sh cust  ?/dbs/tpcc_16sq_disks/cust_241 1121M &

wait

crts.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_1 401M
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_2 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_3 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_4 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_5 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_6 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_7 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_8 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_9 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_10 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_11 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_12 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_13 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_14 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_15 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_16 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_17 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_18 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_19 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_20 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_21 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_22 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_23 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_24 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_25 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_26 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_27 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_28 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_29 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_30 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_31 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_32 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_33 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_34 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_35 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_36 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_37 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_38 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_39 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_40 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_41 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_42 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_43 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_44 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_45 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_46 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_47 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_48 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_49 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_50 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_51 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_52 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_53 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_54 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_55 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_56 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_57 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_58 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_59 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_60 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_61 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_62 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_63 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_64 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_65 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_66 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_67 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_68 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_69 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_70 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_71 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_72 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_73 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_74 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_75 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_76 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_77 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_78 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_79 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_80 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_81 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_82 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_83 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_84 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_85 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_86 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_87 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_88 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_89 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_90 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_91 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_92 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_93 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_94 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_95 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_96 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_97 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_98 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_99 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_100 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_101 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_102 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_103 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_104 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_105 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_106 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_107 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_108 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_109 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_110 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_111 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_112 401M &

wait

addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_113 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_114 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_115 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_116 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_117 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_118 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_119 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_120 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_121 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_122 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_123 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_124 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_125 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_126 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_127 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_128 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_129 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_130 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_131 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_132 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_133 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_134 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_135 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_136 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_137 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_138 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_139 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_140 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_141 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_142 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_143 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_144 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_145 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_146 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_147 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_148 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_149 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_150 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_151 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_152 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_153 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_154 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_155 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_156 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_157 401M &
```

```
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_158 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_159 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_160 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_161 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_162 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_163 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_164 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_165 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_166 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_167 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_168 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_169 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_170 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_171 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_172 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_173 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_174 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_175 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_176 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_177 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_178 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_179 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_180 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_181 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_182 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_183 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_184 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_185 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_186 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_187 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_188 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_189 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_190 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_191 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_192 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_193 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_194 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_195 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_196 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_197 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_198 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_199 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_200 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_201 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_202 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_203 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_204 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_205 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_206 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_207 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_208 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_209 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_210 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_211 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_212 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_213 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_214 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_215 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_216 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_217 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_218 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_219 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_220 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_221 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_222 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_223 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_224 401M &

wait

addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_225 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_226 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_227 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_228 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_229 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_230 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_231 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_232 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_233 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_234 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_235 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_236 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_237 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_238 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_239 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_240 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_241 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_242 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_243 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_244 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_245 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_246 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_247 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_248 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_249 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_250 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_251 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_252 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_253 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_254 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_255 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_256 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_257 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_258 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_259 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_260 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_261 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_262 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_263 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_264 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_265 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_266 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_267 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_268 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_269 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_270 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_271 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_272 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_273 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_274 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_275 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_276 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_277 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_278 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_279 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_280 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_281 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_282 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_283 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_284 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_285 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_286 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_287 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_288 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_289 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_290 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_291 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_292 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_293 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_294 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_295 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_296 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_297 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_298 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_299 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_300 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_301 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_302 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_303 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_304 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_305 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_306 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_307 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_308 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_309 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_310 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_311 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_312 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_313 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_314 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_315 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_316 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_317 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_318 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_319 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_320 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_321 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_322 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_323 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_324 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_325 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_326 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_327 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_328 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_329 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_330 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_331 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_332 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_333 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_334 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_335 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_336 401M &
```

```
wait

addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_337 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_338 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_339 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_340 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_341 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_342 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_343 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_344 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_345 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_346 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_347 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_348 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_349 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_350 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_351 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_352 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_353 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_354 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_355 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_356 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_357 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_358 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_359 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_360 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_361 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_362 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_363 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_364 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_365 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_366 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_367 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_368 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_369 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_370 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_371 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_372 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_373 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_374 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_375 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_376 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_377 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_378 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_379 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_380 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_381 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_382 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_383 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_384 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_385 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_386 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_387 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_388 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_389 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_390 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_391 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_392 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_393 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_394 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_395 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_396 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_397 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_398 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_399 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_400 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_401 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_402 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_403 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_404 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_405 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_406 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_407 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_408 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_409 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_410 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_411 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_412 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_413 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_414 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_415 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_416 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_417 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_418 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_419 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_420 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_421 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_422 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_423 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_424 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_425 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_426 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_427 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_428 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_429 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_430 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_431 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_432 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_433 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_434 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_435 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_436 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_437 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_438 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_439 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_440 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_441 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_442 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_443 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_444 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_445 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_446 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_447 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_448 401M &

wait

addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_449 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_450 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_451 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_452 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_453 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_454 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_455 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_456 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_457 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_458 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_459 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_460 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_461 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_462 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_463 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_464 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_465 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_466 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_467 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_468 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_469 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_470 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_471 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_472 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_473 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_474 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_475 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_476 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_477 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_478 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_479 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_480 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_481 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_482 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_483 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_484 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_485 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_486 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_487 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_488 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_489 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_490 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_491 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_492 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_493 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_494 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_495 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_496 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_497 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_498 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_499 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_500 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_501 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_502 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_503 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_504 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_505 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_506 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_507 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_508 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_509 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_510 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_511 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_512 401M &
```

```
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_513 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_514 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_515 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_516 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_517 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_518 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_519 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_520 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_521 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_522 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_523 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_524 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_525 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_526 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_527 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_528 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_529 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_530 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_531 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_532 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_533 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_534 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_535 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_536 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_537 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_538 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_539 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_540 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_541 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_542 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_543 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_544 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_545 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_546 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_547 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_548 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_549 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_550 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_551 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_552 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_553 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_554 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_555 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_556 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_557 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_558 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_559 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_560 401M &

wait

addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_561 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_562 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_563 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_564 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_565 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_566 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_567 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_568 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_569 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_570 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_571 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_572 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_573 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_574 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_575 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_576 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_577 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_578 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_579 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_580 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_581 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_582 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_583 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_584 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_585 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_586 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_587 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_588 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_589 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_590 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_591 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_592 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_593 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_594 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_595 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_596 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_597 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_598 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_599 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_600 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_601 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_602 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_603 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_604 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_605 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_606 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_607 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_608 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_609 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_610 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_611 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_612 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_613 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_614 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_615 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_616 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_617 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_618 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_619 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_620 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_621 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_622 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_623 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_624 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_625 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_626 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_627 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_628 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_629 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_630 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_631 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_632 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_633 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_634 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_635 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_636 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_637 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_638 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_639 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_640 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_641 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_642 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_643 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_644 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_645 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_646 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_647 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_648 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_649 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_650 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_651 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_652 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_653 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_654 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_655 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_656 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_657 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_658 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_659 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_660 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_661 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_662 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_663 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_664 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_665 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_666 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_667 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_668 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_669 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_670 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_671 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_672 401M &

wait

addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_673 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_674 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_675 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_676 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_677 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_678 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_679 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_680 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_681 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_682 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_683 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_684 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_685 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_686 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_687 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_688 401M &
```

```
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_689 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_690 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_691 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_692 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_693 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_694 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_695 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_696 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_697 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_698 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_699 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_700 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_701 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_702 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_703 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_704 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_705 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_706 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_707 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_708 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_709 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_710 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_711 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_712 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_713 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_714 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_715 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_716 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_717 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_718 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_719 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_720 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_721 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_722 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_723 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_724 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_725 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_726 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_727 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_728 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_729 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_730 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_731 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_732 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_733 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_734 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_735 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_736 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_737 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_738 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_739 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_740 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_741 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_742 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_743 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_744 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_745 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_746 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_747 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_748 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_749 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_750 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_751 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_752 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_753 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_754 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_755 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_756 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_757 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_758 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_759 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_760 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_761 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_762 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_763 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_764 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_765 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_766 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_767 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_768 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_769 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_770 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_771 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_772 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_773 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_774 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_775 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_776 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_777 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_778 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_779 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_780 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_781 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_782 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_783 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_784 401M &

wait

addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_785 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_786 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_787 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_788 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_789 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_790 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_791 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_792 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_793 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_794 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_795 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_796 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_797 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_798 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_799 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_800 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_801 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_802 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_803 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_804 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_805 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_806 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_807 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_808 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_809 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_810 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_811 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_812 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_813 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_814 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_815 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_816 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_817 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_818 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_819 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_820 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_821 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_822 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_823 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_824 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_825 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_826 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_827 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_828 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_829 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_830 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_831 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_832 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_833 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_834 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_835 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_836 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_837 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_838 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_839 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_840 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_841 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_842 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_843 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_844 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_845 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_846 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_847 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_848 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_849 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_850 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_851 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_852 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_853 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_854 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_855 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_856 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_857 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_858 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_859 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_860 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_861 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_862 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_863 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_864 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_865 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_866 401M &
addfile.sh stocks ?/dbs/tpcc_16sq_disks/stocks_867 401M &
```

```
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_868 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_869 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_870 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_871 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_872 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_873 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_874 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_875 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_876 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_877 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_878 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_879 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_880 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_881 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_882 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_883 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_884 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_885 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_886 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_887 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_888 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_889 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_890 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_891 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_892 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_893 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_894 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_895 401M &
addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_896 401M &

wait

addfile.sh stocks  ?/dbs/tpcc_16sq_disks/stocks_897 401M &

wait
```

```
crts.sh temp  ?/dbs/tpcc_16sq_disks/temp_1_1 7000M
wait
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_2_1 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_3_1 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_4_1 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_5_1 7000M &


addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_1_2 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_2_2 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_3_2 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_4_2 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_5_2 7000M &


addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_1_3 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_2_3 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_3_3 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_4_3 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_5_3 7000M &

wait

addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_1_4 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_2_4 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_3_4 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_4_4 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_5_4 7000M &


addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_1_5 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_2_5 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_3_5 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_4_5 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_5_5 7000M &


addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_1_6 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_2_6 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_3_6 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_4_6 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_5_6 7000M &

wait

addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_1_7 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_2_7 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_3_7 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_4_7 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_5_7 7000M &


addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_1_8 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_2_8 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_3_8 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_4_8 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_5_8 7000M &


addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_1_9 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_2_9 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_3_9 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_4_9 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_5_9 7000M &

wait

addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_1_10 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_2_10 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_3_10 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_4_10 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_5_10 7000M &

addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_1_11 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_2_11 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_3_11 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_4_11 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_5_11 7000M &

addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_1_12 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_2_12 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_3_12 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_4_12 7000M &
addfile.sh temp  ?/dbs/tpcc_16sq_disks/temp_5_12 7000M &

wait
```

```
# run catalog if NO_CAT unset

print '\nRunning catalogs \n'

if [ "$NO_CAT" = "" ]
then
svrmgrl <<!
        set echo off;
        connect sys/change_on_install;
        @?/rdbms/admin/catalog;
        @?/rdbms/admin/catproc;
        @?/rdbms/admin/catparr;
        @?/rdbms/admin/catperf;
        connect system/manager;
        @?/rdbms/admin/catdbsyn;
        @?/sqlplus/admin/pupbld;
        exit;
!
fi
```

pload.sh

```
#!/bin/ksh
#
# $Header: pload.sh 7030100.1 96/05/02 19:06:06 plai Generic<base> $ Copyr (c)
1995 Oracle
#
#
#=========================================================
========+
#       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#                  OPEN SYSTEMS PERFORMANCE GROUP                |
#                     All Rights Reserved                        |
#=========================================================
========+
# FILENAME
#     pload.sh
# DESCRIPTION
#     Usage: pload.sh [options]
#           -mu <multiplier>     (# of warehouses)
#=========================================================
========
#

BENCH_HOME=$ORACLE_HOME/bench/tpc
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
```

```
TPCC_LOADER=$BENCH_HOME/tpcc/loader
TPCC_ADMIN=admin

LDIR=data
OUTDIR=outdir
MULT=8704

PATH=${PATH}:$TPCC_SOURCE:$TPCC_SCRIPTS
export PATH

if echo "\c" | grep c >/dev/null 2>&1; then
   N='-n'
else
   C='\c'
fi
export N C

while [ "$#" != "0" ]
do
  case $1 in
     -mu) shift
          if [ "$1" != "" ]
          then
            MULT=$1
            shift
          fi
          ;;
       *) echo "Bag arg: $1"
          exit 1;
          ;;
  esac
done

if [ "$MULT" = "" ]
then
  echo $N "Database multiplier (# of warehouses)? [1]" $C
  read MULT
  if [ "$MULT" = "" ]
  then
     MULT=1
  fi
fi

if [ ! -d $LDIR ]
then
  mkdir $LDIR
fi

if [ ! -d $OUTDIR ]
then
  mkdir $OUTDIR
fi


for TAB in hist neword order
do

                case $TAB in
                        hist) TABLE="HISTORY"; FL="-h" ;;
                        neword) TABLE="NEW_ORDER" ; FL="-n" ;;
                        order) TABLE="ORDERS" ;;
                esac

                date
                print "Starting $TABLE data load . . ."

                #
                # Get the partition info

                getpartinfo.sh $TABLE $MULT > $TAB.partinfo

                NPARTS=$( wc -l $TAB.partinfo )

                I=1
                while [ $I -le $NPARTS ]
                do
                  /etc/mknod ${LDIR}/${TAB}${I}.dat p
                  I=`expr $I + 1`
                done

                if [ $TABLE = "ORDERS" ]; then
                        I=1
                        while [ $I -le $NPARTS ]
                        do
                          /etc/mknod ${LDIR}/ordline${I}.dat p
                          I=`expr $I + 1`
                        done
                fi

                print "     TPCCLOAD phase . . ."

                I=1
                while [ $I -le $NPARTS ]
                do
                  read SW EW
                  if [ $TABLE = "ORDERS" ]; then
                   tpccload -M $MULT -o ${LDIR}/ordline${I}.dat -g -b $SW -e
$EW > \
                    ${LDIR}/order${I}.dat 2> ${OUTDIR}/order${I}.out &
                  else
                   tpccload -M $MULT $FL -g -b $SW -e $EW >
${LDIR}/${TAB}${I}.dat 2> \
                     ${OUTDIR}/${TAB}${I}.out &
                  fi
                  I=`expr $I + 1`
                done < $TAB.partinfo

                sleep 10

                print '\n\n\n\n\n     sqlldrst phase:'
                print '          no. partitions:          '$NPARTS';'
                print '          warehouses:          '$MULT';\n\n\n\n\n\n'

                I=1
                while [ $I -le $NPARTS ]
                do
                  if [ $TABLE = "ORDERS" ]; then
                    sqlldrst tpcc/tpcc control=$TPCC_LOADER/orderP${I}.ctl \
                                log=${OUTDIR}/order${I}.log \
                                bad=${OUTDIR}/order${I}.bad
data=${LDIR}/order${I}.dat \
                                discard=${OUTDIR}/order${I}.dsc &

                    sqlldrst tpcc/tpcc control=$TPCC_LOADER/ordlineP${I}.ctl \
                                log=${OUTDIR}/ordline${I}.log \
                                bad=${OUTDIR}/ordline${I}.bad
data=${LDIR}/ordline${I}.dat \
                                discard=${OUTDIR}/ordline${I}.dsc &
                  else
                    sqlldrst tpcc/tpcc
control=$TPCC_LOADER/${TAB}P${I}.ctl \
                                log=${OUTDIR}/${TAB}${I}.log \
                                bad=${OUTDIR}/${TAB}${I}.bad
data=${LDIR}/${TAB}${I}.dat \
                                discard=${OUTDIR}/${TAB}${I}.dsc &
                  fi
                  I=`expr $I + 1`
                done

                wait

                date
                print "     $TABLE load DONE!"
                I=1
                while [ $I -le $NPARTS ]
                do
                  rm -f ${LDIR}/${TAB}${I}.dat
                  I=`expr $I + 1`
                done

done

I=1
while [ $I -le $NPARTS ]
do
  rm -f ${LDIR}/ordline${I}.dat
  I=`expr $I + 1`
done


print '\n\n\nPLOAD script completed'
```

ploadstock.sh

```
#
# $Header: pload.sh 7030100.1 96/05/02 19:06:06 plai Generic<base> $ Copyr (c)
1995 Oracle
#

#
#=========================================================
========+
#       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#                  OPEN SYSTEMS PERFORMANCE GROUP                |
#                     All Rights Reserved                        |
```

```
#============================================================
========+
# FILENAME
#      pload.sh
# DESCRIPTION
#      Usage: pload.sh [options]
#           -mu <multiplier>    (# of warehouses)
#============================================================
========
#

BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BENCH_GEN/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BENCH_HOME/tpcc/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$TPCC_SCRIPTS/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BENCH_HOME/tpcc/loader

LDIR=data
OUTDIR=outdir
MULT=8704
NUM_SPLIT=10

#T=echo

PATH=${PATH}:$TPCC_SOURCE
export PATH

if echo "\c" | grep c >/dev/null 2>&1; then
   N='-n'
else
   C='\c'
fi
export N C


if [ ! -d $LDIR ]
then
  mkdir $LDIR
fi

if [ ! -d $OUTDIR ]
then
  mkdir $OUTDIR
fi


split_part()
{

SW=$1
EW=$2
NP=$3

(( ADD = ( $EW - $SW + 1 ) / $NP ))
(( REM = ( $EW - $SW + 1 ) - ( $ADD * $NP ) ))

I=$SW
L=1
while [ $I -lt $(( $EW - ADD )) ]
do
                if [ $L -le $(( $NP - $REM )) ]; then
                (( J = $I + $ADD - 1 ))
                else
                (( J = $I + $ADD  ))
                fi
                (( L += 1 ))
                echo $I $J
                (( I = $J + 1 ))
done
echo $I $EW
}

#
#  Load Stock table
#

print `date +%T`' Starting STOCK data load . . .'

#
# Get the partition info

getpartinfo.sh WAREHOUSE $MULT  > ware.partinfo

NPARTS=$(cat ware.partinfo | wc -l )

I=1
while [ $I -le $(( $NPARTS * $NUM_SPLIT )) ]
do
  /etc/mknod ${LDIR}/stock${I}.dat p
  I=`expr $I + 1`
done


print '     TPCCLOAD phase . . .'

                I=1;J=1
                while [ $I -le $NPARTS ]
                do
                  read SW EW
                  split_part $SW $EW $NUM_SPLIT | while read SW EW
                  do
                                (
                                sleep 1
                                $T tpccload -M $MULT -s -g -b $SW -e $EW > \
                                  ${LDIR}/stock${J}.dat 2>
${OUTDIR}/stock${J}.out
                                ) &

                                $T /usr/local/bin/surun rqadmin -assign $! $I
                                J=`expr $J + 1`
                  done
                  I=`expr $I + 1`
                done < ware.partinfo

                sleep 10

                print '\n\n\n\n\n     sqlldr phase:'
                print '          no. partitions:          '$NPARTS';'
                print '          warehouses:          '$MULT';\n\n\n\n\n\n'

                I=1;J=1
                while [ $I -le $NPARTS ]
                do
                   while [ $J -le $(( $I * $NUM_SPLIT )) ]
                   do
                   (
                        sleep 1
                     $T export ORACLE_SID=TPCCX$I
                     $T sqlldr tpcc/tpcc control=$TPCC_LOADER/stock.ctl \
                                log=${OUTDIR}/stock${J}.log \
                                bad=${OUTDIR}/stock${J}.bad \
                                data=${LDIR}/stock${J}.dat \
                                discard=${OUTDIR}/stock${J}.dsc
                   ) &
                     $T /usr/local/bin/surun rqadmin -assign $! $I
                       J=`expr $J + 1`
                   done
                   I=`expr $I + 1`
                done

                wait

                I=1
                while [ $I -le $(( $NPARTS * $NUM_SPLIT )) ]
                do
                  rm -f ${LDIR}/stock${I}.dat
                  I=`expr $I + 1`
                done


print ' '
print `date +%T`' End STOCK data load . . .'
```

ploadcust.sh

```
#
# $Header: pload.sh 7030100.1 96/05/02 19:06:06 plai Generic<base> $ Copyr (c)
1995 Oracle
#

#
#============================================================
========+
#       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#                  OPEN SYSTEMS PERFORMANCE GROUP                |
#                       All Rights Reserved                      |
```

```
#============================================================
========+
# FILENAME
#      pload.sh
# DESCRIPTION
#      Usage: pload.sh [options]
#           -mu <multiplier>     (# of warehouses)
#============================================================
========
#

BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BENCH_GEN/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BENCH_HOME/tpcc/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$TPCC_SCRIPTS/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BENCH_HOME/tpcc/loader

LDIR=data
OUTDIR=outdir
MULT=8704
NUM_SPLIT=8


#T=echo

PATH=${PATH}:$TPCC_SOURCE
export PATH

if echo "\c" | grep c >/dev/null 2>&1; then
   N='-n'
else
   C='\c'
fi
export N C


if [ ! -d $LDIR ]
then
  mkdir $LDIR
fi

if [ ! -d $OUTDIR ]
then
  mkdir $OUTDIR
fi


split_part()
{

SW=$1
EW=$2
NP=$3

(( ADD = ( $EW - $SW + 1 ) / $NP ))
(( REM = ( $EW - $SW + 1 ) - ( $ADD * $NP ) ))

I=$SW
L=1
while [ $I -lt $(( $EW - ADD )) ]
do
                if [ $L -le $(( $NP - $REM )) ]; then
                (( J = $I + $ADD - 1 ))
                else
                (( J = $I + $ADD  ))
                fi
                (( L += 1 ))
                echo $I $J
                (( I = $J + 1 ))
done
echo $I $EW
}


#
#  Load cust table
#

print `date +%T`' Starting CUST data load . . .'

#
# Get the partition info

getpartinfo.sh WAREHOUSE $MULT  > ware.partinfo.C

NPARTS=$(cat ware.partinfo.C | wc -l )

                I=1;J=1
                while [ $I -le $NPARTS ]
                do
                  read SW EW
                  split_part $SW $EW $NUM_SPLIT | while read SW EW
                  do
                                (
                                sleep 1
                        $T export ORACLE_SID=TPCCX$I
                                $T tpccload -M $MULT -c -b $SW -e $EW > \
                                  ${OUTDIR}/cust${J}.out 2>&1
                                ) &

                                $T /usr/local/bin/surun rqadmin -assign $! $I
                                J=`expr $J + 1`
                  done
                  I=`expr $I + 1`
                done < ware.partinfo.C

                wait

print ' '
print `date +%T`' End CUST data load . . .'
```

tpcc_tab.sql

```
rem
rem
============================================================
========+
rem        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
rem                OPEN SYSTEMS PERFORMANCE GROUP                   |
rem                   All Rights Reserved                           |
rem
============================================================
========+
rem  FILENAME
rem       tpcc_tab.sql
rem  DESCRIPTION
rem       Create tables for TPC-C database.
rem
============================================================
========
rem

rem
rem FIRST, create TPCC userid and connect to it.
rem
       grant connect,resource,unlimited tablespace to tpcc identified by tpcc;
       alter user tpcc temporary tablespace temp;
                connect tpcc/tpcc

rem
rem NEXT, DROP all first
rem
rem             drop cluster icluster including tables;
rem             drop table warehouse;
rem             drop table district;
rem             drop table history;
rem             drop table orders;
rem             drop table new_order;
rem             drop table order_line;
rem             drop table item;

set timing on
set echo on;

rem
rem LAST, CREATE all tables
rem

      create table warehouse (
            w_id           number,
            w_ytd          number(12),
            w_tax          number(4),
            w_name         varchar2(10),
            w_street_1     varchar2(20),
            w_street_2     varchar2(20),
            w_city         varchar2(20),
            w_state        char(2),
            w_zip          char(9)
```

```
        )
        partition by range (w_id)
        (
partition ware_P1 values less than (545) tablespace ware_1,
partition ware_P2 values less than (1089) tablespace ware_2,
partition ware_P3 values less than (1633) tablespace ware_3,
partition ware_P4 values less than (2177) tablespace ware_4,
partition ware_P5 values less than (2721) tablespace ware_5,
partition ware_P6 values less than (3265) tablespace ware_6,
partition ware_P7 values less than (3809) tablespace ware_7,
partition ware_P8 values less than (4353) tablespace ware_8,
partition ware_P9 values less than (4897) tablespace ware_9,
partition ware_P10 values less than (5441) tablespace ware_10,
partition ware_P11 values less than (5985) tablespace ware_11,
partition ware_P12 values less than (6529) tablespace ware_12,
partition ware_P13 values less than (7073) tablespace ware_13,
partition ware_P14 values less than (7617) tablespace ware_14,
partition ware_P15 values less than (8161) tablespace ware_15,
partition ware_P16 values less than (MAXVALUE) tablespace ware_16
        )
        initrans 4
        pctfree 95 pctused 4
        storage (initial 1000K next 40K pctincrease 0);


    create table district (
        d_id            number,
        d_w_id          number,
        d_ytd           number(12),
        d_tax           number(4),
        d_next_o_id     number,
        d_name          varchar2(10),
        d_street_1      varchar2(20),
        d_street_2      varchar2(20),
        d_city          varchar2(20),
        d_state         char(2),
        d_zip           char(9)
    )
    partition by range (d_w_id)
    (
partition dist_P1 values less than (545) tablespace ware_1,
partition dist_P2 values less than (1089) tablespace ware_2,
partition dist_P3 values less than (1633) tablespace ware_3,
partition dist_P4 values less than (2177) tablespace ware_4,
partition dist_P5 values less than (2721) tablespace ware_5,
partition dist_P6 values less than (3265) tablespace ware_6,
partition dist_P7 values less than (3809) tablespace ware_7,
partition dist_P8 values less than (4353) tablespace ware_8,
partition dist_P9 values less than (4897) tablespace ware_9,
partition dist_P10 values less than (5441) tablespace ware_10,
partition dist_P11 values less than (5985) tablespace ware_11,
partition dist_P12 values less than (6529) tablespace ware_12,
partition dist_P13 values less than (7073) tablespace ware_13,
partition dist_P14 values less than (7617) tablespace ware_14,
partition dist_P15 values less than (8161) tablespace ware_15,
partition dist_P16 values less than (MAXVALUE) tablespace ware_16
    )
        initrans 4
        pctfree 95 pctused 4
        storage (initial 10M next 1M pctincrease 0);


        create table history (
        h_c_id          number,
        h_c_d_id        number,
        h_c_w_id        number,
        h_d_id          number,
        h_w_id          number,
        h_date          date,
        h_amount        number(6),
        h_data          varchar2(24)
        )
        partition by range (h_w_id)
        (
partition hist_P1 values less than (545) tablespace hist_1,
partition hist_P2 values less than (1089) tablespace hist_2,
partition hist_P3 values less than (1633) tablespace hist_3,
partition hist_P4 values less than (2177) tablespace hist_4,
partition hist_P5 values less than (2721) tablespace hist_5,
partition hist_P6 values less than (3265) tablespace hist_6,
partition hist_P7 values less than (3809) tablespace hist_7,
partition hist_P8 values less than (4353) tablespace hist_8,
partition hist_P9 values less than (4897) tablespace hist_9,
partition hist_P10 values less than (5441) tablespace hist_10,
partition hist_P11 values less than (5985) tablespace hist_11,
partition hist_P12 values less than (6529) tablespace hist_12,
partition hist_P13 values less than (7073) tablespace hist_13,
partition hist_P14 values less than (7617) tablespace hist_14,
partition hist_P15 values less than (8161) tablespace hist_15,
partition hist_P16 values less than (MAXVALUE) tablespace hist_16
        )
        initrans 3
        pctfree 1
        storage (initial 10K next 400M pctincrease 0 maxextents unlimited
                freelist groups 80 freelists 17) ;


        create table new_order (
                no_o_id         number,
                no_d_id         number,
                no_w_id         number
        )
        partition by range (no_w_id)
        (
partition nord_P1 values less than (545) tablespace nord_1,
partition nord_P2 values less than (1089) tablespace nord_2,
partition nord_P3 values less than (1633) tablespace nord_3,
partition nord_P4 values less than (2177) tablespace nord_4,
partition nord_P5 values less than (2721) tablespace nord_5,
partition nord_P6 values less than (3265) tablespace nord_6,
partition nord_P7 values less than (3809) tablespace nord_7,
partition nord_P8 values less than (4353) tablespace nord_8,
partition nord_P9 values less than (4897) tablespace nord_9,
partition nord_P10 values less than (5441) tablespace nord_10,
partition nord_P11 values less than (5985) tablespace nord_11,
partition nord_P12 values less than (6529) tablespace nord_12,
partition nord_P13 values less than (7073) tablespace nord_13,
partition nord_P14 values less than (7617) tablespace nord_14,
partition nord_P15 values less than (8161) tablespace nord_15,
partition nord_P16 values less than (MAXVALUE) tablespace nord_16
        )
        initrans 4
        pctfree 5
        storage (initial 10K next 150M pctincrease 0 maxextents unlimited
                freelist groups 80 freelists 17);


      create table orders (
        o_id            number,
        o_d_id          number,
        o_w_id          number,
        o_c_id          number,
        o_entry_d       date,
        o_carrier_id    number,
        o_ol_cnt        number,
        o_all_local     number
        )
        partition by range (o_w_id)
        (
partition order_P1 values less than (182) tablespace ord_1,
partition order_P2 values less than (363) tablespace ord_2,
partition order_P3 values less than (545) tablespace ord_3,
partition order_P4 values less than (726) tablespace ord_4,
partition order_P5 values less than (907) tablespace ord_5,
partition order_P6 values less than (1089) tablespace ord_6,
partition order_P7 values less than (1270) tablespace ord_7,
partition order_P8 values less than (1451) tablespace ord_8,
partition order_P9 values less than (1633) tablespace ord_9,
partition order_P10 values less than (1814) tablespace ord_10,
partition order_P11 values less than (1995) tablespace ord_11,
partition order_P12 values less than (2177) tablespace ord_12,
partition order_P13 values less than (2358) tablespace ord_13,
partition order_P14 values less than (2539) tablespace ord_14,
partition order_P15 values less than (2721) tablespace ord_15,
partition order_P16 values less than (2902) tablespace ord_16,
partition order_P17 values less than (3083) tablespace ord_17,
partition order_P18 values less than (3265) tablespace ord_18,
partition order_P19 values less than (3446) tablespace ord_19,
partition order_P20 values less than (3627) tablespace ord_20,
partition order_P21 values less than (3809) tablespace ord_21,
partition order_P22 values less than (3990) tablespace ord_22,
partition order_P23 values less than (4171) tablespace ord_23,
partition order_P24 values less than (4353) tablespace ord_24,
partition order_P25 values less than (4534) tablespace ord_25,
partition order_P26 values less than (4715) tablespace ord_26,
partition order_P27 values less than (4897) tablespace ord_27,
partition order_P28 values less than (5078) tablespace ord_28,
partition order_P29 values less than (5259) tablespace ord_29,
partition order_P30 values less than (5441) tablespace ord_30,
partition order_P31 values less than (5622) tablespace ord_31,
partition order_P32 values less than (5803) tablespace ord_32,
partition order_P33 values less than (5985) tablespace ord_33,
partition order_P34 values less than (6166) tablespace ord_34,
partition order_P35 values less than (6347) tablespace ord_35,
partition order_P36 values less than (6529) tablespace ord_36,
partition order_P37 values less than (6710) tablespace ord_37,
partition order_P38 values less than (6891) tablespace ord_38,
partition order_P39 values less than (7073) tablespace ord_39,
partition order_P40 values less than (7254) tablespace ord_40,
partition order_P41 values less than (7435) tablespace ord_41,
partition order_P42 values less than (7617) tablespace ord_42,
partition order_P43 values less than (7798) tablespace ord_43,
```

```
partition order_P44 values less than (7979) tablespace ord_44,
partition order_P45 values less than (8161) tablespace ord_45,
partition order_P46 values less than (8342) tablespace ord_46,
partition order_P47 values less than (8523) tablespace ord_47,
partition order_P48 values less than (MAXVALUE) tablespace ord_48
        )
        initrans 3
        pctfree 5
        storage (initial 10K next 220M pctincrease 0 maxextents unlimited
                freelist groups 80 freelists 17);


        create table order_line (
        ol_o_id         number,
        ol_d_id         number,
        ol_w_id         number,
        ol_number       number,
        ol_delivery_d   date,
        ol_i_id         number,
        ol_supply_w_id  number,
        ol_quantity     number,
        ol_amount       number(6),
        ol_dist_info    char(24)
        )
        partition by range (ol_w_id)
        (
partition ordl_P1 values less than (182) tablespace ordl_1,
partition ordl_P2 values less than (363) tablespace ordl_2,
partition ordl_P3 values less than (545) tablespace ordl_3,
partition ordl_P4 values less than (726) tablespace ordl_4,
partition ordl_P5 values less than (907) tablespace ordl_5,
partition ordl_P6 values less than (1089) tablespace ordl_6,
partition ordl_P7 values less than (1270) tablespace ordl_7,
partition ordl_P8 values less than (1451) tablespace ordl_8,
partition ordl_P9 values less than (1633) tablespace ordl_9,
partition ordl_P10 values less than (1814) tablespace ordl_10,
partition ordl_P11 values less than (1995) tablespace ordl_11,
partition ordl_P12 values less than (2177) tablespace ordl_12,
partition ordl_P13 values less than (2358) tablespace ordl_13,
partition ordl_P14 values less than (2539) tablespace ordl_14,
partition ordl_P15 values less than (2721) tablespace ordl_15,
partition ordl_P16 values less than (2902) tablespace ordl_16,
partition ordl_P17 values less than (3083) tablespace ordl_17,
partition ordl_P18 values less than (3265) tablespace ordl_18,
partition ordl_P19 values less than (3446) tablespace ordl_19,
partition ordl_P20 values less than (3627) tablespace ordl_20,
partition ordl_P21 values less than (3809) tablespace ordl_21,
partition ordl_P22 values less than (3990) tablespace ordl_22,
partition ordl_P23 values less than (4171) tablespace ordl_23,
partition ordl_P24 values less than (4353) tablespace ordl_24,
partition ordl_P25 values less than (4534) tablespace ordl_25,
partition ordl_P26 values less than (4715) tablespace ordl_26,
partition ordl_P27 values less than (4897) tablespace ordl_27,
partition ordl_P28 values less than (5078) tablespace ordl_28,
partition ordl_P29 values less than (5259) tablespace ordl_29,
partition ordl_P30 values less than (5441) tablespace ordl_30,
partition ordl_P31 values less than (5622) tablespace ordl_31,
partition ordl_P32 values less than (5803) tablespace ordl_32,
partition ordl_P33 values less than (5985) tablespace ordl_33,
partition ordl_P34 values less than (6166) tablespace ordl_34,
partition ordl_P35 values less than (6347) tablespace ordl_35,
partition ordl_P36 values less than (6529) tablespace ordl_36,
partition ordl_P37 values less than (6710) tablespace ordl_37,
partition ordl_P38 values less than (6891) tablespace ordl_38,
partition ordl_P39 values less than (7073) tablespace ordl_39,
partition ordl_P40 values less than (7254) tablespace ordl_40,
partition ordl_P41 values less than (7435) tablespace ordl_41,
partition ordl_P42 values less than (7617) tablespace ordl_42,
partition ordl_P43 values less than (7798) tablespace ordl_43,
partition ordl_P44 values less than (7979) tablespace ordl_44,
partition ordl_P45 values less than (8161) tablespace ordl_45,
partition ordl_P46 values less than (8342) tablespace ordl_46,
partition ordl_P47 values less than (8523) tablespace ordl_47,
partition ordl_P48 values less than (MAXVALUE) tablespace ordl_48
        )
        initrans 4
        pctfree 5
        storage (initial 10K next 1500M pctincrease 0 maxextents unlimited
                freelist groups 80 freelists 17);


rem
rem ITEM table
rem

        create cluster icluster (
        i_id            number(6,0)
        )
        hashkeys        100000
        hash is         i_id
        size            120
        initrans        3
        pctfree         0
        tablespace      items
        storage (initial 740K next 740K pctincrease 0);

        create table item (
        i_id            number(6,0),
        i_im_id         number,
        i_name          varchar2(24),
        i_price         number(5,0),
        i_data          varchar2(50)
        )
        cluster icluster(i_id);


rem
rem done
rem

        exit;
```

tpcc_tab2.sql

```
rem
rem
================================================================
========+
rem        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
rem                  OPEN SYSTEMS PERFORMANCE GROUP                 |
rem                     All Rights Reserved                         |
rem
================================================================
========+
rem  FILENAME
rem       tpcc_tab2.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem
================================================================
========
rem

rem
rem DROP all first
rem
rem drop cluster ccluster including tables;
rem drop table customer;

set timing on
set echo on;

rem
rem CUSTOMER table
rem

        create cluster ccluster (
          c_id          number(5,0),
          c_d_id        number(2,0),
          c_w_id        number(5,0)
        )
        hashkeys        262080000
        hash is         ( trunc((c_w_id - 1)/544) * 60000 + c_w_id * 30000 +
                        c_d_id * 3000 + c_id - 3001 )
        size            850
        initrans        3
        pctfree         0
        tablespace      cust
        storage (initial 1092000K next 1092000K pctincrease 0 minextents 120
                                maxextents unlimited freelist groups 16 freelists 19
);

                create table customer (
        c_id            number(5,0),
        c_d_id          number(2,0),
        c_w_id          number(5,0),
        c_first         varchar2(16),
        c_middle        char(2),
        c_last          varchar2(16),
        c_street_1      varchar2(20),
        c_street_2      varchar2(20),
        c_city          varchar2(20),
        c_state         char(2),
        c_zip           char(9),
        c_phone         char(16),
```

```
    c_since         date,
    c_credit        char(2),
    c_credit_lim    number(12),
    c_discount      number(4),
    c_balance       number(12),
    c_ytd_payment   number(12),
    c_payment_cnt   number(8),
    c_delivery_cnt  number(8),
    c_data          varchar2(500)
    )
    cluster ccluster (c_id, c_d_id, c_w_id);

rem
rem done
rem

    exit;
```

tpcc_tab3.sql

```
rem
rem
================================================================
========+
rem        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
rem                OPEN SYSTEMS PERFORMANCE GROUP                |
rem                    All Rights Reserved                       |
rem
================================================================
========+
rem  FILENAME
rem       tpcc_tab3.sql
rem  DESCRIPTION
rem       Create stock table for TPC-C database.
rem
================================================================
========
rem

rem
rem DROP all first
rem
rem drop cluster scluster including tables;
rem drop table stock;

set timing on
set echo on;

rem
rem STOCK table
rem

    create cluster scluster (
      s_i_id        number(6,0),
      s_w_id        number(5,0)
    )
    hashkeys        873600000
    hash is         ( (s_i_id - 1) * 544 + mod((s_w_id - 1), 544) +
                    trunc ((s_w_id - 1) / 544) * 54600000 + 100000 )
    size            350
    initrans        3
    pctfree         0
    tablespace      stocks
    storage (initial 390000K next 390000K pctincrease 0 minextents 448
                    maxextents unlimited  freelist groups 16 freelists 19 );

                create table stock (
    s_i_id          number(6,0),
    s_w_id          number(5,0),
    s_quantity      number(6,0),
    s_dist_01       char(24),
    s_dist_02       char(24),
    s_dist_03       char(24),
    s_dist_04       char(24),
    s_dist_05       char(24),
    s_dist_06       char(24),
    s_dist_07       char(24),
    s_dist_08       char(24),
    s_dist_09       char(24),
    s_dist_10       char(24),
    s_ytd           number(10,0),
    s_order_cnt     number(6,0),
    s_remote_cnt    number(6,0),
    s_data          varchar2(50)
    )
    cluster scluster (s_i_id, s_w_id);

rem
rem done
rem

    exit;
```

tpcc_ix2.sql

```
rem
rem
================================================================
========+
rem        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
rem                OPEN SYSTEMS PERFORMANCE GROUP                |
rem                    All Rights Reserved                       |
rem
================================================================
========+
rem  FILENAME
rem       tpcc_ix1.sql
rem  DESCRIPTION
rem       Create indexes for TPC-C database.
rem
================================================================
========
rem
set echo on;

rem drop index iwarehouse;
rem drop index idistrict;
rem drop index icustomer;
rem drop index icustomer2;
rem drop index istock;
rem drop index iitem;

set timing on

create unique index iwarehouse on warehouse(w_id)
    local
    initrans 3
    storage (initial 200K next 20K pctincrease 0) pctfree 1;

create unique index idistrict on district(d_w_id, d_id)
    local
    initrans 3
    storage (initial 2000K next 60K pctincrease 0) pctfree 1;

create unique index iitem on item(i_id)
    tablespace items
    storage (initial 100K next 100K pctincrease 0) pctfree 1;

create unique index icustomer on customer(c_w_id, c_d_id, c_id)
    tablespace icust1
    initrans 3
    nologging
    parallel 50
    storage (initial 64M next 8M pctincrease 0 maxextents unlimited ) pctfree 1;

create unique index icustomer2 on customer(c_last, c_w_id, c_d_id, c_first, c_id)
    tablespace icust2
    initrans 3
    nologging
    parallel 50
    storage (initial 100M next 50M pctincrease 0 maxextents unlimited ) pctfree
1;

create unique index istock on stock(s_i_id, s_w_id)
    tablespace istk
    initrans 3
    nologging
    parallel 16
    storage (initial 1100M next 45M pctincrease 0 maxextents unlimited ) pctfree
1;

alter index iitem      deallocate unused;
alter index icustomer  deallocate unused;
alter index icustomer2 deallocate unused;
alter index istock     deallocate unused;

exit;
```

create_rollback_segments.sh

```
#!/bin/ksh

ORACLE_HOME=/usr/oracle
ORACLE_SID=TPCC1
export ORACLE_HOME ORACLE_SID

(
echo "connect internal"
typeset -Z5 n=1000
for q in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
do
i=1
(( m =  $q * 1000 ))
while [ $i -le 200 ]
do
#(( j = ( $i - 1 ) % 3 + 1 + (( $q - 1 ) * 3 )  ))
#(( j = ( $i - 1 ) % 3 + 1 + (( $q - 1 ) * 3 )  ))
(( n = $i + $m ))
echo "\
  CREATE ROLLBACK SEGMENT t$n
    TABLESPACE roll_$q
    STORAGE (initial 100K next 100K minextents 2);
"
let i=i+1
done
done
) |pg

#$ORACLE_HOME/bin/svrmgrl
```

Data Distribution

Controller	Drives	Disk / Channel	12 Size	Use
Bridge 1	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 3	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 4 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 8	1150 MB each	stock & iordl
Bridge 2	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 4	500 MB each	stock & ord
	9 GB	Disk 2 , Chan 4 – 7	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 2 – 4	1950 MB each	stock & ordl
	9 GB	Disk 3 , Chan 5 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ord1
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 2	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 3	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 4	2000 MB each	hist & icust1
	9 GB	Disk 6 , Chan 5 – 8	1150 MB each	stock & iordl
Bridge 3	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 5	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 6 – 8	1950 MB each	stock & ordl
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 5	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 6	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 7	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 8	2000 MB each	hist & icust1
Bridge 4	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 3	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 4 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 8	1150 MB each	stock & iordl
Bridge 5	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 4	500 MB each	stock & ord
	9 GB	Disk 2 , Chan 4 – 7	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 2 – 4	1950 MB each	stock & ordl
	9 GB	Disk 3 , Chan 5 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ord1
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 2	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 3	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 4	2000 MB each	hist & icust1
	9 GB	Disk 6 , Chan 5 – 8	1150 MB each	stock & iordl
Bridge 6	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 5	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 6 – 8	1950 MB each	stock & ordl

	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 5	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 6	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 7	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 8	2000 MB each	hist & icust1
Bridge 7	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 3	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 4 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 8	1150 MB each	stock & iordl
Bridge 8	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 4	500 MB each	stock & ord
	9 GB	Disk 2 , Chan 4 – 7	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 2 – 4	1950 MB each	stock & ordl
	9 GB	Disk 3 , Chan 5 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ord1
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 2	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 3	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 4	2000 MB each	hist & icust1
	9 GB	Disk 6 , Chan 5 – 8	1150 MB each	stock & iordl
Bridge 9	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 5	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 6 – 8	1950 MB each	stock & ordl
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 5	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 6	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 7	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 8	2000 MB each	hist & icust1
Bridge 10	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 3	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 4 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 8	1150 MB each	stock & iordl
Bridge 11	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 4	500 MB each	stock & ord
	9 GB	Disk 2 , Chan 4 – 7	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 2 – 4	1950 MB each	stock & ordl
	9 GB	Disk 3 , Chan 5 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ord1
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 2	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 3	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 4	2000 MB each	hist & icust1
	9 GB	Disk 6 , Chan 5 – 8	1150 MB each	stock & iordl
Bridge 12	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 5	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 6 – 8	1950 MB each	stock & ordl
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 5	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 6	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 7	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 8	2000 MB each	hist & icust1
Bridge 13	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 3	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 4 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1

	9 GB	Disk 6 , Chan 1 – 8	1150 MB each	stock & iordl
Bridge 14	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 4	500 MB each	stock & ord
	9 GB	Disk 2 , Chan 4 – 7	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 2 – 4	1950 MB each	stock & ordl
	9 GB	Disk 3 , Chan 5 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ord1
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 2	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 3	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 4	2000 MB each	hist & icust1
	9 GB	Disk 6 , Chan 5 – 8	1150 MB each	stock & iordl
Bridge 15	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 5	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 6 – 8	1950 MB each	stock & ordl
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 5	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 6	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 7	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 8	2000 MB each	hist & icust1
Bridge 16	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 3	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 4 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 8	1150 MB each	stock & iordl
Bridge 17	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 4	500 MB each	stock & ord
	9 GB	Disk 2 , Chan 4 – 7	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 2 – 4	1950 MB each	stock & ordl
	9 GB	Disk 3 , Chan 5 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ord1
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 2	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 3	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 4	2000 MB each	hist & icust1
	9 GB	Disk 6 , Chan 5 – 8	1150 MB each	stock & iordl
Bridge 18	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 5	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 6 – 8	1950 MB each	stock & ordl
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 5	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 6	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 7	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 8	2000 MB each	hist & icust1
Bridge 19	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 3	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 4 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 8	1150 MB each	stock & iordl
Bridge 20	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 4	500 MB each	stock & ord
	9 GB	Disk 2 , Chan 4 – 7	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 2 – 4	1950 MB each	stock & ordl
	9 GB	Disk 3 , Chan 5 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ord1
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 2	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 3	800 MB each	stock & inord

	9 GB	Disk 6 , Chan 4	2000 MB each	hist & icust1
	9 GB	Disk 6 , Chan 5 – 8	1150 MB each	stock & iordl
Bridge 21	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 5	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 6 – 8	1950 MB each	stock & ordl
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 5	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 6	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 7	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 8	2000 MB each	hist & icust1
Bridge 22	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 3	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 4 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 8	1150 MB each	stock & iordl
Bridge 23	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 4	500 MB each	stock & ord
	9 GB	Disk 2 , Chan 4 – 7	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 2 – 4	1950 MB each	stock & ordl
	9 GB	Disk 3 , Chan 5 – 8	500 MB each	stock & ord
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ord1
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 2	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 3	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 4	2000 MB each	hist & icust1
	9 GB	Disk 6 , Chan 5 – 8	1150 MB each	stock & iordl
Bridge 24	9 GB	Disk 1 , Chan 1 – 8	1300 MB each	cust & iord2
	9 GB	Disk 2 , Chan 1 – 8	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 1 – 5	500 MB each	stock & ord
	9 GB	Disk 3 , Chan 6 – 8	1950 MB each	stock & ordl
	9 GB	Disk 4 , Chan 1 – 8	1950 MB each	stock & ordl
	9 GB	Disk 5 , Chan 1 – 8	500 MB each	stock & iord1
	9 GB	Disk 6 , Chan 1 – 5	1150 MB each	stock & iordl
	9 GB	Disk 6 , Chan 6	1050 MB each	stock & nord
	9 GB	Disk 6 , Chan 7	800 MB each	stock & inord
	9 GB	Disk 6 , Chan 8	2000 MB each	hist & icust1
Bridge 25	9 GB	Disk 1 , Chan 1 –3	1000 MB each	System. ware & item
	9 GB	Disk 2 , Chan 1 –3	1000 MB each	System, ware & item
	9 GB	Disk 3 , Chan 1 –3	1000 MB each	System, ware & item
	9 GB	Disk 4 , Chan 1 –3	1500 MB each	roll & istk
	9 GB	Disk 5 , Chan 1 –3	1500 MB each	roll & istk
	9 GB	Disk 6 , Chan 1 –3	1500 MB each	roll & istk
	9 GB	Chan 4, Disk 1 - 6	1600 MB each	icust2
Bridge 26	9 GB	Disk 1 , Chan 1 –3	1000 MB each	System. ware & item
	9 GB	Disk 2 , Chan 1 –3	1000 MB each	System, ware & item
	9 GB	Disk 3 , Chan 1 –3	1000 MB each	System, ware & item
	9 GB	Disk 4 , Chan 1 –3	1500 MB each	roll & istk
	9 GB	Disk 5 , Chan 1 –3	1500 MB each	roll & istk
	9 GB	Disk 6 , Chan 1 –3	1500 MB each	roll & istk
	9 GB	Chan 4, Disk 1 - 6	1600 MB each	icust2
PCI Cntrl 1	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 2	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 3	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 4	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 5	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 6	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 7	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 8	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 9	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 10	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 11	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 12	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 13	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 14	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 15	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 16	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 17	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 18	18 GB	Disk 1-3	8000 MB each	log

PCI Cntrl 19	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 20	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 21	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 22	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 23	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 24	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 25	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 26	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 27	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 28	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 29	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 30	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 31	18 GB	Disk 1-3	8000 MB each	log
PCI Cntrl 32	18 GB	Disk 1-3	8000 MB each	log

Stored Procedures

New.sql

```
--new
~
--new
============================================================
========+
~
--new       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA       |
~
--new                 OPEN SYSTEMS PERFORMANCE GROUP                  |
~
--new                    All Rights Reserved                          |
--new
============================================================
========+
--new  FILENAME
--new       new.sql
--new  DESCRIPTION
--new       SQL script to create a stored package for new order
--new       transactions.
--new
============================================================
========
--new
  DECLARE
     not_serializable         EXCEPTION;
     PRAGMA EXCEPTION_INIT(not_serializable,-8177);
     deadlock                 EXCEPTION;
     PRAGMA EXCEPTION_INIT(deadlock,-60);
     snapshot_too_old         EXCEPTION;
     PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);
   BEGIN
     LOOP BEGIN
       SELECT c_discount, c_last, c_credit
         INTO :c_discount, :c_last, :c_credit
         FROM customer
         WHERE c_id = :c_id
           AND c_d_id = :d_id
           AND c_w_id = :w_id;

       UPDATE wh_dist SET d_next_o_id = d_next_o_id + 1, d_tax=d_tax+0
        WHERE  d_id = :d_id
          AND  w_id = :w_id
        RETURNING d_tax, d_next_o_id-1, w_tax
          INTO :d_tax, :o_id, :w_tax;

       INSERT INTO new_order (no_o_id, no_d_id, no_w_id)
           VALUES (:o_id , :d_id, :w_id);
       INSERT INTO orders (o_id, o_w_id, o_d_id, o_c_id, o_carrier_id,
                    o_ol_cnt, o_all_local,o_entry_d)
           VALUES (:o_id, :w_id, :d_id, :c_id, 11,
                :o_ol_cnt, :o_all_local, :cr_date);
       RETURN;

       EXCEPTION
         WHEN not_serializable OR deadlock OR snapshot_too_old THEN
           ROLLBACK;
           :retry := :retry + 1;
       END;
     END LOOP;
   END;
```

Pay_id.sql

```
DECLARE /* paynz */
~
--      cust_rowid              ROWID;
~
--      dist_name               VARCHAR2(11);
--      ware_name               VARCHAR2(11);
     not_serializable         EXCEPTION;
     PRAGMA EXCEPTION_INIT(not_serializable,-8177);
     deadlock                 EXCEPTION;
     PRAGMA EXCEPTION_INIT(deadlock,-60);
     snapshot_too_old         EXCEPTION;
     PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);
   BEGIN
     LOOP BEGIN
         UPDATE  customer
           SET  c_balance = c_balance - :h_amount,
                c_ytd_payment = c_ytd_payment + :h_amount,
                c_payment_cnt = c_payment_cnt+1
           WHERE    c_id = :c_id AND c_d_id = :c_d_id AND
                c_w_id = :c_w_id
         RETURNING rowid, c_first, c_middle, c_last, 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
           INTO pay.cust_rowid,:c_first, :c_middle, :c_last, :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 := ' ';

       IF :c_credit = 'BC' THEN
         UPDATE customer
                   SET c_data= substr ((to_char (:c_id) || ' ' ||
                        to_char (:c_d_id) || ' ' ||
                        to_char (:c_w_id) || ' ' ||
                        to_char (:d_id) || ' ' ||
                        to_char (:w_id) || ' ' ||
                        to_char (:h_amount, '9999.99') || ' ' |
                        || c_data, 1, 500)
          WHERE rowid = pay.cust_rowid
       RETURNING substr(c_data,1, 200)
          INTO :c_data;

       END IF;

         UPDATE district
           SET d_ytd = d_ytd + :h_amount
         WHERE d_id = :d_id
           AND d_w_id = :w_id
       RETURNING d_name, d_street_1, d_street_2, d_city,d_state, d_zip
          INTO pay.dist_name,:d_street_1,:d_street_2,:d_city,:d_state,
             :d_zip;

         UPDATE warehouse
           SET w_ytd = w_ytd + :h_amount
         WHERE w_id = :w_id
       RETURNING w_name, w_street_1, w_street_2, w_city, w_state, w_zip
          INTO pay.ware_name, :w_street_1, :w_street_2, :w_city, :w_state,
             :w_zip;

       INSERT INTO history  (h_c_id, h_c_d_id, h_c_w_id, h_d_id, h_w_id,
                    h_amount, h_date, h_data)
       VALUES
         (:c_id, :c_d_id, :c_w_id, :d_id, :w_id, :h_amount,
         :cr_date, pay.ware_name || '    ' || pay.dist_name);
       COMMIT;
       :h_date := to_char (:cr_date, 'DD-MM-YYYY.HH24:MI:SS');
       EXIT;

       EXCEPTION
         WHEN not_serializable OR deadlock OR snapshot_too_old THEN
           ROLLBACK;
           :retry := :retry + 1;
       END;

     END LOOP;
   END;
```

Pay_ln.sql

```
DECLARE /* payz */
--      TYPE rowidarray IS TABLE OF ROWID INDEX BY
BINARY_INTEGER;
--      cust_rowid              ROWID;
--      dist_name               VARCHAR2(11);
--      ware_name               VARCHAR2(11);
--      c_num                   BINARY_INTEGER;
--      row_id                  rowidarray;
     not_serializable         EXCEPTION;
     PRAGMA EXCEPTION_INIT(not_serializable,-8177);
     deadlock                 EXCEPTION;
     PRAGMA EXCEPTION_INIT(deadlock,-60);
     snapshot_too_old         EXCEPTION;
     PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);
     CURSOR c_cur IS
       SELECT rowid
       FROM customer
       WHERE c_d_id = :c_d_id AND c_w_id = :c_w_id AND c_last = :c_last
       ORDER BY c_w_id, c_d_id, c_last, c_first;
   BEGIN
     LOOP BEGIN

       pay.c_num := 0;
       FOR c_id_rec IN c_cur LOOP
         pay.c_num := pay.c_num + 1;
         pay.row_id(pay.c_num) := c_id_rec.rowid;
       END LOOP;
```

```
 pay.cust_rowid := pay.row_id ((pay.c_num + 1) / 2);

UPDATE customer
  SET c_balance = c_balance - :h_amount,
      c_ytd_payment = c_ytd_payment+ :h_amount,
      c_payment_cnt = c_payment_cnt+1
WHERE rowid = pay.cust_rowid
RETURNING
       c_id, c_first, c_middle, c_last, 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
    INTO :c_id, :c_first, :c_middle, :c_last,
         :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 := ' ';
IF :c_credit = 'BC' THEN
   UPDATE customer
     SET c_data = substr ((to_char (:c_id) || ' ' ||
                          to_char (:c_d_id) || ' ' ||
                          to_char (:c_w_id) || ' ' ||
                          to_char (:d_id) || ' ' ||
                          to_char (:w_id) || ' ' ||
                          to_char (:h_amount/100, '9999.99') || ' | ')
                          || c_data, 1, 500)
      WHERE rowid = pay.cust_rowid
      RETURNING substr(c_data,1, 200)
      INTO :c_data;

END IF;

UPDATE district
   SET d_ytd = d_ytd+:h_amount
   WHERE d_id = :d_id
     AND d_w_id = :w_id
   RETURNING  d_name, d_street_1, d_street_2, d_city,
                    d_state, d_zip
   INTO pay.dist_name, :d_street_1, :d_street_2, :d_city,
       :d_state, :d_zip;

UPDATE warehouse
   SET w_ytd = w_ytd+:h_amount
   WHERE w_id = :w_id
   RETURNING w_name,
          w_street_1, w_street_2, w_city, w_state, w_zip
    INTO pay.ware_name,
        :w_street_1, :w_street_2, :w_city, :w_state, :w_zip;

INSERT INTO history (h_c_id, h_c_d_id, h_c_w_id, h_d_id, h_w_id,
                h_amount, h_date, h_data)
    VALUES (:c_id, :c_d_id, :c_w_id, :d_id, :w_id, :h_amount,
    :cr_date, pay.ware_name || '    ' || pay.dist_name);
COMMIT;
:h_date := to_char (:cr_date, 'DD-MM-YYYY.HH24:MI:SS');
EXIT;

EXCEPTION
   WHEN not_serializable OR deadlock OR snapshot_too_old THEN
      ROLLBACK;
      :retry := :retry + 1;
END;

  END LOOP;
END;
```

Appendix C – Tunable Parameters

Server Configuration Parameters

Oracle8 init.ora Configuration Parameters

```
#####################################################
### init.ora
#####################################################

_db_block_hash_buckets   = 400009
_db_block_write_batch    = 1000
_lm_direct_sends         = all
buffer_pool_recycle      = (buffers:10000, lru_latches:3)
compatible               = 8.0.4
control_files            = ?/dbs/tpcc_16sq_disks/ctrl1, ?/dbs/tpcc_16sq_disks/ctrl2
cpu_count                = 4
cursor_space_for_time    = TRUE
db_block_checkpoint_batch= 650
db_block_lru_latches     = 12
db_block_max_dirty_target= 0
db_block_size            = 2048
db_files                 = 2650
db_name                  = tp16
discrete_transactions_enabled= FALSE
distributed_transactions = 0
dml_locks                = 0
enqueue_resources        = 4000
gc_defer_time            = 0
gc_files_to_locks        = "380-492,373-379,941-1060=140each:1061=10:501-
940,493-500,1062-1509=1000each:1510=10:1,2=100each:3-294,296-302,304-
306,308,310,323-372,1571-1572=1each"
gc_latches               = 8
gc_releasable_locks      = 20000
gc_rollback_locks        = "0,31-3230=20each"
hash_join_enabled        = FALSE
lm_locks                 = 2200000
lm_procs                 = 200
lm_ress                  = 1050000
log_archive_buffer_size  = 32
log_archive_start        = FALSE
log_buffer               = 1048576
log_checkpoint_interval  = 1000000000
log_checkpoints_to_alert = TRUE
log_simultaneous_copies  = 12
max_dump_file_size       = 100000
max_rollback_segments    = 401
open_cursors             = 200
parallel_min_message_pool= 0
parallel_min_servers     = 0
parallel_server          = TRUE
processes                = 150
replication_dependency_tracking= FALSE
shared_pool_size         = 40000000
spin_count               = 3500
timed_statistics         = FALSE
transaction_auditing     = FALSE
transactions_per_rollback_segment= 1
use_indirect_data_buffers= TRUE
use_post_wait_driver     = TRUE


#####################################################
### init1.ora
#####################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=1
thread=1
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t01001, t01002, t01003, t01004, t01005, \
t01006, t01007, t01008, t01009, t01010, \
t01011, t01012, t01013, t01014, t01015, \
t01016, t01017, t01018, t01019, t01020, \
t01021, t01022, t01023, t01024, t01025, \
t01026, t01027, t01028, t01029, t01030, \
t01031, t01032, t01033, t01034, t01035, \
t01036, t01037, t01038, t01039, t01040, \
t01041, t01042, t01043, t01044, t01045, \
t01046, t01047, t01048, t01049, t01050, \
t01051, t01052, t01053, t01054, t01055, \
t01056, t01057, t01058, t01059, t01060, \
t01061, t01062, t01063, t01064, t01065, \
t01066, t01067, t01068, t01069, t01070, \
t01071, t01072, t01073, t01074, t01075, \
t01076, t01077, t01078, t01079, t01080, \
t01081, t01082, t01083, t01084, t01085, \
t01086, t01087, t01088, t01089, t01090, \
t01091, t01092, t01093, t01094, t01095, \
t01096, t01097, t01098, t01099, t01100, \
t01101, t01102, t01103, t01104, t01105, \
t01106, t01107, t01108, t01109, t01110, \
t01111, t01112, t01113, t01114, t01115, \
t01116, t01117, t01118, t01119, t01120, \
t01121, t01122, t01123, t01124, t01125, \
t01126, t01127, t01128, t01129, t01130, \
t01131, t01132, t01133, t01134, t01135, \
t01136, t01137, t01138, t01139, t01140, \
t01141, t01142, t01143, t01144, t01145, \
t01146, t01147, t01148, t01149, t01150, \
t01151, t01152, t01153, t01154, t01155, \
t01156, t01157, t01158, t01159, t01160, \
t01161, t01162, t01163, t01164, t01165, \
t01166, t01167, t01168, t01169, t01170, \
t01171, t01172, t01173, t01174, t01175, \
t01176, t01177, t01178, t01179, t01180, \
t01181, t01182, t01183, t01184, t01185, \
t01186, t01187, t01188, t01189, t01190, \
t01191, t01192, t01193, t01194, t01195, \
t01196, t01197, t01198, t01199, t01200 )


#####################################################
### init2.ora
#####################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=2
thread=2
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 128
rollback_segments = (
t02001, t02002, t02003, t02004, t02005, \
t02006, t02007, t02008, t02009, t02010, \
t02011, t02012, t02013, t02014, t02015, \
t02016, t02017, t02018, t02019, t02020, \
t02021, t02022, t02023, t02024, t02025, \
t02026, t02027, t02028, t02029, t02030, \
t02031, t02032, t02033, t02034, t02035, \
t02036, t02037, t02038, t02039, t02040, \
t02041, t02042, t02043, t02044, t02045, \
t02046, t02047, t02048, t02049, t02050, \
t02051, t02052, t02053, t02054, t02055, \
t02056, t02057, t02058, t02059, t02060, \
t02061, t02062, t02063, t02064, t02065, \
t02066, t02067, t02068, t02069, t02070, \
t02071, t02072, t02073, t02074, t02075, \
t02076, t02077, t02078, t02079, t02080, \
t02081, t02082, t02083, t02084, t02085, \
t02086, t02087, t02088, t02089, t02090, \
t02091, t02092, t02093, t02094, t02095, \
t02096, t02097, t02098, t02099, t02100, \
t02101, t02102, t02103, t02104, t02105, \
t02106, t02107, t02108, t02109, t02110, \
t02111, t02112, t02113, t02114, t02115, \
t02116, t02117, t02118, t02119, t02120, \
t02121, t02122, t02123, t02124, t02125, \
t02126, t02127, t02128, t02129, t02130, \
t02131, t02132, t02133, t02134, t02135, \
t02136, t02137, t02138, t02139, t02140, \
t02141, t02142, t02143, t02144, t02145, \
t02146, t02147, t02148, t02149, t02150, \
t02151, t02152, t02153, t02154, t02155, \
t02156, t02157, t02158, t02159, t02160, \
t02161, t02162, t02163, t02164, t02165, \
t02166, t02167, t02168, t02169, t02170, \
t02171, t02172, t02173, t02174, t02175, \
t02176, t02177, t02178, t02179, t02180, \
t02181, t02182, t02183, t02184, t02185, \
t02186, t02187, t02188, t02189, t02190, \
```

```
t02191, t02192, t02193, t02194, t02195, \
t02196, t02197, t02198, t02199, t02200 )


######################################################
### init3.ora
######################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=5
thread=5
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t05001, t05002, t05003, t05004, t05005, \
t05006, t05007, t05008, t05009, t05010, \
t05011, t05012, t05013, t05014, t05015, \
t05016, t05017, t05018, t05019, t05020, \
t05021, t05022, t05023, t05024, t05025, \
t05026, t05027, t05028, t05029, t05030, \
t05031, t05032, t05033, t05034, t05035, \
t05036, t05037, t05038, t05039, t05040, \
t05041, t05042, t05043, t05044, t05045, \
t05046, t05047, t05048, t05049, t05050, \
t05051, t05052, t05053, t05054, t05055, \
t05056, t05057, t05058, t05059, t05060, \
t05061, t05062, t05063, t05064, t05065, \
t05066, t05067, t05068, t05069, t05070, \
t05071, t05072, t05073, t05074, t05075, \
t05076, t05077, t05078, t05079, t05080, \
t05081, t05082, t05083, t05084, t05085, \
t05086, t05087, t05088, t05089, t05090, \
t05091, t05092, t05093, t05094, t05095, \
t05096, t05097, t05098, t05099, t05100, \
t05101, t05102, t05103, t05104, t05105, \
t05106, t05107, t05108, t05109, t05110, \
t05111, t05112, t05113, t05114, t05115, \
t05116, t05117, t05118, t05119, t05120, \
t05121, t05122, t05123, t05124, t05125, \
t05126, t05127, t05128, t05129, t05130, \
t05131, t05132, t05133, t05134, t05135, \
t05136, t05137, t05138, t05139, t05140, \
t05141, t05142, t05143, t05144, t05145, \
t05146, t05147, t05148, t05149, t05150, \
t05151, t05152, t05153, t05154, t05155, \
t05156, t05157, t05158, t05159, t05160, \
t05161, t05162, t05163, t05164, t05165, \
t05166, t05167, t05168, t05169, t05170, \
t05171, t05172, t05173, t05174, t05175, \
t05176, t05177, t05178, t05179, t05180, \
t05181, t05182, t05183, t05184, t05185, \
t05186, t05187, t05188, t05189, t05190, \
t05191, t05192, t05193, t05194, t05195, \
t05196, t05197, t05198, t05199, t05200 )


######################################################
### init4.ora
######################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=6
thread=6
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t06001, t06002, t06003, t06004, t06005, \
t06006, t06007, t06008, t06009, t06010, \
t06011, t06012, t06013, t06014, t06015, \
t06016, t06017, t06018, t06019, t06020, \
t06021, t06022, t06023, t06024, t06025, \
t06026, t06027, t06028, t06029, t06030, \
t06031, t06032, t06033, t06034, t06035, \
t06036, t06037, t06038, t06039, t06040, \
t06041, t06042, t06043, t06044, t06045, \
t06046, t06047, t06048, t06049, t06050, \
t06051, t06052, t06053, t06054, t06055, \
t06056, t06057, t06058, t06059, t06060, \
t06061, t06062, t06063, t06064, t06065, \
t06066, t06067, t06068, t06069, t06070, \
t06071, t06072, t06073, t06074, t06075, \
t06076, t06077, t06078, t06079, t06080, \
t06081, t06082, t06083, t06084, t06085, \
t06086, t06087, t06088, t06089, t06090, \
t06091, t06092, t06093, t06094, t06095, \
t06096, t06097, t06098, t06099, t06100, \
t06101, t06102, t06103, t06104, t06105, \
t06106, t06107, t06108, t06109, t06110, \
t06111, t06112, t06113, t06114, t06115, \
t06116, t06117, t06118, t06119, t06120, \
t06121, t06122, t06123, t06124, t06125, \
t06126, t06127, t06128, t06129, t06130, \
t06131, t06132, t06133, t06134, t06135, \
t06136, t06137, t06138, t06139, t06140, \
t06141, t06142, t06143, t06144, t06145, \
t06146, t06147, t06148, t06149, t06150, \
t06151, t06152, t06153, t06154, t06155, \
t06156, t06157, t06158, t06159, t06160, \
t06161, t06162, t06163, t06164, t06165, \
t06166, t06167, t06168, t06169, t06170, \
t06171, t06172, t06173, t06174, t06175, \
t06176, t06177, t06178, t06179, t06180, \
t06181, t06182, t06183, t06184, t06185, \
t06186, t06187, t06188, t06189, t06190, \
t06191, t06192, t06193, t06194, t06195, \
t06196, t06197, t06198, t06199, t06200 )


######################################################
### init5.ora
######################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=9
thread=9
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t09001, t09002, t09003, t09004, t09005, \
t09006, t09007, t09008, t09009, t09010, \
t09011, t09012, t09013, t09014, t09015, \
t09016, t09017, t09018, t09019, t09020, \
t09021, t09022, t09023, t09024, t09025, \
t09026, t09027, t09028, t09029, t09030, \
t09031, t09032, t09033, t09034, t09035, \
t09036, t09037, t09038, t09039, t09040, \
t09041, t09042, t09043, t09044, t09045, \
t09046, t09047, t09048, t09049, t09050, \
t09051, t09052, t09053, t09054, t09055, \
t09056, t09057, t09058, t09059, t09060, \
t09061, t09062, t09063, t09064, t09065, \
t09066, t09067, t09068, t09069, t09070, \
t09071, t09072, t09073, t09074, t09075, \
t09076, t09077, t09078, t09079, t09080, \
t09081, t09082, t09083, t09084, t09085, \
t09086, t09087, t09088, t09089, t09090, \
t09091, t09092, t09093, t09094, t09095, \
t09096, t09097, t09098, t09099, t09100, \
t09101, t09102, t09103, t09104, t09105, \
t09106, t09107, t09108, t09109, t09110, \
t09111, t09112, t09113, t09114, t09115, \
t09116, t09117, t09118, t09119, t09120, \
t09121, t09122, t09123, t09124, t09125, \
t09126, t09127, t09128, t09129, t09130, \
t09131, t09132, t09133, t09134, t09135, \
t09136, t09137, t09138, t09139, t09140, \
t09141, t09142, t09143, t09144, t09145, \
t09146, t09147, t09148, t09149, t09150, \
t09151, t09152, t09153, t09154, t09155, \
t09156, t09157, t09158, t09159, t09160, \
t09161, t09162, t09163, t09164, t09165, \
t09166, t09167, t09168, t09169, t09170, \
t09171, t09172, t09173, t09174, t09175, \
t09176, t09177, t09178, t09179, t09180, \
t09181, t09182, t09183, t09184, t09185, \
t09186, t09187, t09188, t09189, t09190, \
t09191, t09192, t09193, t09194, t09195, \
t09196, t09197, t09198, t09199, t09200 )


######################################################
### init6.ora
######################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=10
thread=10
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t10001, t10002, t10003, t10004, t10005, \
t10006, t10007, t10008, t10009, t10010, \
t10011, t10012, t10013, t10014, t10015, \
t10016, t10017, t10018, t10019, t10020, \
```

```
t10021, t10022, t10023, t10024, t10025, \
t10026, t10027, t10028, t10029, t10030, \
t10031, t10032, t10033, t10034, t10035, \
t10036, t10037, t10038, t10039, t10040, \
t10041, t10042, t10043, t10044, t10045, \
t10046, t10047, t10048, t10049, t10050, \
t10051, t10052, t10053, t10054, t10055, \
t10056, t10057, t10058, t10059, t10060, \
t10061, t10062, t10063, t10064, t10065, \
t10066, t10067, t10068, t10069, t10070, \
t10071, t10072, t10073, t10074, t10075, \
t10076, t10077, t10078, t10079, t10080, \
t10081, t10082, t10083, t10084, t10085, \
t10086, t10087, t10088, t10089, t10090, \
t10091, t10092, t10093, t10094, t10095, \
t10096, t10097, t10098, t10099, t10100, \
t10101, t10102, t10103, t10104, t10105, \
t10106, t10107, t10108, t10109, t10110, \
t10111, t10112, t10113, t10114, t10115, \
t10116, t10117, t10118, t10119, t10120, \
t10121, t10122, t10123, t10124, t10125, \
t10126, t10127, t10128, t10129, t10130, \
t10131, t10132, t10133, t10134, t10135, \
t10136, t10137, t10138, t10139, t10140, \
t10141, t10142, t10143, t10144, t10145, \
t10146, t10147, t10148, t10149, t10150, \
t10151, t10152, t10153, t10154, t10155, \
t10156, t10157, t10158, t10159, t10160, \
t10161, t10162, t10163, t10164, t10165, \
t10166, t10167, t10168, t10169, t10170, \
t10171, t10172, t10173, t10174, t10175, \
t10176, t10177, t10178, t10179, t10180, \
t10181, t10182, t10183, t10184, t10185, \
t10186, t10187, t10188, t10189, t10190, \
t10191, t10192, t10193, t10194, t10195, \
t10196, t10197, t10198, t10199, t10200 )


#########################################################
### init7.ora
#########################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=13
thread=13
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t13001, t13002, t13003, t13004, t13005, \
t13006, t13007, t13008, t13009, t13010, \
t13011, t13012, t13013, t13014, t13015, \
t13016, t13017, t13018, t13019, t13020, \
t13021, t13022, t13023, t13024, t13025, \
t13026, t13027, t13028, t13029, t13030, \
t13031, t13032, t13033, t13034, t13035, \
t13036, t13037, t13038, t13039, t13040, \
t13041, t13042, t13043, t13044, t13045, \
t13046, t13047, t13048, t13049, t13050, \
t13051, t13052, t13053, t13054, t13055, \
t13056, t13057, t13058, t13059, t13060, \
t13061, t13062, t13063, t13064, t13065, \
t13066, t13067, t13068, t13069, t13070, \
t13071, t13072, t13073, t13074, t13075, \
t13076, t13077, t13078, t13079, t13080, \
t13081, t13082, t13083, t13084, t13085, \
t13086, t13087, t13088, t13089, t13090, \
t13091, t13092, t13093, t13094, t13095, \
t13096, t13097, t13098, t13099, t13100, \
t13101, t13102, t13103, t13104, t13105, \
t13106, t13107, t13108, t13109, t13110, \
t13111, t13112, t13113, t13114, t13115, \
t13116, t13117, t13118, t13119, t13120, \
t13121, t13122, t13123, t13124, t13125, \
t13126, t13127, t13128, t13129, t13130, \
t13131, t13132, t13133, t13134, t13135, \
t13136, t13137, t13138, t13139, t13140, \
t13141, t13142, t13143, t13144, t13145, \
t13146, t13147, t13148, t13149, t13150, \
t13151, t13152, t13153, t13154, t13155, \
t13156, t13157, t13158, t13159, t13160, \
t13161, t13162, t13163, t13164, t13165, \
t13166, t13167, t13168, t13169, t13170, \
t13171, t13172, t13173, t13174, t13175, \
t13176, t13177, t13178, t13179, t13180, \
t13181, t13182, t13183, t13184, t13185, \
t13186, t13187, t13188, t13189, t13190, \
t13191, t13192, t13193, t13194, t13195, \
t13196, t13197, t13198, t13199, t13200 )


#########################################################
### init8.ora
#########################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=14
thread=14
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t14001, t14002, t14003, t14004, t14005, \
t14006, t14007, t14008, t14009, t14010, \
t14011, t14012, t14013, t14014, t14015, \
t14016, t14017, t14018, t14019, t14020, \
t14021, t14022, t14023, t14024, t14025, \
t14026, t14027, t14028, t14029, t14030, \
t14031, t14032, t14033, t14034, t14035, \
t14036, t14037, t14038, t14039, t14040, \
t14041, t14042, t14043, t14044, t14045, \
t14046, t14047, t14048, t14049, t14050, \
t14051, t14052, t14053, t14054, t14055, \
t14056, t14057, t14058, t14059, t14060, \
t14061, t14062, t14063, t14064, t14065, \
t14066, t14067, t14068, t14069, t14070, \
t14071, t14072, t14073, t14074, t14075, \
t14076, t14077, t14078, t14079, t14080, \
t14081, t14082, t14083, t14084, t14085, \
t14086, t14087, t14088, t14089, t14090, \
t14091, t14092, t14093, t14094, t14095, \
t14096, t14097, t14098, t14099, t14100, \
t14101, t14102, t14103, t14104, t14105, \
t14106, t14107, t14108, t14109, t14110, \
t14111, t14112, t14113, t14114, t14115, \
t14116, t14117, t14118, t14119, t14120, \
t14121, t14122, t14123, t14124, t14125, \
t14126, t14127, t14128, t14129, t14130, \
t14131, t14132, t14133, t14134, t14135, \
t14136, t14137, t14138, t14139, t14140, \
t14141, t14142, t14143, t14144, t14145, \
t14146, t14147, t14148, t14149, t14150, \
t14151, t14152, t14153, t14154, t14155, \
t14156, t14157, t14158, t14159, t14160, \
t14161, t14162, t14163, t14164, t14165, \
t14166, t14167, t14168, t14169, t14170, \
t14171, t14172, t14173, t14174, t14175, \
t14176, t14177, t14178, t14179, t14180, \
t14181, t14182, t14183, t14184, t14185, \
t14186, t14187, t14188, t14189, t14190, \
t14191, t14192, t14193, t14194, t14195, \
t14196, t14197, t14198, t14199, t14200 )


#########################################################
### init9.ora
#########################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=3
thread=3
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t03001, t03002, t03003, t03004, t03005, \
t03006, t03007, t03008, t03009, t03010, \
t03011, t03012, t03013, t03014, t03015, \
t03016, t03017, t03018, t03019, t03020, \
t03021, t03022, t03023, t03024, t03025, \
t03026, t03027, t03028, t03029, t03030, \
t03031, t03032, t03033, t03034, t03035, \
t03036, t03037, t03038, t03039, t03040, \
t03041, t03042, t03043, t03044, t03045, \
t03046, t03047, t03048, t03049, t03050, \
t03051, t03052, t03053, t03054, t03055, \
t03056, t03057, t03058, t03059, t03060, \
t03061, t03062, t03063, t03064, t03065, \
t03066, t03067, t03068, t03069, t03070, \
t03071, t03072, t03073, t03074, t03075, \
t03076, t03077, t03078, t03079, t03080, \
t03081, t03082, t03083, t03084, t03085, \
t03086, t03087, t03088, t03089, t03090, \
t03091, t03092, t03093, t03094, t03095, \
t03096, t03097, t03098, t03099, t03100, \
t03101, t03102, t03103, t03104, t03105, \
t03106, t03107, t03108, t03109, t03110, \
t03111, t03112, t03113, t03114, t03115, \
```

```
t03116, t03117, t03118, t03119, t03120, \
t03121, t03122, t03123, t03124, t03125, \
t03126, t03127, t03128, t03129, t03130, \
t03131, t03132, t03133, t03134, t03135, \
t03136, t03137, t03138, t03139, t03140, \
t03141, t03142, t03143, t03144, t03145, \
t03146, t03147, t03148, t03149, t03150, \
t03151, t03152, t03153, t03154, t03155, \
t03156, t03157, t03158, t03159, t03160, \
t03161, t03162, t03163, t03164, t03165, \
t03166, t03167, t03168, t03169, t03170, \
t03171, t03172, t03173, t03174, t03175, \
t03176, t03177, t03178, t03179, t03180, \
t03181, t03182, t03183, t03184, t03185, \
t03186, t03187, t03188, t03189, t03190, \
t03191, t03192, t03193, t03194, t03195, \
t03196, t03197, t03198, t03199, t03200 )


######################################################
### init10.ora
######################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=4
thread=4
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t04001, t04002, t04003, t04004, t04005, \
t04006, t04007, t04008, t04009, t04010, \
t04011, t04012, t04013, t04014, t04015, \
t04016, t04017, t04018, t04019, t04020, \
t04021, t04022, t04023, t04024, t04025, \
t04026, t04027, t04028, t04029, t04030, \
t04031, t04032, t04033, t04034, t04035, \
t04036, t04037, t04038, t04039, t04040, \
t04041, t04042, t04043, t04044, t04045, \
t04046, t04047, t04048, t04049, t04050, \
t04051, t04052, t04053, t04054, t04055, \
t04056, t04057, t04058, t04059, t04060, \
t04061, t04062, t04063, t04064, t04065, \
t04066, t04067, t04068, t04069, t04070, \
t04071, t04072, t04073, t04074, t04075, \
t04076, t04077, t04078, t04079, t04080, \
t04081, t04082, t04083, t04084, t04085, \
t04086, t04087, t04088, t04089, t04090, \
t04091, t04092, t04093, t04094, t04095, \
t04096, t04097, t04098, t04099, t04100, \
t04101, t04102, t04103, t04104, t04105, \
t04106, t04107, t04108, t04109, t04110, \
t04111, t04112, t04113, t04114, t04115, \
t04116, t04117, t04118, t04119, t04120, \
t04121, t04122, t04123, t04124, t04125, \
t04126, t04127, t04128, t04129, t04130, \
t04131, t04132, t04133, t04134, t04135, \
t04136, t04137, t04138, t04139, t04140, \
t04141, t04142, t04143, t04144, t04145, \
t04146, t04147, t04148, t04149, t04150, \
t04151, t04152, t04153, t04154, t04155, \
t04156, t04157, t04158, t04159, t04160, \
t04161, t04162, t04163, t04164, t04165, \
t04166, t04167, t04168, t04169, t04170, \
t04171, t04172, t04173, t04174, t04175, \
t04176, t04177, t04178, t04179, t04180, \
t04181, t04182, t04183, t04184, t04185, \
t04186, t04187, t04188, t04189, t04190, \
t04191, t04192, t04193, t04194, t04195, \
t04196, t04197, t04198, t04199, t04200 )


######################################################
### init11.ora
######################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=7
thread=7
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t07001, t07002, t07003, t07004, t07005, \
t07006, t07007, t07008, t07009, t07010, \
t07011, t07012, t07013, t07014, t07015, \
t07016, t07017, t07018, t07019, t07020, \
t07021, t07022, t07023, t07024, t07025, \
t07026, t07027, t07028, t07029, t07030, \
t07031, t07032, t07033, t07034, t07035, \
t07036, t07037, t07038, t07039, t07040, \
t07041, t07042, t07043, t07044, t07045, \
t07046, t07047, t07048, t07049, t07050, \
t07051, t07052, t07053, t07054, t07055, \
t07056, t07057, t07058, t07059, t07060, \
t07061, t07062, t07063, t07064, t07065, \
t07066, t07067, t07068, t07069, t07070, \
t07071, t07072, t07073, t07074, t07075, \
t07076, t07077, t07078, t07079, t07080, \
t07081, t07082, t07083, t07084, t07085, \
t07086, t07087, t07088, t07089, t07090, \
t07091, t07092, t07093, t07094, t07095, \
t07096, t07097, t07098, t07099, t07100, \
t07101, t07102, t07103, t07104, t07105, \
t07106, t07107, t07108, t07109, t07110, \
t07111, t07112, t07113, t07114, t07115, \
t07116, t07117, t07118, t07119, t07120, \
t07121, t07122, t07123, t07124, t07125, \
t07126, t07127, t07128, t07129, t07130, \
t07131, t07132, t07133, t07134, t07135, \
t07136, t07137, t07138, t07139, t07140, \
t07141, t07142, t07143, t07144, t07145, \
t07146, t07147, t07148, t07149, t07150, \
t07151, t07152, t07153, t07154, t07155, \
t07156, t07157, t07158, t07159, t07160, \
t07161, t07162, t07163, t07164, t07165, \
t07166, t07167, t07168, t07169, t07170, \
t07171, t07172, t07173, t07174, t07175, \
t07176, t07177, t07178, t07179, t07180, \
t07181, t07182, t07183, t07184, t07185, \
t07186, t07187, t07188, t07189, t07190, \
t07191, t07192, t07193, t07194, t07195, \
t07196, t07197, t07198, t07199, t07200 )


######################################################
### init12.ora
######################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=8
thread=8
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t08001, t08002, t08003, t08004, t08005, \
t08006, t08007, t08008, t08009, t08010, \
t08011, t08012, t08013, t08014, t08015, \
t08016, t08017, t08018, t08019, t08020, \
t08021, t08022, t08023, t08024, t08025, \
t08026, t08027, t08028, t08029, t08030, \
t08031, t08032, t08033, t08034, t08035, \
t08036, t08037, t08038, t08039, t08040, \
t08041, t08042, t08043, t08044, t08045, \
t08046, t08047, t08048, t08049, t08050, \
t08051, t08052, t08053, t08054, t08055, \
t08056, t08057, t08058, t08059, t08060, \
t08061, t08062, t08063, t08064, t08065, \
t08066, t08067, t08068, t08069, t08070, \
t08071, t08072, t08073, t08074, t08075, \
t08076, t08077, t08078, t08079, t08080, \
t08081, t08082, t08083, t08084, t08085, \
t08086, t08087, t08088, t08089, t08090, \
t08091, t08092, t08093, t08094, t08095, \
t08096, t08097, t08098, t08099, t08100, \
t08101, t08102, t08103, t08104, t08105, \
t08106, t08107, t08108, t08109, t08110, \
t08111, t08112, t08113, t08114, t08115, \
t08116, t08117, t08118, t08119, t08120, \
t08121, t08122, t08123, t08124, t08125, \
t08126, t08127, t08128, t08129, t08130, \
t08131, t08132, t08133, t08134, t08135, \
t08136, t08137, t08138, t08139, t08140, \
t08141, t08142, t08143, t08144, t08145, \
t08146, t08147, t08148, t08149, t08150, \
t08151, t08152, t08153, t08154, t08155, \
t08156, t08157, t08158, t08159, t08160, \
t08161, t08162, t08163, t08164, t08165, \
t08166, t08167, t08168, t08169, t08170, \
t08171, t08172, t08173, t08174, t08175, \
t08176, t08177, t08178, t08179, t08180, \
t08181, t08182, t08183, t08184, t08185, \
t08186, t08187, t08188, t08189, t08190, \
t08191, t08192, t08193, t08194, t08195, \
t08196, t08197, t08198, t08199, t08200 )
```

```
####################################################
### init13.ora
####################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=11
thread=11
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t11001, t11002, t11003, t11004, t11005, \
t11006, t11007, t11008, t11009, t11010, \
t11011, t11012, t11013, t11014, t11015, \
t11016, t11017, t11018, t11019, t11020, \
t11021, t11022, t11023, t11024, t11025, \
t11026, t11027, t11028, t11029, t11030, \
t11031, t11032, t11033, t11034, t11035, \
t11036, t11037, t11038, t11039, t11040, \
t11041, t11042, t11043, t11044, t11045, \
t11046, t11047, t11048, t11049, t11050, \
t11051, t11052, t11053, t11054, t11055, \
t11056, t11057, t11058, t11059, t11060, \
t11061, t11062, t11063, t11064, t11065, \
t11066, t11067, t11068, t11069, t11070, \
t11071, t11072, t11073, t11074, t11075, \
t11076, t11077, t11078, t11079, t11080, \
t11081, t11082, t11083, t11084, t11085, \
t11086, t11087, t11088, t11089, t11090, \
t11091, t11092, t11093, t11094, t11095, \
t11096, t11097, t11098, t11099, t11100, \
t11101, t11102, t11103, t11104, t11105, \
t11106, t11107, t11108, t11109, t11110, \
t11111, t11112, t11113, t11114, t11115, \
t11116, t11117, t11118, t11119, t11120, \
t11121, t11122, t11123, t11124, t11125, \
t11126, t11127, t11128, t11129, t11130, \
t11131, t11132, t11133, t11134, t11135, \
t11136, t11137, t11138, t11139, t11140, \
t11141, t11142, t11143, t11144, t11145, \
t11146, t11147, t11148, t11149, t11150, \
t11151, t11152, t11153, t11154, t11155, \
t11156, t11157, t11158, t11159, t11160, \
t11161, t11162, t11163, t11164, t11165, \
t11166, t11167, t11168, t11169, t11170, \
t11171, t11172, t11173, t11174, t11175, \
t11176, t11177, t11178, t11179, t11180, \
t11181, t11182, t11183, t11184, t11185, \
t11186, t11187, t11188, t11189, t11190, \
t11191, t11192, t11193, t11194, t11195, \
t11196, t11197, t11198, t11199, t11200 )


####################################################
### init14.ora
####################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=12
thread=12
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t12001, t12002, t12003, t12004, t12005, \
t12006, t12007, t12008, t12009, t12010, \
t12011, t12012, t12013, t12014, t12015, \
t12016, t12017, t12018, t12019, t12020, \
t12021, t12022, t12023, t12024, t12025, \
t12026, t12027, t12028, t12029, t12030, \
t12031, t12032, t12033, t12034, t12035, \
t12036, t12037, t12038, t12039, t12040, \
t12041, t12042, t12043, t12044, t12045, \
t12046, t12047, t12048, t12049, t12050, \
t12051, t12052, t12053, t12054, t12055, \
t12056, t12057, t12058, t12059, t12060, \
t12061, t12062, t12063, t12064, t12065, \
t12066, t12067, t12068, t12069, t12070, \
t12071, t12072, t12073, t12074, t12075, \
t12076, t12077, t12078, t12079, t12080, \
t12081, t12082, t12083, t12084, t12085, \
t12086, t12087, t12088, t12089, t12090, \
t12091, t12092, t12093, t12094, t12095, \
t12096, t12097, t12098, t12099, t12100, \
t12101, t12102, t12103, t12104, t12105, \
t12106, t12107, t12108, t12109, t12110, \
t12111, t12112, t12113, t12114, t12115, \
t12116, t12117, t12118, t12119, t12120, \
t12121, t12122, t12123, t12124, t12125, \
t12126, t12127, t12128, t12129, t12130, \
t12131, t12132, t12133, t12134, t12135, \
t12136, t12137, t12138, t12139, t12140, \
t12141, t12142, t12143, t12144, t12145, \
t12146, t12147, t12148, t12149, t12150, \
t12151, t12152, t12153, t12154, t12155, \
t12156, t12157, t12158, t12159, t12160, \
t12161, t12162, t12163, t12164, t12165, \
t12166, t12167, t12168, t12169, t12170, \
t12171, t12172, t12173, t12174, t12175, \
t12176, t12177, t12178, t12179, t12180, \
t12181, t12182, t12183, t12184, t12185, \
t12186, t12187, t12188, t12189, t12190, \
t12191, t12192, t12193, t12194, t12195, \
t12196, t12197, t12198, t12199, t12200 )


####################################################
### init15.ora
####################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=15
thread=15
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t15001, t15002, t15003, t15004, t15005, \
t15006, t15007, t15008, t15009, t15010, \
t15011, t15012, t15013, t15014, t15015, \
t15016, t15017, t15018, t15019, t15020, \
t15021, t15022, t15023, t15024, t15025, \
t15026, t15027, t15028, t15029, t15030, \
t15031, t15032, t15033, t15034, t15035, \
t15036, t15037, t15038, t15039, t15040, \
t15041, t15042, t15043, t15044, t15045, \
t15046, t15047, t15048, t15049, t15050, \
t15051, t15052, t15053, t15054, t15055, \
t15056, t15057, t15058, t15059, t15060, \
t15061, t15062, t15063, t15064, t15065, \
t15066, t15067, t15068, t15069, t15070, \
t15071, t15072, t15073, t15074, t15075, \
t15076, t15077, t15078, t15079, t15080, \
t15081, t15082, t15083, t15084, t15085, \
t15086, t15087, t15088, t15089, t15090, \
t15091, t15092, t15093, t15094, t15095, \
t15096, t15097, t15098, t15099, t15100, \
t15101, t15102, t15103, t15104, t15105, \
t15106, t15107, t15108, t15109, t15110, \
t15111, t15112, t15113, t15114, t15115, \
t15116, t15117, t15118, t15119, t15120, \
t15121, t15122, t15123, t15124, t15125, \
t15126, t15127, t15128, t15129, t15130, \
t15131, t15132, t15133, t15134, t15135, \
t15136, t15137, t15138, t15139, t15140, \
t15141, t15142, t15143, t15144, t15145, \
t15146, t15147, t15148, t15149, t15150, \
t15151, t15152, t15153, t15154, t15155, \
t15156, t15157, t15158, t15159, t15160, \
t15161, t15162, t15163, t15164, t15165, \
t15166, t15167, t15168, t15169, t15170, \
t15171, t15172, t15173, t15174, t15175, \
t15176, t15177, t15178, t15179, t15180, \
t15181, t15182, t15183, t15184, t15185, \
t15186, t15187, t15188, t15189, t15190, \
t15191, t15192, t15193, t15194, t15195, \
t15196, t15197, t15198, t15199, t15200 )


####################################################
### init16.ora
####################################################

db_block_buffers = 1200000
buffer_pool_keep = (buffers:800000, lru_latches:6)
instance_number=16
thread=16
ifile=/usr/oracle/dbs/mult/init.ora
parallel_max_servers = 0
rollback_segments = (
t16001, t16002, t16003, t16004, t16005, \
t16006, t16007, t16008, t16009, t16010, \
t16011, t16012, t16013, t16014, t16015, \
t16016, t16017, t16018, t16019, t16020, \
t16021, t16022, t16023, t16024, t16025, \
t16026, t16027, t16028, t16029, t16030, \
t16031, t16032, t16033, t16034, t16035, \
t16036, t16037, t16038, t16039, t16040, \
```

```
t16041, t16042, t16043, t16044, t16045, \
t16046, t16047, t16048, t16049, t16050, \
t16051, t16052, t16053, t16054, t16055, \
t16056, t16057, t16058, t16059, t16060, \
t16061, t16062, t16063, t16064, t16065, \
t16066, t16067, t16068, t16069, t16070, \
t16071, t16072, t16073, t16074, t16075, \
t16076, t16077, t16078, t16079, t16080, \
t16081, t16082, t16083, t16084, t16085, \
t16086, t16087, t16088, t16089, t16090, \
t16091, t16092, t16093, t16094, t16095, \
t16096, t16097, t16098, t16099, t16100, \
t16101, t16102, t16103, t16104, t16105, \
t16106, t16107, t16108, t16109, t16110, \
t16111, t16112, t16113, t16114, t16115, \
t16116, t16117, t16118, t16119, t16120, \
t16121, t16122, t16123, t16124, t16125, \
t16126, t16127, t16128, t16129, t16130, \
t16131, t16132, t16133, t16134, t16135, \
t16136, t16137, t16138, t16139, t16140, \
t16141, t16142, t16143, t16144, t16145, \
t16146, t16147, t16148, t16149, t16150, \
t16151, t16152, t16153, t16154, t16155, \
t16156, t16157, t16158, t16159, t16160, \
t16161, t16162, t16163, t16164, t16165, \
t16166, t16167, t16168, t16169, t16170, \
t16171, t16172, t16173, t16174, t16175, \
t16176, t16177, t16178, t16179, t16180, \
t16181, t16182, t16183, t16184, t16185, \
t16186, t16187, t16188, t16189, t16190, \
t16191, t16192, t16193, t16194, t16195, \
t16196, t16197, t16198, t16199, t16200 )
```

DYNIX/ptx 4.4.4 OS Tunable Parameters

```
*
* i386 Configuration
*
*
* Tunable Parameters
*
        0       reserved physical memory (RESPHYSMEM)
        1       percent free memory for buffer cache (BUFPCT)
     6500       buffers in buffer cache (NBUF)
     5000       inodes  (NINODE)
  2000000       entries in file table (NFILE)
       32       entries in mount table (NMOUNT)
     2048       entries in proc table (NPROC)
     1736       processes per user id (MAXUP)
        0       hash slots for buffer cache (NHBUF)
        0       buffers for paging I/O (NPBUF)
    10000       number of streams queues (NQUEUE)
    11040       number of streams head structures (NSTREAM)
       10       number of private stream buffers (NBLKPRIV)
     2000       number of 4096 bytes stream buffers (NBLK4096)
    15800       number of 2048 bytes stream buffers (NBLK2048)
     3020       number of 1024 bytes stream buffers (NBLK1024)
     2520       number of 512 bytes stream buffers (NBLK512)
    47104       number of 256 bytes stream buffers (NBLK256)
    67120       number of 128 bytes stream buffers (NBLK128)
    14219       number of 64 bytes stream buffers (NBLK64)
    46170       number of 16 bytes stream buffers (NBLK16)
   196930       number of 4 bytes stream buffers (NBLK4)
    60002       maximum user and group id allowed (C_MAXUID)
       20       number of symbolic link traversals (MAXSYMLINKS)
     1152       entries in mapped file table (NMFILE)
     3000       max number of fifo's (NFIFO)
     8192       max write size to a fifo (FIFOSZ)
     1024       fifo block size (FIFOBSZ)
       20       #of extra 4KB pages for buffer cache (BUFPAGES_INCR)
       40       page-rate threshold (MAXPGIO)
        0       0=>Traditional usage; 1=>POSIX.1 usage (CUSERID_VERSION)
        1       DMA-limit page reserve multiplier (DMABUF_PGRESV_MULT)
       32       DMA-limit page reserve divisor (DMABUF_PGRESV_DIV)
     1024       max #of pending async I/O request per process (MAXAIO)
     1024       #of async I/O buffers, 0-disables async I/O (NABUF)
     2048       #of file descriptor tables (NOFILETAB)
        0       0=>Traditional usage; 1=>POSIX.1 usage (POSIX_HANGUP)
        1       vectored superuser privilege mechanism on/off (PRIV_ENABLE)
        0       mark all locks with engine id, enable checking (P_LOCK_DEBUG)
      300       deadlock detection limit in millions (P_LOCK_LIMIT)
       40       time clock is adjusted on each tick for adjtime (TICKADJ)
        0       #of kernel semaphores to track for statistics (SEMSTATS)
*
* Max. Per-process Resource Limit Tunables
*
 9223372036854775807    CPU usage soft limit (seconds) (SCPULIM)
 9223372036854775807    CPU usage hard limit (seconds) (HCPULIM)
 9223372036854775807    file size soft limit (bytes) (SFSZLIM)
 9223372036854775807    file size hard limit (bytes) (HFSZLIM)
 9223372036854775807    data segment size soft limit (bytes) (SDATLIM)
 9223372036854775807    data segment size hard limit (bytes) (HDATLIM)
 9223372036854775807    stack segment size soft limit (bytes) (SSTKLIM)
 9223372036854775807    stack segment size hard limit (bytes) (HSTKLIM)
 9223372036854775807    core file size soft limit (bytes) (SCORLIM)
 9223372036854775807    core file size hard limit (bytes) (HCORLIM)
        8192            number of open files soft limit (SFNOLIM)
        8192            number of open files hard limit (HFNOLIM)
 9223372036854775807    total virtual space soft limit (bytes)(SVMMLIM)
 9223372036854775807    total virtual space hard limit (bytes)(HVMMLIM)
*
* Disk Quota Tunables
*
        1       entries in disk quota table (NDQUOT)
        1       hash entries for disk quotas (NDQHASH)
   604800       seconds before hard errors for inodes (DQ_FTIMEDEFAULT)
   604800       seconds before hard errors for disk space (DQ_BTIMEDEFAULT)
*
* Vmtune Tunables
*
   786432       max # clusters for Rset (MAXADDR/CLBYTES)
        5       slop for maxRS calculation (default 20*K/NBPG)
        1       extra entries in Usrptmap (INCR_PTSIZE)
       16       min # clusters for Rset (vt_minRS)
   786432       max # clusters for Rset (vt_maxRS)
        5       # HW pages slop in exec (vt_RSexecslop)
        4       Rset multipler (vt_RSexecmult)
        5       Rset divider (vt_RSexecdiv)
        0       low dirty-list size (vt_dirtylow)
        0       high dirty-list size (vt_dirtyhigh)
       32       pageout kluster look-ahead (vt_klout_look)
      100       ticks between PFF adjust (vt_PFFvtime)
        2       pages to drop if PFF < PFFlow (vt_PFFdecr)
        2       low PFF rate; <= PFFhigh (vt_PFFlow)
        5       pages to add if PFF > PFFhigh (vt_PFFincr)
       15       high PFF rate (vt_PFFhigh)
        0       low free-list for swapping (vt_minfree)
        0       high free-list for swapping (vt_desfree)
        0       max dirty-list before swap (vt_maxdirty)
*
* Filename Cache Tunables
*
     2500       filename cache entries (NCSIZE)
      625       filename cache hash entries (NCHSIZE)
*
* Utsname Tunables
*
      4.0       release (REL)
     tmb1       node name (NODE)
DYNIX/ptx       system name (SYS)
   V4.4.4       version (VER)
*
* Streams Tunables
*
    11040       number of multiplexor links (NMUXLINK)
        6       maximum number of pushes allowed (NSTRPUSH)
    22080       initial number of stream event calls (NSTREVENT)
    30288       maximum stream message size (STRMSGSZ)
     1024       max size of ctl part of message (STRCTLSZ)
       80       max low priority block usage (STRLOFRAC)
       90       max medium priority block usage (STRMEDFRAC)
       20       max time to spend serving queues in runqueues  (STRNSCHED)
        2       max grace STRNSCHED (STRNSCHED_GRACE)
      900       max time spent in runqueues() before warning
(STRNSCHED_WARN)
       50       max % of STREAMS buffers in per-engine pools
(STRTARGETFRAC)
       10       max target value for each per-engine STREAMS pool
(STRTARGETMAX)
       10       #of str_ap structures per block (AUTOPUSH_BLOCK_SIZE)
     8280       maximum #of permanent streams messages (NPERM)
*
* IPC Messages
*
       20       entries in msg map (MSGMAP)
     2048       max message size (MSGMAX)
     2048       max bytes on queue (MSGMNB)
       10       message queue identifiers (MSGMNI)
        8       message segment size (MSGSSZ)
       10       system message headers (MSGTQL)
```

```
     128        message segments (MSGSEG)
*
* IPC Semaphores
*
      10        entries in semaphore map (SEMMAP)
      10        semaphore identifiers (SEMMNI)
      60        semaphores in system (SEMMNS)
      50        undo structures in system (SEMMNU)
      25        max semaphores per id (SEMMSL)
      10        max operations per semop call (SEMOPM)
      25        max undo entries per process (SEMUME)
   32767        semaphore maximum value (SEMVMX)
   16384        adjust on exit max value (SEMAEM)
       5        undo structure reclaim threshold (SEMURT
*
* IPC Shared Memory
*
7516192768      max shared memory segment size (SHMMAX)
       1        min shared memory segment size (SHMMIN)
     100        shared memory identifiers (SHMMNI)
    3500        max attached shm segments per process (SHMSEG)
       1        SHM_LOCK operations allowed flag (SHM_LOCK_OK)
       0        min size of segment for SHM_LOCK (SHM_LOCK_MIN)
7516192768      max size of segment for SHM_LOCK (SHM_LOCK_MAX)
       1        implicit SHM_LOCK enabled flag (SHM_LOCKDF_OK)
       0        min size of segment for implicit locking (SHM_LOCKDF_MIN)
7516192768      max size of segment for implicit locking (SHM_LOCKDF_MAX)
      -1        SHM_LOCK effective uid (SHM_LOCK_UID)
      -1        implicit SHM_LOCK effective uid (SHM_LOCKDF_UID)
 2097152        break address pad on 0 address (SHMGAP)
*
* File and Record Locking
*
      64        records configured on system (NFILCK)
*
* Cache affinity tunables
*
       2        div for cache decay, based on cpu speed
(AFFHI_CPUSPEED_DIV)
       1        mult for cache decay, based on cpu speed
(AFFHI_CPUSPEED_MULT)
       1        mult for cache decay, based on cache size
(AFFHI_CACHESZ_MULT)
      30        div for cache decay, based on cache size
(AFFHI_CACHESZ_DIV)
       1        mult for cache warmth, based on cpu speed
(AFFLO_CPUSPEED_MULT)
      12        div for cache warmth, based on cpu speed
(AFFLO_CPUSPEED_DIV)
       1        mult for cache warmth, based on cache size
(AFFLO_CACHESZ_MULT)
     200        div for cache warmth, based on cache size
(AFFLO_CACHESZ_DIV)
*
* Tunables related to UFS
*
       1        chown restricted to super-user. 1-Yes, 0-No
(CHOWN_RESTRICTED)
       1        1-enable parallel reads on regular files,0-not (PARALLEL_IO)
       1        1-sticky bit set on non text files are not cached (STICKYHACK)
*
* Pentium Bugs
*
       0        Enable(1)/Disable(0) Pentium FDIV Bug workaround
(FDIV_BUG)
       0        Enable(1)/Disable(0) Pentium FIST Bug workaround (FIST_BUG)
*
* Tunables related to ACL
*
     582        max number of bytes in an ACL (MAXACLSIZE)
     100        number of in-core auxiliary inodes to allocate (NAUXINODE)
*
* Swap related tunables
*
       2        swap space allocation policy (SWAP_ALLOC_POLICY)
     300        min interval between low swap space messages
(SWAP_LOW_RATE)
       0        min free swap threshold for low swap messages
(SWAP_LOW_WATER)
      60        min interval between out of swap space messages
(SWAP_NONE_RATE)
*
* Tunables of IO devices
*
      64        #of ticlts flow control structs (NTICLTS_FLOW)
      64        max #of ticlts endpoints (NTICLTS)
      64        max #of simultaneous ticotsord connections (NTICOTSORD)
   43296        max timod modules allowed simultaneously (NTLI)
   43296        max #of simultaneous tirdwr instanciations (NTRW)
     100        #of ptem structures (NPTEM)
    2150        max #of bytes to buffer above XOFF threshold
(TTYBUF_LATENCY)
    4096        max #of bytes for a users private kbd map tables (NKBDPVMEM)
      20        default timeout value for kbd timeout mode (KBDTIME)
     100        max #of kbd modules pushed at once on a system (NKBDSTR)
       6        max #of private mapping tables a user can attach (NKBDU)
       8        Target private region in SSI CB
(SSI_MAX_TARGET_PRIVATE)
       8        Bus private region in SSI CB (SSI_MAX_BUS_PRIVATE)
      16        Adapter private region in SSI CB
(SSI_MAX_ADAPTER_PRIVATE)
```

System Configuration

```
SCI
System Configuration:
type    no
MEM      0  on quad 0, shared 3964.0Mb, private 4.0Mb
MEM      1  on quad 1, shared 4092.0Mb, private 4.0Mb
MEM      2  on quad 2, shared 4092.0Mb, private 4.0Mb
MEM      3  on quad 3, shared 4092.0Mb, private 4.0Mb
MEM      4  on quad 4, shared 4092.0Mb, private 4.0Mb
MEM      5  on quad 5, shared 4092.0Mb, private 4.0Mb
MEM      6  on quad 6, shared 4092.0Mb, private 4.0Mb
MEM      7  on quad 7, shared 4092.0Mb, private 4.0Mb
MEM      8  on quad 8, shared 4092.0Mb, private 4.0Mb
MEM      9  on quad 9, shared 4092.0Mb, private 4.0Mb
MEM     10  on quad 10, shared 4092.0Mb, private 4.0Mb
MEM     11  on quad 11, shared 4092.0Mb, private 4.0Mb
MEM     12  on quad 12, shared 4092.0Mb, private 4.0Mb
MEM     13  on quad 13, shared 4092.0Mb, private 4.0Mb
MEM     14  on quad 14, shared 4092.0Mb, private 4.0Mb
MEM     15  on quad 15, shared 4084.0Mb, private 4.0Mb
QLC      0  on quad 0, sw ver 4.40
QLC      1  on quad 0, sw ver 4.40
QLC      2  on quad 1, sw ver 4.40
QLC      3  on quad 1, sw ver 4.40
QLC      4  on quad 2, sw ver 4.40
QLC      5  on quad 2, sw ver 4.40
QLC      6  on quad 3, sw ver 4.40
QLC      7  on quad 3, sw ver 4.40
QLC      8  on quad 4, sw ver 4.40
QLC      9  on quad 4, sw ver 4.40
QLC     10  on quad 5, sw ver 4.40
QLC     11  on quad 5, sw ver 4.40
QLC     12  on quad 6, sw ver 4.40
QLC     13  on quad 6, sw ver 4.40
QLC     14  on quad 7, sw ver 4.40
QLC     15  on quad 7, sw ver 4.40
QLC     16  on quad 8, sw ver 4.40
QLC     17  on quad 8, sw ver 4.40
QLC     18  on quad 9, sw ver 4.40
QLC     19  on quad 9, sw ver 4.40
QLC     20  on quad 10, sw ver 4.40
QLC     21  on quad 10, sw ver 4.40
QLC     22  on quad 11, sw ver 4.40
QLC     23  on quad 11, sw ver 4.40
QLC     24  on quad 12, sw ver 4.40
QLC     25  on quad 12, sw ver 4.40
QLC     26  on quad 13, sw ver 4.40
QLC     27  on quad 13, sw ver 4.40
QLC     28  on quad 14, sw ver 4.40
QLC     29  on quad 14, sw ver 4.40
QLC     30  on quad 15, sw ver 4.40
QLC     31  on quad 15, sw ver 4.40
PE       0  on quad 0, hw ver 2.2
PE       1  on quad 0, hw ver 2.2
PE       2  on quad 0, hw ver 2.2
PE       3  on quad 0, hw ver 2.2
PE       4  on quad 1, hw ver 2.1
PE       5  on quad 2, hw ver 2.1
PE       6  on quad 3, hw ver 2.1
PE       7  on quad 4, hw ver 2.1
PE       8  on quad 5, hw ver 2.1
PE       9  on quad 6, hw ver 2.1
PE      10  on quad 7, hw ver 2.1
PE      11  on quad 8, hw ver 2.1
PE      12  on quad 9, hw ver 2.1
PE      13  on quad 10, hw ver 2.1
PE      14  on quad 11, hw ver 2.1
PE      15  on quad 12, hw ver 2.2
PE      16  on quad 13, hw ver 2.1
PE      17  on quad 14, hw ver 2.1
PE      18  on quad 15, hw ver 2.1
FF       0  on quad 0, hw ver 4, sw ver FF2.12 0
FF       1  on quad 0, hw ver 4, sw ver FF2.12 0
```

```
FF      4  on quad 1, hw ver 4, sw ver FF2.12 0
FF      5  on quad 1, hw ver 4, sw ver FF2.12 0
FF     10  on quad 2, hw ver 4, sw ver FF2.12 0
FF     66  on quad 2, hw ver 4, sw ver FF2.12 0
FF     15  on quad 3, hw ver 4, sw ver FF2.12 0
FF     68  on quad 3, hw ver 4, sw ver FF2.12 0
FF     17  on quad 4, hw ver 4, sw ver FF2.12 0
FF     69  on quad 4, hw ver 4, sw ver FF2.12 0
FF     21  on quad 5, hw ver 4, sw ver FF2.12 0
FF     71  on quad 5, hw ver 4, sw ver FF2.12 0
FF     26  on quad 6, hw ver 4, sw ver FF2.12 0
FF     27  on quad 6, hw ver 4, sw ver FF2.12 0
FF     30  on quad 7, hw ver 4, sw ver FF2.12 0
FF     31  on quad 7, hw ver 4, sw ver FF2.12 0
FF     72  on quad 8, hw ver 4, sw ver FF2.12 0
FF     73  on quad 8, hw ver 4, sw ver FF2.12 0
FF     75  on quad 9, hw ver 4, sw ver FF2.12 0
FF     76  on quad 9, hw ver 4, sw ver FF2.12 0
FF     43  on quad 10, hw ver 4, sw ver FF2.12 0
FF     80  on quad 10, hw ver 4, sw ver FF2.12 0
FF     46  on quad 11, hw ver 4, sw ver FF2.12 0
FF     47  on quad 11, hw ver 4, sw ver FF2.12 0
FF     49  on quad 12, hw ver 4, sw ver FF2.12 0
FF     82  on quad 12, hw ver 4, sw ver FF2.12 0
FF     84  on quad 13, hw ver 4, sw ver FF2.12 0
FF     85  on quad 13, hw ver 4, sw ver FF2.12 0
FF     90  on quad 14, hw ver 4, sw ver FF2.12 0
FF     91  on quad 14, hw ver 4, sw ver FF2.12 0
FF     62  on quad 15, hw ver 4, sw ver FF2.12 0
FF     93  on quad 15, hw ver 4, sw ver FF2.12 0
PROC/P6   0  on quad 0, 405MHz, Apic: 1
PROC/P6   1  on quad 0, 406MHz, Apic: 8
PROC/P6   2  on quad 0, 405MHz, Apic: 2
PROC/P6   3  on quad 0, 405MHz, Apic: 4
PROC/P6   4  on quad 1, 405MHz, Apic: 17
PROC/P6   5  on quad 1, 405MHz, Apic: 24
PROC/P6   6  on quad 1, 405MHz, Apic: 18
PROC/P6   7  on quad 1, 405MHz, Apic: 20
PROC/P6   8  on quad 2, 405MHz, Apic: 33
PROC/P6   9  on quad 2, 405MHz, Apic: 40
PROC/P6  10  on quad 2, 405MHz, Apic: 34
PROC/P6  11  on quad 2, 405MHz, Apic: 36
PROC/P6  12  on quad 3, 405MHz, Apic: 49
PROC/P6  13  on quad 3, 405MHz, Apic: 56
PROC/P6  14  on quad 3, 405MHz, Apic: 50
PROC/P6  15  on quad 3, 405MHz, Apic: 52
PROC/P6  16  on quad 4, 405MHz, Apic: 65
PROC/P6  17  on quad 4, 405MHz, Apic: 72
PROC/P6  18  on quad 4, 405MHz, Apic: 66
PROC/P6  19  on quad 4, 405MHz, Apic: 68
PROC/P6  20  on quad 5, 405MHz, Apic: 81
PROC/P6  21  on quad 5, 405MHz, Apic: 88
PROC/P6  22  on quad 5, 405MHz, Apic: 82
PROC/P6  23  on quad 5, 405MHz, Apic: 84
PROC/P6  24  on quad 6, 405MHz, Apic: 97
PROC/P6  25  on quad 6, 405MHz, Apic: 104
PROC/P6  26  on quad 6, 405MHz, Apic: 98
PROC/P6  27  on quad 6, 405MHz, Apic: 100
PROC/P6  28  on quad 7, 405MHz, Apic: 113
PROC/P6  29  on quad 7, 405MHz, Apic: 120
PROC/P6  30  on quad 7, 405MHz, Apic: 114
PROC/P6  31  on quad 7, 405MHz, Apic: 116
PROC/P6  32  on quad 8, 405MHz, Apic: 129
PROC/P6  33  on quad 8, 405MHz, Apic: 136
PROC/P6  34  on quad 8, 405MHz, Apic: 130
PROC/P6  35  on quad 8, 405MHz, Apic: 132
PROC/P6  36  on quad 9, 405MHz, Apic: 145
PROC/P6  37  on quad 9, 405MHz, Apic: 152
PROC/P6  38  on quad 9, 405MHz, Apic: 146
PROC/P6  39  on quad 9, 405MHz, Apic: 148
PROC/P6  40  on quad 10, 405MHz, Apic: 161
PROC/P6  41  on quad 10, 405MHz, Apic: 168
PROC/P6  42  on quad 10, 405MHz, Apic: 162
PROC/P6  43  on quad 10, 405MHz, Apic: 164
PROC/P6  44  on quad 11, 405MHz, Apic: 177
PROC/P6  45  on quad 11, 405MHz, Apic: 184
PROC/P6  46  on quad 11, 405MHz, Apic: 178
PROC/P6  47  on quad 11, 405MHz, Apic: 180
PROC/P6  48  on quad 12, 405MHz, Apic: 193
PROC/P6  49  on quad 12, 405MHz, Apic: 200
PROC/P6  50  on quad 12, 405MHz, Apic: 194
PROC/P6  51  on quad 12, 405MHz, Apic: 196
PROC/P6  52  on quad 13, 405MHz, Apic: 209
PROC/P6  53  on quad 13, 405MHz, Apic: 216
PROC/P6  54  on quad 13, 405MHz, Apic: 210
PROC/P6  55  on quad 13, 405MHz, Apic: 212
PROC/P6  56  on quad 14, 405MHz, Apic: 225
PROC/P6  57  on quad 14, 405MHz, Apic: 232
PROC/P6  58  on quad 14, 405MHz, Apic: 226
PROC/P6  59  on quad 14, 405MHz, Apic: 228
PROC/P6  60  on quad 15, 405MHz, Apic: 241
PROC/P6  61  on quad 15, 405MHz, Apic: 248
PROC/P6  62  on quad 15, 405MHz, Apic: 242
PROC/P6  63  on quad 15, 405MHz, Apic: 244
nodeid: 0xfacade
masterid: none
```

Disk Configuration Dump

```
NAME       CFGTYPE  DEVNUM  UNIT       FLAGS  OnBUS    OnDEVICE
quad0      quad        0  0x00000000  L      sci      sci0
asy0       asy         0  0x00000000  L      eisa     quad0
asy1       asy         1  0x00000001  L      eisa     quad0
mdc0       mdc         0  0x00000000  L      eisa     quad0
ff0        ff          0  0x00000000  SP     pci      quad0
fabric0    fabric      0  0x00fe5800  SM     fc       ff0
fcbr0      fcbr        0  0x00fe0000  SP     fcp      fabric0
scsibus0   scsibus     0  0x00000070  SM     mscsi    fcbr0
pbay0      pbay        0  0x00000100  S      scsi     scsibus0
sd3000     sd          1  0x00000000  S      scsi     scsibus0
sd3001     sd          2  0x00000010  S      scsi     scsibus0
sd3002     sd          3  0x00000020  S      scsi     scsibus0
sd3003     sd          4  0x00000030  S      scsi     scsibus0
sd3004     sd          5  0x00000040  S      scsi     scsibus0
sd3005     sd          6  0x00000050  S      scsi     scsibus0
scsibus1   scsibus     1  0x01000070  SM     mscsi    fcbr0
pbay1      pbay        1  0x00000100  S      scsi     scsibus1
sd3006     sd          7  0x00000000  S      scsi     scsibus1
sd3007     sd          8  0x00000010  S      scsi     scsibus1
sd3008     sd          9  0x00000020  S      scsi     scsibus1
sd3009     sd         10  0x00000030  S      scsi     scsibus1
sd3010     sd         11  0x00000040  S      scsi     scsibus1
sd3011     sd         12  0x00000050  S      scsi     scsibus1
scsibus2   scsibus     2  0x02000060  SM     mscsi    fcbr0
pbay2      pbay        2  0x00000100  S      scsi     scsibus2
sd3012     sd         13  0x00000000  S      scsi     scsibus2
sd3013     sd         14  0x00000010  S      scsi     scsibus2
sd3014     sd         15  0x00000020  S      scsi     scsibus2
sd3015     sd         16  0x00000030  S      scsi     scsibus2
sd3016     sd         17  0x00000040  S      scsi     scsibus2
sd3017     sd         18  0x00000050  S      scsi     scsibus2
scsibus3   scsibus     3  0x03000060  SM     mscsi    fcbr0
pbay3      pbay        3  0x00000100  S      scsi     scsibus3
sd3018     sd         19  0x00000000  S      scsi     scsibus3
sd3019     sd         20  0x00000010  S      scsi     scsibus3
sd3020     sd         21  0x00000020  S      scsi     scsibus3
sd3021     sd         22  0x00000030  S      scsi     scsibus3
sd3022     sd         23  0x00000040  S      scsi     scsibus3
sd3023     sd         24  0x00000050  S      scsi     scsibus3
scsibus4   scsibus     4  0x04000070  SM     mscsi    fcbr0
pbay4      pbay        4  0x00000100  S      scsi     scsibus4
sd3024     sd         25  0x00000000  S      scsi     scsibus4
sd3025     sd         26  0x00000010  S      scsi     scsibus4
sd3026     sd         27  0x00000020  S      scsi     scsibus4
sd3027     sd         28  0x00000030  S      scsi     scsibus4
sd3028     sd         29  0x00000040  S      scsi     scsibus4
sd3029     sd         30  0x00000050  S      scsi     scsibus4
scsibus5   scsibus     5  0x05000070  SM     mscsi    fcbr0
pbay5      pbay        5  0x00000100  S      scsi     scsibus5
sd3030     sd         31  0x00000000  S      scsi     scsibus5
sd3031     sd         32  0x00000010  S      scsi     scsibus5
sd3032     sd         33  0x00000020  S      scsi     scsibus5
sd3033     sd         34  0x00000030  S      scsi     scsibus5
sd3034     sd         35  0x00000040  S      scsi     scsibus5
sd3035     sd         36  0x00000050  S      scsi     scsibus5
scsibus6   scsibus     6  0x06000060  SM     mscsi    fcbr0
pbay6      pbay        6  0x00000100  S      scsi     scsibus6
sd3036     sd         37  0x00000000  S      scsi     scsibus6
sd3037     sd         38  0x00000010  S      scsi     scsibus6
sd3038     sd         39  0x00000020  S      scsi     scsibus6
sd3039     sd         40  0x00000030  S      scsi     scsibus6
sd3040     sd         41  0x00000040  S      scsi     scsibus6
sd3041     sd         42  0x00000050  S      scsi     scsibus6
scsibus7   scsibus     7  0x07000060  SM     mscsi    fcbr0
pbay7      pbay        7  0x00000100  S      scsi     scsibus7
sd3042     sd         43  0x00000000  S      scsi     scsibus7
sd3043     sd         44  0x00000010  S      scsi     scsibus7
sd3044     sd         45  0x00000020  S      scsi     scsibus7
sd3045     sd         46  0x00000030  S      scsi     scsibus7
sd3046     sd         47  0x00000040  S      scsi     scsibus7
sd3047     sd         48  0x00000050  S      scsi     scsibus7
fcbr1      fcbr        1  0x00fe0800  SP     fcp      fabric0
scsibus8   scsibus     8  0x00000070  SM     mscsi    fcbr1
pbay8      pbay        8  0x00000100  S      scsi     scsibus8
sd3048     sd         49  0x00000000  S      scsi     scsibus8
sd3049     sd         50  0x00000010  S      scsi     scsibus8
```

Appendix C – Tunable Parameters

```
sd3050    sd       51  0x00000020  S    scsi   scsibus8
sd3051    sd       52  0x00000030  S    scsi   scsibus8
sd3052    sd       53  0x00000040  S    scsi   scsibus8
sd3053    sd       54  0x00000050  S    scsi   scsibus8
scsibus9  scsibus   9  0x01000070  SM   mscsi  fcbr1
pbay9     pbay      9  0x00000100  S    scsi   scsibus9
sd3054    sd       55  0x00000000  S    scsi   scsibus9
sd3055    sd       56  0x00000010  S    scsi   scsibus9
sd3056    sd       57  0x00000020  S    scsi   scsibus9
sd3057    sd       58  0x00000030  S    scsi   scsibus9
sd3058    sd       59  0x00000040  S    scsi   scsibus9
sd3059    sd       60  0x00000050  S    scsi   scsibus9
scsibus10 scsibus  10  0x02000060  SM   mscsi  fcbr1
pbay10    pbay     10  0x00000100  S    scsi   scsibus10
sd3060    sd       61  0x00000000  S    scsi   scsibus10
sd3061    sd       62  0x00000010  S    scsi   scsibus10
sd3062    sd       63  0x00000020  S    scsi   scsibus10
sd3063    sd       64  0x00000030  S    scsi   scsibus10
sd3064    sd       65  0x00000040  S    scsi   scsibus10
sd3065    sd       66  0x00000050  S    scsi   scsibus10
scsibus11 scsibus  11  0x03000060  SM   mscsi  fcbr1
pbay11    pbay     11  0x00000100  S    scsi   scsibus11
sd3066    sd       67  0x00000000  S    scsi   scsibus11
sd3067    sd       68  0x00000010  S    scsi   scsibus11
sd3068    sd       69  0x00000020  S    scsi   scsibus11
sd3069    sd       70  0x00000030  S    scsi   scsibus11
sd3070    sd       71  0x00000040  S    scsi   scsibus11
sd3071    sd       72  0x00000050  S    scsi   scsibus11
scsibus12 scsibus  12  0x04000070  SM   mscsi  fcbr1
pbay12    pbay     12  0x00000100  S    scsi   scsibus12
sd3072    sd       73  0x00000000  S    scsi   scsibus12
sd3073    sd       74  0x00000010  S    scsi   scsibus12
sd3074    sd       75  0x00000020  S    scsi   scsibus12
sd3075    sd       76  0x00000030  S    scsi   scsibus12
sd3076    sd       77  0x00000040  S    scsi   scsibus12
sd3077    sd       78  0x00000050  S    scsi   scsibus12
scsibus13 scsibus  13  0x05000070  SM   mscsi  fcbr1
pbay13    pbay     13  0x00000100  S    scsi   scsibus13
sd3078    sd       79  0x00000000  S    scsi   scsibus13
sd3079    sd       80  0x00000010  S    scsi   scsibus13
sd3080    sd       81  0x00000020  S    scsi   scsibus13
sd3081    sd       82  0x00000030  S    scsi   scsibus13
sd3082    sd       83  0x00000040  S    scsi   scsibus13
sd3083    sd       84  0x00000050  S    scsi   scsibus13
scsibus14 scsibus  14  0x06000060  SM   mscsi  fcbr1
pbay14    pbay     14  0x00000100  S    scsi   scsibus14
sd3084    sd       85  0x00000000  S    scsi   scsibus14
sd3085    sd       86  0x00000010  S    scsi   scsibus14
sd3086    sd       87  0x00000020  S    scsi   scsibus14
sd3087    sd       88  0x00000030  S    scsi   scsibus14
sd3088    sd       89  0x00000040  S    scsi   scsibus14
sd3089    sd       90  0x00000050  S    scsi   scsibus14
scsibus15 scsibus  15  0x07000060  SM   mscsi  fcbr1
pbay15    pbay     15  0x00000100  S    scsi   scsibus15
sd3090    sd       91  0x00000000  S    scsi   scsibus15
sd3091    sd       92  0x00000010  S    scsi   scsibus15
sd3092    sd       93  0x00000020  S    scsi   scsibus15
sd3093    sd       94  0x00000030  S    scsi   scsibus15
sd3094    sd       95  0x00000040  S    scsi   scsibus15
sd3095    sd       96  0x00000050  S    scsi   scsibus15
fcbr2     fcbr      2  0x00fe1000  SP   fcp    fabric0
scsibus16 scsibus  16  0x00000070  SM   mscsi  fcbr2
pbay16    pbay     16  0x00000100  S    scsi   scsibus16
sd3096    sd       97  0x00000000  S    scsi   scsibus16
sd3097    sd       98  0x00000010  S    scsi   scsibus16
sd3098    sd       99  0x00000020  S    scsi   scsibus16
sd3099    sd      100  0x00000030  S    scsi   scsibus16
sd3100    sd      101  0x00000040  S    scsi   scsibus16
sd3101    sd      102  0x00000050  S    scsi   scsibus16
scsibus17 scsibus  17  0x01000070  SM   mscsi  fcbr2
pbay17    pbay     17  0x00000100  S    scsi   scsibus17
sd3102    sd      103  0x00000000  S    scsi   scsibus17
sd3103    sd      104  0x00000010  S    scsi   scsibus17
sd3104    sd      105  0x00000020  S    scsi   scsibus17
sd3105    sd      106  0x00000030  S    scsi   scsibus17
sd3106    sd      107  0x00000040  S    scsi   scsibus17
sd3107    sd      108  0x00000050  S    scsi   scsibus17
scsibus18 scsibus  18  0x02000060  SM   mscsi  fcbr2
pbay18    pbay     18  0x00000100  S    scsi   scsibus18
sd3108    sd      109  0x00000000  S    scsi   scsibus18
sd3109    sd      110  0x00000010  S    scsi   scsibus18
sd3110    sd      111  0x00000020  S    scsi   scsibus18
sd3111    sd      112  0x00000030  S    scsi   scsibus18
sd3112    sd      113  0x00000040  S    scsi   scsibus18
sd3113    sd      114  0x00000050  S    scsi   scsibus18
scsibus19 scsibus  19  0x03000060  SM   mscsi  fcbr2
pbay19    pbay     19  0x00000100  S    scsi   scsibus19
sd3114    sd      115  0x00000000  S    scsi   scsibus19
sd3115    sd      116  0x00000010  S    scsi   scsibus19
sd3116    sd      117  0x00000020  S    scsi   scsibus19
sd3117    sd      118  0x00000030  S    scsi   scsibus19
sd3118    sd      119  0x00000040  S    scsi   scsibus19
sd3119    sd      120  0x00000050  S    scsi   scsibus19
scsibus20 scsibus  20  0x04000070  SM   mscsi  fcbr2
pbay20    pbay     20  0x00000100  S    scsi   scsibus20
sd3120    sd      121  0x00000000  S    scsi   scsibus20
sd3121    sd      122  0x00000010  S    scsi   scsibus20
sd3122    sd      123  0x00000020  S    scsi   scsibus20
sd3123    sd      124  0x00000030  S    scsi   scsibus20
sd3124    sd      125  0x00000040  S    scsi   scsibus20
sd3125    sd      126  0x00000050  S    scsi   scsibus20
scsibus21 scsibus  21  0x05000070  SM   mscsi  fcbr2
pbay21    pbay     21  0x00000100  S    scsi   scsibus21
sd3126    sd      127  0x00000000  S    scsi   scsibus21
sd3127    sd      128  0x00000010  S    scsi   scsibus21
sd3128    sd      129  0x00000020  S    scsi   scsibus21
sd3129    sd      130  0x00000030  S    scsi   scsibus21
sd3130    sd      131  0x00000040  S    scsi   scsibus21
sd3131    sd      132  0x00000050  S    scsi   scsibus21
scsibus22 scsibus  22  0x06000060  SM   mscsi  fcbr2
pbay22    pbay     22  0x00000100  S    scsi   scsibus22
sd3132    sd      133  0x00000000  S    scsi   scsibus22
sd3133    sd      134  0x00000010  S    scsi   scsibus22
sd3134    sd      135  0x00000020  S    scsi   scsibus22
sd3135    sd      136  0x00000030  S    scsi   scsibus22
sd3136    sd      137  0x00000040  S    scsi   scsibus22
sd3137    sd      138  0x00000050  S    scsi   scsibus22
scsibus23 scsibus  23  0x07000060  SM   mscsi  fcbr2
pbay23    pbay     23  0x00000100  S    scsi   scsibus23
sd3138    sd      139  0x00000000  S    scsi   scsibus23
sd3139    sd      140  0x00000010  S    scsi   scsibus23
sd3140    sd      141  0x00000020  S    scsi   scsibus23
sd3141    sd      142  0x00000030  S    scsi   scsibus23
sd3142    sd      143  0x00000040  S    scsi   scsibus23
sd3143    sd      144  0x00000050  S    scsi   scsibus23
fcbr3     fcbr      3  0x00fe1800  SP   fcp    fabric0
scsibus24 scsibus  24  0x00000070  SM   mscsi  fcbr3
pbay24    pbay     24  0x00000100  S    scsi   scsibus24
sd3144    sd      145  0x00000000  S    scsi   scsibus24
sd3145    sd      146  0x00000010  S    scsi   scsibus24
sd3146    sd      147  0x00000020  S    scsi   scsibus24
sd3147    sd      148  0x00000030  S    scsi   scsibus24
sd3148    sd      149  0x00000040  S    scsi   scsibus24
sd3149    sd      150  0x00000050  S    scsi   scsibus24
scsibus25 scsibus  25  0x01000070  SM   mscsi  fcbr3
pbay25    pbay     25  0x00000100  S    scsi   scsibus25
sd3150    sd      151  0x00000000  S    scsi   scsibus25
sd3151    sd      152  0x00000010  S    scsi   scsibus25
sd3152    sd      153  0x00000020  S    scsi   scsibus25
sd3153    sd      154  0x00000030  S    scsi   scsibus25
sd3154    sd      155  0x00000040  S    scsi   scsibus25
sd3155    sd      156  0x00000050  S    scsi   scsibus25
scsibus26 scsibus  26  0x02000060  SM   mscsi  fcbr3
pbay26    pbay     26  0x00000100  S    scsi   scsibus26
sd3156    sd      157  0x00000000  S    scsi   scsibus26
sd3157    sd      158  0x00000010  S    scsi   scsibus26
sd3158    sd      159  0x00000020  S    scsi   scsibus26
sd3159    sd      160  0x00000030  S    scsi   scsibus26
sd3160    sd      161  0x00000040  S    scsi   scsibus26
sd3161    sd      162  0x00000050  S    scsi   scsibus26
scsibus27 scsibus  27  0x03000060  SM   mscsi  fcbr3
pbay27    pbay     27  0x00000100  S    scsi   scsibus27
sd3162    sd      163  0x00000000  S    scsi   scsibus27
sd3163    sd      164  0x00000010  S    scsi   scsibus27
sd3164    sd      165  0x00000020  S    scsi   scsibus27
sd3165    sd      166  0x00000030  S    scsi   scsibus27
sd3166    sd      167  0x00000040  S    scsi   scsibus27
sd3167    sd      168  0x00000050  S    scsi   scsibus27
scsibus28 scsibus  28  0x04000070  SM   mscsi  fcbr3
pbay28    pbay     28  0x00000100  S    scsi   scsibus28
sd3168    sd      169  0x00000000  S    scsi   scsibus28
sd3169    sd      170  0x00000010  S    scsi   scsibus28
sd3170    sd      171  0x00000020  S    scsi   scsibus28
sd3171    sd      172  0x00000030  S    scsi   scsibus28
sd3172    sd      173  0x00000040  S    scsi   scsibus28
sd3173    sd      174  0x00000050  S    scsi   scsibus28
scsibus29 scsibus  29  0x05000070  SM   mscsi  fcbr3
pbay29    pbay     29  0x00000100  S    scsi   scsibus29
sd3174    sd      175  0x00000000  S    scsi   scsibus29
sd3175    sd      176  0x00000010  S    scsi   scsibus29
sd3176    sd      177  0x00000020  S    scsi   scsibus29
sd3177    sd      178  0x00000030  S    scsi   scsibus29
sd3178    sd      179  0x00000040  S    scsi   scsibus29
sd3179    sd      180  0x00000050  S    scsi   scsibus29
scsibus30 scsibus  30  0x06000060  SM   mscsi  fcbr3
pbay30    pbay     30  0x00000100  S    scsi   scsibus30
sd3180    sd      181  0x00000000  S    scsi   scsibus30
sd3181    sd      182  0x00000010  S    scsi   scsibus30
```

Appendix C – Tunable Parameters

```
sd3182    sd       183  0x00000020  S    scsi   scsibus30
sd3183    sd       184  0x00000030  S    scsi   scsibus30
sd3184    sd       185  0x00000040  S    scsi   scsibus30
sd3185    sd       186  0x00000050  S    scsi   scsibus30
scsibus31 scsibus   31  0x07000060  SM   mscsi  fcbr3
pbay31    pbay      31  0x00000100  S    scsi   scsibus31
sd3186    sd       187  0x00000000  S    scsi   scsibus31
sd3187    sd       188  0x00000010  S    scsi   scsibus31
sd3188    sd       189  0x00000020  S    scsi   scsibus31
sd3189    sd       190  0x00000030  S    scsi   scsibus31
sd3190    sd       191  0x00000040  S    scsi   scsibus31
sd3191    sd       192  0x00000050  S    scsi   scsibus31
fcbr4     fcbr       4  0x00fe2000  SP   fcp    fabric0
scsibus32 scsibus   32  0x00000070  SM   mscsi  fcbr4
pbay32    pbay      32  0x00000100  S    scsi   scsibus32
sd3192    sd       193  0x00000000  S    scsi   scsibus32
sd3193    sd       194  0x00000010  S    scsi   scsibus32
sd3194    sd       195  0x00000020  S    scsi   scsibus32
sd3195    sd       196  0x00000030  S    scsi   scsibus32
sd3196    sd       197  0x00000040  S    scsi   scsibus32
sd3197    sd       198  0x00000050  S    scsi   scsibus32
scsibus33 scsibus   33  0x01000070  SM   mscsi  fcbr4
pbay33    pbay      33  0x00000100  S    scsi   scsibus33
sd3198    sd       199  0x00000000  S    scsi   scsibus33
sd3199    sd       200  0x00000010  S    scsi   scsibus33
sd3200    sd       201  0x00000020  S    scsi   scsibus33
sd3201    sd       202  0x00000030  S    scsi   scsibus33
sd3202    sd       203  0x00000040  S    scsi   scsibus33
sd3203    sd       204  0x00000050  S    scsi   scsibus33
scsibus34 scsibus   34  0x02000060  SM   mscsi  fcbr4
pbay34    pbay      34  0x00000100  S    scsi   scsibus34
sd3204    sd       205  0x00000000  S    scsi   scsibus34
sd3205    sd       206  0x00000010  S    scsi   scsibus34
sd3206    sd       207  0x00000020  S    scsi   scsibus34
sd3207    sd       208  0x00000030  S    scsi   scsibus34
sd3208    sd       209  0x00000040  S    scsi   scsibus34
sd3209    sd       210  0x00000050  S    scsi   scsibus34
scsibus35 scsibus   35  0x03000060  SM   mscsi  fcbr4
pbay35    pbay      35  0x00000100  S    scsi   scsibus35
sd3210    sd       211  0x00000000  S    scsi   scsibus35
sd3211    sd       212  0x00000010  S    scsi   scsibus35
sd3212    sd       213  0x00000020  S    scsi   scsibus35
sd3213    sd       214  0x00000030  S    scsi   scsibus35
sd3214    sd       215  0x00000040  S    scsi   scsibus35
sd3215    sd       216  0x00000050  S    scsi   scsibus35
scsibus36 scsibus   36  0x04000070  SM   mscsi  fcbr4
pbay36    pbay      36  0x00000100  S    scsi   scsibus36
sd3216    sd       217  0x00000000  S    scsi   scsibus36
sd3217    sd       218  0x00000010  S    scsi   scsibus36
sd3218    sd       219  0x00000020  S    scsi   scsibus36
sd3219    sd       220  0x00000030  S    scsi   scsibus36
sd3220    sd       221  0x00000040  S    scsi   scsibus36
sd3221    sd       222  0x00000050  S    scsi   scsibus36
scsibus37 scsibus   37  0x05000070  SM   mscsi  fcbr4
pbay37    pbay      37  0x00000100  S    scsi   scsibus37
sd3222    sd       223  0x00000000  S    scsi   scsibus37
sd3223    sd       224  0x00000010  S    scsi   scsibus37
sd3224    sd       225  0x00000020  S    scsi   scsibus37
sd3225    sd       226  0x00000030  S    scsi   scsibus37
sd3226    sd       227  0x00000040  S    scsi   scsibus37
sd3227    sd       228  0x00000050  S    scsi   scsibus37
scsibus38 scsibus   38  0x06000060  SM   mscsi  fcbr4
pbay38    pbay      38  0x00000100  S    scsi   scsibus38
sd3228    sd       229  0x00000000  S    scsi   scsibus38
sd3229    sd       230  0x00000010  S    scsi   scsibus38
sd3230    sd       231  0x00000020  S    scsi   scsibus38
sd3231    sd       232  0x00000030  S    scsi   scsibus38
sd3232    sd       233  0x00000040  S    scsi   scsibus38
sd3233    sd       234  0x00000050  S    scsi   scsibus38
scsibus39 scsibus   39  0x07000060  SM   mscsi  fcbr4
pbay39    pbay      39  0x00000100  S    scsi   scsibus39
sd3234    sd       235  0x00000000  S    scsi   scsibus39
sd3235    sd       236  0x00000010  S    scsi   scsibus39
sd3236    sd       237  0x00000020  S    scsi   scsibus39
sd3237    sd       238  0x00000030  S    scsi   scsibus39
sd3238    sd       239  0x00000040  S    scsi   scsibus39
sd3239    sd       240  0x00000050  S    scsi   scsibus39
fcbr5     fcbr       5  0x00fe2800  SP   fcp    fabric0
scsibus40 scsibus   40  0x00000070  SM   mscsi  fcbr5
pbay40    pbay      40  0x00000100  S    scsi   scsibus40
sd3240    sd       241  0x00000000  S    scsi   scsibus40
sd3241    sd       242  0x00000010  S    scsi   scsibus40
sd3242    sd       243  0x00000020  S    scsi   scsibus40
sd3243    sd       244  0x00000030  S    scsi   scsibus40
sd3244    sd       245  0x00000040  S    scsi   scsibus40
sd3245    sd       246  0x00000050  S    scsi   scsibus40
scsibus41 scsibus   41  0x01000070  SM   mscsi  fcbr5
pbay41    pbay      41  0x00000100  S    scsi   scsibus41
sd3246    sd       247  0x00000000  S    scsi   scsibus41
sd3247    sd       248  0x00000010  S    scsi   scsibus41
sd3248    sd       249  0x00000020  S    scsi   scsibus41
sd3249    sd       250  0x00000030  S    scsi   scsibus41
sd3250    sd       251  0x00000040  S    scsi   scsibus41
sd3251    sd       252  0x00000050  S    scsi   scsibus41
scsibus42 scsibus   42  0x02000060  SM   mscsi  fcbr5
pbay42    pbay      42  0x00000100  S    scsi   scsibus42
sd3252    sd       253  0x00000000  S    scsi   scsibus42
sd3253    sd       254  0x00000010  S    scsi   scsibus42
sd3254    sd       255  0x00000020  S    scsi   scsibus42
sd3255    sd       256  0x00000030  S    scsi   scsibus42
sd3256    sd       257  0x00000040  S    scsi   scsibus42
sd3257    sd       258  0x00000050  S    scsi   scsibus42
scsibus43 scsibus   43  0x03000060  SM   mscsi  fcbr5
pbay43    pbay      43  0x00000100  S    scsi   scsibus43
sd3258    sd       259  0x00000000  S    scsi   scsibus43
sd3259    sd       260  0x00000010  S    scsi   scsibus43
sd3260    sd       261  0x00000020  S    scsi   scsibus43
sd3261    sd       262  0x00000030  S    scsi   scsibus43
sd3262    sd       263  0x00000040  S    scsi   scsibus43
sd3263    sd       264  0x00000050  S    scsi   scsibus43
scsibus44 scsibus   44  0x04000070  SM   mscsi  fcbr5
pbay44    pbay      44  0x00000100  S    scsi   scsibus44
sd3264    sd       265  0x00000000  S    scsi   scsibus44
sd3265    sd       266  0x00000010  S    scsi   scsibus44
sd3266    sd       267  0x00000020  S    scsi   scsibus44
sd3267    sd       268  0x00000030  S    scsi   scsibus44
sd3268    sd       269  0x00000040  S    scsi   scsibus44
sd3269    sd       270  0x00000050  S    scsi   scsibus44
scsibus45 scsibus   45  0x05000070  SM   mscsi  fcbr5
pbay45    pbay      45  0x00000100  S    scsi   scsibus45
sd3270    sd       271  0x00000000  S    scsi   scsibus45
sd3271    sd       272  0x00000010  S    scsi   scsibus45
sd3272    sd       273  0x00000020  S    scsi   scsibus45
sd3273    sd       274  0x00000030  S    scsi   scsibus45
sd3274    sd       275  0x00000040  S    scsi   scsibus45
sd3275    sd       276  0x00000050  S    scsi   scsibus45
scsibus46 scsibus   46  0x06000060  SM   mscsi  fcbr5
pbay46    pbay      46  0x00000100  S    scsi   scsibus46
sd3276    sd       277  0x00000000  S    scsi   scsibus46
sd3277    sd       278  0x00000010  S    scsi   scsibus46
sd3278    sd       279  0x00000020  S    scsi   scsibus46
sd3279    sd       280  0x00000030  S    scsi   scsibus46
sd3280    sd       281  0x00000040  S    scsi   scsibus46
sd3281    sd       282  0x00000050  S    scsi   scsibus46
scsibus47 scsibus   47  0x07000060  SM   mscsi  fcbr5
pbay47    pbay      47  0x00000100  S    scsi   scsibus47
sd3282    sd       283  0x00000000  S    scsi   scsibus47
sd3283    sd       284  0x00000010  S    scsi   scsibus47
sd3284    sd       285  0x00000020  S    scsi   scsibus47
sd3285    sd       286  0x00000030  S    scsi   scsibus47
sd3286    sd       287  0x00000040  S    scsi   scsibus47
sd3287    sd       288  0x00000050  S    scsi   scsibus47
sdi0      sdi        0  0x00ffffff  SM   sdi    ff0
ff1       ff         1  0x00000001  SP   pci    quad0
fabric1   fabric     1  0x00fe5800  SM   fc     ff1
fcbr8     fcbr       8  0x00fe0000  SP   fcp    fabric1
scsibus64 scsibus   64  0x00000070  SM   mscsi  fcbr8
pbay64    pbay      64  0x00000100  S    scsi   scsibus64
sd3288    sd       385  0x00000000  S    scsi   scsibus64
sd3289    sd       386  0x00000010  S    scsi   scsibus64
sd3290    sd       387  0x00000020  S    scsi   scsibus64
sd3291    sd       388  0x00000030  S    scsi   scsibus64
sd3292    sd       389  0x00000040  S    scsi   scsibus64
sd3293    sd       390  0x00000050  S    scsi   scsibus64
scsibus65 scsibus   65  0x01000070  SM   mscsi  fcbr8
pbay65    pbay      65  0x00000100  S    scsi   scsibus65
sd3294    sd       391  0x00000000  S    scsi   scsibus65
sd3295    sd       392  0x00000010  S    scsi   scsibus65
sd3296    sd       393  0x00000020  S    scsi   scsibus65
sd3297    sd       394  0x00000030  S    scsi   scsibus65
sd3298    sd       395  0x00000040  S    scsi   scsibus65
sd3299    sd       396  0x00000050  S    scsi   scsibus65
scsibus66 scsibus   66  0x02000060  SM   mscsi  fcbr8
pbay66    pbay      66  0x00000100  S    scsi   scsibus66
sd3300    sd       397  0x00000000  S    scsi   scsibus66
sd3301    sd       398  0x00000010  S    scsi   scsibus66
sd3302    sd       399  0x00000020  S    scsi   scsibus66
sd3303    sd       400  0x00000030  S    scsi   scsibus66
sd3304    sd       401  0x00000040  S    scsi   scsibus66
sd3305    sd       402  0x00000050  S    scsi   scsibus66
scsibus67 scsibus   67  0x03000060  SM   mscsi  fcbr8
pbay67    pbay      67  0x00000100  S    scsi   scsibus67
sd3306    sd       403  0x00000000  S    scsi   scsibus67
sd3307    sd       404  0x00000010  S    scsi   scsibus67
sd3308    sd       405  0x00000020  S    scsi   scsibus67
sd3309    sd       406  0x00000030  S    scsi   scsibus67
sd3310    sd       407  0x00000040  S    scsi   scsibus67
sd3311    sd       408  0x00000050  S    scsi   scsibus67
```

Appendix C – Tunable Parameters

```
scsibus68   scsibus     68 0x04000070 SM     mscsi    fcbr8
pbay68      pbay        68 0x00000100 S      scsi     scsibus68
sd3312      sd         409 0x00000000 S      scsi     scsibus68
sd3313      sd         410 0x00000010 S      scsi     scsibus68
sd3314      sd         411 0x00000020 S      scsi     scsibus68
sd3315      sd         412 0x00000030 S      scsi     scsibus68
sd3316      sd         413 0x00000040 S      scsi     scsibus68
sd3317      sd         414 0x00000050 S      scsi     scsibus68
scsibus69   scsibus     69 0x05000070 SM     mscsi    fcbr8
pbay69      pbay        69 0x00000100 S      scsi     scsibus69
sd3318      sd         415 0x00000000 S      scsi     scsibus69
sd3319      sd         416 0x00000010 S      scsi     scsibus69
sd3320      sd         417 0x00000020 S      scsi     scsibus69
sd3321      sd         418 0x00000030 S      scsi     scsibus69
sd3322      sd         419 0x00000040 S      scsi     scsibus69
sd3323      sd         420 0x00000050 S      scsi     scsibus69
scsibus70   scsibus     70 0x06000060 SM     mscsi    fcbr8
pbay70      pbay        70 0x00000100 S      scsi     scsibus70
sd3324      sd         421 0x00000000 S      scsi     scsibus70
sd3325      sd         422 0x00000010 S      scsi     scsibus70
sd3326      sd         423 0x00000020 S      scsi     scsibus70
sd3327      sd         424 0x00000030 S      scsi     scsibus70
sd3328      sd         425 0x00000040 S      scsi     scsibus70
sd3329      sd         426 0x00000050 S      scsi     scsibus70
scsibus71   scsibus     71 0x07000060 SM     mscsi    fcbr8
pbay71      pbay        71 0x00000100 S      scsi     scsibus71
sd3330      sd         427 0x00000000 S      scsi     scsibus71
sd3331      sd         428 0x00000010 S      scsi     scsibus71
sd3332      sd         429 0x00000020 S      scsi     scsibus71
sd3333      sd         430 0x00000030 S      scsi     scsibus71
sd3334      sd         431 0x00000040 S      scsi     scsibus71
sd3335      sd         432 0x00000050 S      scsi     scsibus71
fcbr9       fcbr         9 0x00fe0800 SP     fcp      fabric1
scsibus72   scsibus     72 0x00000070 SM     mscsi    fcbr9
pbay72      pbay        72 0x00000100 S      scsi     scsibus72
sd3336      sd         433 0x00000000 S      scsi     scsibus72
sd3337      sd         434 0x00000010 S      scsi     scsibus72
sd3338      sd         435 0x00000020 S      scsi     scsibus72
sd3339      sd         436 0x00000030 S      scsi     scsibus72
sd3340      sd         437 0x00000040 S      scsi     scsibus72
sd3341      sd         438 0x00000050 S      scsi     scsibus72
scsibus73   scsibus     73 0x01000070 SM     mscsi    fcbr9
pbay73      pbay        73 0x00000100 S      scsi     scsibus73
sd3342      sd         439 0x00000000 S      scsi     scsibus73
sd3343      sd         440 0x00000010 S      scsi     scsibus73
sd3344      sd         441 0x00000020 S      scsi     scsibus73
sd3345      sd         442 0x00000030 S      scsi     scsibus73
sd3346      sd         443 0x00000040 S      scsi     scsibus73
sd3347      sd         444 0x00000050 S      scsi     scsibus73
scsibus74   scsibus     74 0x02000060 SM     mscsi    fcbr9
pbay74      pbay        74 0x00000100 S      scsi     scsibus74
sd3348      sd         445 0x00000000 S      scsi     scsibus74
sd3349      sd         446 0x00000010 S      scsi     scsibus74
sd3350      sd         447 0x00000020 S      scsi     scsibus74
sd3351      sd         448 0x00000030 S      scsi     scsibus74
sd3352      sd         449 0x00000040 S      scsi     scsibus74
sd3353      sd         450 0x00000050 S      scsi     scsibus74
scsibus75   scsibus     75 0x03000060 SM     mscsi    fcbr9
pbay75      pbay        75 0x00000100 S      scsi     scsibus75
sd3354      sd         451 0x00000000 S      scsi     scsibus75
sd3355      sd         452 0x00000010 S      scsi     scsibus75
sd3356      sd         453 0x00000020 S      scsi     scsibus75
sd3357      sd         454 0x00000030 S      scsi     scsibus75
sd3358      sd         455 0x00000040 S      scsi     scsibus75
sd3359      sd         456 0x00000050 S      scsi     scsibus75
scsibus76   scsibus     76 0x04000070 SM     mscsi    fcbr9
pbay76      pbay        76 0x00000100 S      scsi     scsibus76
sd3360      sd         457 0x00000000 S      scsi     scsibus76
sd3361      sd         458 0x00000010 S      scsi     scsibus76
sd3362      sd         459 0x00000020 S      scsi     scsibus76
sd3363      sd         460 0x00000030 S      scsi     scsibus76
sd3364      sd         461 0x00000040 S      scsi     scsibus76
sd3365      sd         462 0x00000050 S      scsi     scsibus76
scsibus77   scsibus     77 0x05000070 SM     mscsi    fcbr9
pbay77      pbay        77 0x00000100 S      scsi     scsibus77
sd3366      sd         463 0x00000000 S      scsi     scsibus77
sd3367      sd         464 0x00000010 S      scsi     scsibus77
sd3368      sd         465 0x00000020 S      scsi     scsibus77
sd3369      sd         466 0x00000030 S      scsi     scsibus77
sd3370      sd         467 0x00000040 S      scsi     scsibus77
sd3371      sd         468 0x00000050 S      scsi     scsibus77
scsibus78   scsibus     78 0x06000060 SM     mscsi    fcbr9
pbay78      pbay        78 0x00000100 S      scsi     scsibus78
sd3372      sd         469 0x00000000 S      scsi     scsibus78
sd3373      sd         470 0x00000010 S      scsi     scsibus78
sd3374      sd         471 0x00000020 S      scsi     scsibus78
sd3375      sd         472 0x00000030 S      scsi     scsibus78
sd3376      sd         473 0x00000040 S      scsi     scsibus78
sd3377      sd         474 0x00000050 S      scsi     scsibus78
scsibus79   scsibus     79 0x07000060 SM     mscsi    fcbr9
pbay79      pbay        79 0x00000100 S      scsi     scsibus79
sd3378      sd         475 0x00000000 S      scsi     scsibus79
sd3379      sd         476 0x00000010 S      scsi     scsibus79
sd3380      sd         477 0x00000020 S      scsi     scsibus79
sd3381      sd         478 0x00000030 S      scsi     scsibus79
sd3382      sd         479 0x00000040 S      scsi     scsibus79
sd3383      sd         480 0x00000050 S      scsi     scsibus79
fcbr10      fcbr        10 0x00fe1000 SP     fcp      fabric1
scsibus80   scsibus     80 0x00000070 SM     mscsi    fcbr10
pbay80      pbay        80 0x00000100 S      scsi     scsibus80
sd3384      sd         481 0x00000000 S      scsi     scsibus80
sd3385      sd         482 0x00000010 S      scsi     scsibus80
sd3386      sd         483 0x00000020 S      scsi     scsibus80
sd3387      sd         484 0x00000030 S      scsi     scsibus80
sd3388      sd         485 0x00000040 S      scsi     scsibus80
sd3389      sd         486 0x00000050 S      scsi     scsibus80
scsibus81   scsibus     81 0x01000070 SM     mscsi    fcbr10
pbay81      pbay        81 0x00000100 S      scsi     scsibus81
sd3390      sd         487 0x00000000 S      scsi     scsibus81
sd3391      sd         488 0x00000010 S      scsi     scsibus81
sd3392      sd         489 0x00000020 S      scsi     scsibus81
sd3393      sd         490 0x00000030 S      scsi     scsibus81
sd3394      sd         491 0x00000040 S      scsi     scsibus81
sd3395      sd         492 0x00000050 S      scsi     scsibus81
scsibus82   scsibus     82 0x02000060 SM     mscsi    fcbr10
pbay82      pbay        82 0x00000100 S      scsi     scsibus82
sd3396      sd         493 0x00000000 S      scsi     scsibus82
sd3397      sd         494 0x00000010 S      scsi     scsibus82
sd3398      sd         495 0x00000020 S      scsi     scsibus82
sd3399      sd         496 0x00000030 S      scsi     scsibus82
sd3400      sd         497 0x00000040 S      scsi     scsibus82
sd3401      sd         498 0x00000050 S      scsi     scsibus82
scsibus83   scsibus     83 0x03000060 SM     mscsi    fcbr10
pbay83      pbay        83 0x00000100 S      scsi     scsibus83
sd3402      sd         499 0x00000000 S      scsi     scsibus83
sd3403      sd         500 0x00000010 S      scsi     scsibus83
sd3404      sd         501 0x00000020 S      scsi     scsibus83
sd3405      sd         502 0x00000030 S      scsi     scsibus83
sd3406      sd         503 0x00000040 S      scsi     scsibus83
sd3407      sd         504 0x00000050 S      scsi     scsibus83
scsibus84   scsibus     84 0x04000070 SM     mscsi    fcbr10
pbay84      pbay        84 0x00000100 S      scsi     scsibus84
sd3408      sd         505 0x00000000 S      scsi     scsibus84
sd3409      sd         506 0x00000010 S      scsi     scsibus84
sd3410      sd         507 0x00000020 S      scsi     scsibus84
sd3411      sd         508 0x00000030 S      scsi     scsibus84
sd3412      sd         509 0x00000040 S      scsi     scsibus84
sd3413      sd         510 0x00000050 S      scsi     scsibus84
scsibus85   scsibus     85 0x05000070 SM     mscsi    fcbr10
pbay85      pbay        85 0x00000100 S      scsi     scsibus85
sd3414      sd         511 0x00000000 S      scsi     scsibus85
sd3415      sd         512 0x00000010 S      scsi     scsibus85
sd3416      sd         513 0x00000020 S      scsi     scsibus85
sd3417      sd         514 0x00000030 S      scsi     scsibus85
sd3418      sd         515 0x00000040 S      scsi     scsibus85
sd3419      sd         516 0x00000050 S      scsi     scsibus85
scsibus86   scsibus     86 0x06000060 SM     mscsi    fcbr10
pbay86      pbay        86 0x00000100 S      scsi     scsibus86
sd3420      sd         517 0x00000000 S      scsi     scsibus86
sd3421      sd         518 0x00000010 S      scsi     scsibus86
sd3422      sd         519 0x00000020 S      scsi     scsibus86
sd3423      sd         520 0x00000030 S      scsi     scsibus86
sd3424      sd         521 0x00000040 S      scsi     scsibus86
sd3425      sd         522 0x00000050 S      scsi     scsibus86
scsibus87   scsibus     87 0x07000060 SM     mscsi    fcbr10
pbay87      pbay        87 0x00000100 S      scsi     scsibus87
sd3426      sd         523 0x00000000 S      scsi     scsibus87
sd3427      sd         524 0x00000010 S      scsi     scsibus87
sd3428      sd         525 0x00000020 S      scsi     scsibus87
sd3429      sd         526 0x00000030 S      scsi     scsibus87
sd3430      sd         527 0x00000040 S      scsi     scsibus87
sd3431      sd         528 0x00000050 S      scsi     scsibus87
fcbr11      fcbr        11 0x00fe1800 SP     fcp      fabric1
scsibus88   scsibus     88 0x00000070 SM     mscsi    fcbr11
pbay88      pbay        88 0x00000100 S      scsi     scsibus88
sd3432      sd         529 0x00000000 S      scsi     scsibus88
sd3433      sd         530 0x00000010 S      scsi     scsibus88
sd3434      sd         531 0x00000020 S      scsi     scsibus88
sd3435      sd         532 0x00000030 S      scsi     scsibus88
sd3436      sd         533 0x00000040 S      scsi     scsibus88
sd3437      sd         534 0x00000050 S      scsi     scsibus88
scsibus89   scsibus     89 0x01000070 SM     mscsi    fcbr11
pbay89      pbay        89 0x00000100 S      scsi     scsibus89
sd3438      sd         535 0x00000000 S      scsi     scsibus89
sd3439      sd         536 0x00000010 S      scsi     scsibus89
sd3440      sd         537 0x00000020 S      scsi     scsibus89
sd3441      sd         538 0x00000030 S      scsi     scsibus89
sd3442      sd         539 0x00000040 S      scsi     scsibus89
```

Appendix C – Tunable Parameters

```
sd3443     sd       540  0x00000050 S    scsi    scsibus89
scsibus90  scsibus   90  0x02000060 SM   mscsi   fcbr11
pbay90     pbay      90  0x00000100 S    scsi    scsibus90
sd3444     sd       541  0x00000000 S    scsi    scsibus90
sd3445     sd       542  0x00000010 S    scsi    scsibus90
sd3446     sd       543  0x00000020 S    scsi    scsibus90
sd3447     sd       544  0x00000030 S    scsi    scsibus90
sd3448     sd       545  0x00000040 S    scsi    scsibus90
sd3449     sd       546  0x00000050 S    scsi    scsibus90
scsibus91  scsibus   91  0x03000060 SM   mscsi   fcbr11
pbay91     pbay      91  0x00000100 S    scsi    scsibus91
sd3450     sd       547  0x00000000 S    scsi    scsibus91
sd3451     sd       548  0x00000010 S    scsi    scsibus91
sd3452     sd       549  0x00000020 S    scsi    scsibus91
sd3453     sd       550  0x00000030 S    scsi    scsibus91
sd3454     sd       551  0x00000040 S    scsi    scsibus91
sd3455     sd       552  0x00000050 S    scsi    scsibus91
scsibus92  scsibus   92  0x04000070 SM   mscsi   fcbr11
pbay92     pbay      92  0x00000100 S    scsi    scsibus92
sd3456     sd       553  0x00000000 S    scsi    scsibus92
sd3457     sd       554  0x00000010 S    scsi    scsibus92
sd3458     sd       555  0x00000020 S    scsi    scsibus92
sd3459     sd       556  0x00000030 S    scsi    scsibus92
sd3460     sd       557  0x00000040 S    scsi    scsibus92
sd3461     sd       558  0x00000050 S    scsi    scsibus92
scsibus93  scsibus   93  0x05000070 SM   mscsi   fcbr11
pbay93     pbay      93  0x00000100 S    scsi    scsibus93
sd3462     sd       559  0x00000000 S    scsi    scsibus93
sd3463     sd       560  0x00000010 S    scsi    scsibus93
sd3464     sd       561  0x00000020 S    scsi    scsibus93
sd3465     sd       562  0x00000030 S    scsi    scsibus93
sd3466     sd       563  0x00000040 S    scsi    scsibus93
sd3467     sd       564  0x00000050 S    scsi    scsibus93
scsibus94  scsibus   94  0x06000060 SM   mscsi   fcbr11
pbay94     pbay      94  0x00000100 S    scsi    scsibus94
sd3468     sd       565  0x00000000 S    scsi    scsibus94
sd3469     sd       566  0x00000010 S    scsi    scsibus94
sd3470     sd       567  0x00000020 S    scsi    scsibus94
sd3471     sd       568  0x00000030 S    scsi    scsibus94
sd3472     sd       569  0x00000040 S    scsi    scsibus94
sd3473     sd       570  0x00000050 S    scsi    scsibus94
scsibus95  scsibus   95  0x07000060 SM   mscsi   fcbr11
pbay95     pbay      95  0x00000100 S    scsi    scsibus95
sd3474     sd       571  0x00000000 S    scsi    scsibus95
sd3475     sd       572  0x00000010 S    scsi    scsibus95
sd3476     sd       573  0x00000020 S    scsi    scsibus95
sd3477     sd       574  0x00000030 S    scsi    scsibus95
sd3478     sd       575  0x00000040 S    scsi    scsibus95
sd3479     sd       576  0x00000050 S    scsi    scsibus95
fcbr12     fcbr      12  0x00fe2000 SP   fcp     fabric1
scsibus96  scsibus   96  0x00000070 SM   mscsi   fcbr12
pbay96     pbay      96  0x00000100 S    scsi    scsibus96
sd3480     sd       577  0x00000000 S    scsi    scsibus96
sd3481     sd       578  0x00000010 S    scsi    scsibus96
sd3482     sd       579  0x00000020 S    scsi    scsibus96
sd3483     sd       580  0x00000030 S    scsi    scsibus96
sd3484     sd       581  0x00000040 S    scsi    scsibus96
sd3485     sd       582  0x00000050 S    scsi    scsibus96
scsibus97  scsibus   97  0x01000070 SM   mscsi   fcbr12
pbay97     pbay      97  0x00000100 S    scsi    scsibus97
sd3486     sd       583  0x00000000 S    scsi    scsibus97
sd3487     sd       584  0x00000010 S    scsi    scsibus97
sd3488     sd       585  0x00000020 S    scsi    scsibus97
sd3489     sd       586  0x00000030 S    scsi    scsibus97
sd3490     sd       587  0x00000040 S    scsi    scsibus97
sd3491     sd       588  0x00000050 S    scsi    scsibus97
scsibus98  scsibus   98  0x02000060 SM   mscsi   fcbr12
pbay98     pbay      98  0x00000100 S    scsi    scsibus98
sd3492     sd       589  0x00000000 S    scsi    scsibus98
sd3493     sd       590  0x00000010 S    scsi    scsibus98
sd3494     sd       591  0x00000020 S    scsi    scsibus98
sd3495     sd       592  0x00000030 S    scsi    scsibus98
sd3496     sd       593  0x00000040 S    scsi    scsibus98
sd3497     sd       594  0x00000050 S    scsi    scsibus98
scsibus99  scsibus   99  0x03000060 SM   mscsi   fcbr12
pbay99     pbay      99  0x00000100 S    scsi    scsibus99
sd3498     sd       595  0x00000000 S    scsi    scsibus99
sd3499     sd       596  0x00000010 S    scsi    scsibus99
sd3500     sd       597  0x00000020 S    scsi    scsibus99
sd3501     sd       598  0x00000030 S    scsi    scsibus99
sd3502     sd       599  0x00000040 S    scsi    scsibus99
sd3503     sd       600  0x00000050 S    scsi    scsibus99
scsibus100 scsibus  100  0x04000070 SM   mscsi   fcbr12
pbay100    pbay     100  0x00000100 S    scsi    scsibus100
sd3504     sd       601  0x00000000 S    scsi    scsibus100
sd3505     sd       602  0x00000010 S    scsi    scsibus100
sd3506     sd       603  0x00000020 S    scsi    scsibus100
sd3507     sd       604  0x00000030 S    scsi    scsibus100
sd3508     sd       605  0x00000040 S    scsi    scsibus100
sd3509     sd       606  0x00000050 S    scsi    scsibus100
scsibus101 scsibus  101  0x05000070 SM   mscsi   fcbr12
pbay101    pbay     101  0x00000100 S    scsi    scsibus101
sd3510     sd       607  0x00000000 S    scsi    scsibus101
sd3511     sd       608  0x00000010 S    scsi    scsibus101
sd3512     sd       609  0x00000020 S    scsi    scsibus101
sd3513     sd       610  0x00000030 S    scsi    scsibus101
sd3514     sd       611  0x00000040 S    scsi    scsibus101
sd3515     sd       612  0x00000050 S    scsi    scsibus101
scsibus102 scsibus  102  0x06000060 SM   mscsi   fcbr12
pbay102    pbay     102  0x00000100 S    scsi    scsibus102
sd3516     sd       613  0x00000000 S    scsi    scsibus102
sd3517     sd       614  0x00000010 S    scsi    scsibus102
sd3518     sd       615  0x00000020 S    scsi    scsibus102
sd3519     sd       616  0x00000030 S    scsi    scsibus102
sd3520     sd       617  0x00000040 S    scsi    scsibus102
sd3521     sd       618  0x00000050 S    scsi    scsibus102
scsibus103 scsibus  103  0x07000060 SM   mscsi   fcbr12
pbay103    pbay     103  0x00000100 S    scsi    scsibus103
sd3522     sd       619  0x00000000 S    scsi    scsibus103
sd3523     sd       620  0x00000010 S    scsi    scsibus103
sd3524     sd       621  0x00000020 S    scsi    scsibus103
sd3525     sd       622  0x00000030 S    scsi    scsibus103
sd3526     sd       623  0x00000040 S    scsi    scsibus103
sd3527     sd       624  0x00000050 S    scsi    scsibus103
fcbr13     fcbr      13  0x00fe2800 SP   fcp     fabric1
scsibus104 scsibus  104  0x00000070 SM   mscsi   fcbr13
pbay104    pbay     104  0x00000100 S    scsi    scsibus104
sd3528     sd       625  0x00000000 S    scsi    scsibus104
sd3529     sd       626  0x00000010 S    scsi    scsibus104
sd3530     sd       627  0x00000020 S    scsi    scsibus104
sd3531     sd       628  0x00000030 S    scsi    scsibus104
sd3532     sd       629  0x00000040 S    scsi    scsibus104
sd3533     sd       630  0x00000050 S    scsi    scsibus104
scsibus105 scsibus  105  0x01000070 SM   mscsi   fcbr13
pbay105    pbay     105  0x00000100 S    scsi    scsibus105
sd3534     sd       631  0x00000000 S    scsi    scsibus105
sd3535     sd       632  0x00000010 S    scsi    scsibus105
sd3536     sd       633  0x00000020 S    scsi    scsibus105
sd3537     sd       634  0x00000030 S    scsi    scsibus105
sd3538     sd       635  0x00000040 S    scsi    scsibus105
sd3539     sd       636  0x00000050 S    scsi    scsibus105
scsibus106 scsibus  106  0x02000060 SM   mscsi   fcbr13
pbay106    pbay     106  0x00000100 S    scsi    scsibus106
sd3540     sd       637  0x00000000 S    scsi    scsibus106
sd3541     sd       638  0x00000010 S    scsi    scsibus106
sd3542     sd       639  0x00000020 S    scsi    scsibus106
sd3543     sd       640  0x00000030 S    scsi    scsibus106
sd3544     sd       641  0x00000040 S    scsi    scsibus106
sd3545     sd       642  0x00000050 S    scsi    scsibus106
scsibus107 scsibus  107  0x03000060 SM   mscsi   fcbr13
pbay107    pbay     107  0x00000100 S    scsi    scsibus107
sd3546     sd       643  0x00000000 S    scsi    scsibus107
sd3547     sd       644  0x00000010 S    scsi    scsibus107
sd3548     sd       645  0x00000020 S    scsi    scsibus107
sd3549     sd       646  0x00000030 S    scsi    scsibus107
sd3550     sd       647  0x00000040 S    scsi    scsibus107
sd3551     sd       648  0x00000050 S    scsi    scsibus107
scsibus108 scsibus  108  0x04000070 SM   mscsi   fcbr13
pbay108    pbay     108  0x00000100 S    scsi    scsibus108
sd3552     sd       649  0x00000000 S    scsi    scsibus108
sd3553     sd       650  0x00000010 S    scsi    scsibus108
sd3554     sd       651  0x00000020 S    scsi    scsibus108
sd3555     sd       652  0x00000030 S    scsi    scsibus108
sd3556     sd       653  0x00000040 S    scsi    scsibus108
sd3557     sd       654  0x00000050 S    scsi    scsibus108
scsibus109 scsibus  109  0x05000070 SM   mscsi   fcbr13
pbay109    pbay     109  0x00000100 S    scsi    scsibus109
sd3558     sd       655  0x00000000 S    scsi    scsibus109
sd3559     sd       656  0x00000010 S    scsi    scsibus109
sd3560     sd       657  0x00000020 S    scsi    scsibus109
sd3561     sd       658  0x00000030 S    scsi    scsibus109
sd3562     sd       659  0x00000040 S    scsi    scsibus109
sd3563     sd       660  0x00000050 S    scsi    scsibus109
scsibus110 scsibus  110  0x06000060 SM   mscsi   fcbr13
pbay110    pbay     110  0x00000100 S    scsi    scsibus110
sd3564     sd       661  0x00000000 S    scsi    scsibus110
sd3565     sd       662  0x00000010 S    scsi    scsibus110
sd3566     sd       663  0x00000020 S    scsi    scsibus110
sd3567     sd       664  0x00000030 S    scsi    scsibus110
sd3568     sd       665  0x00000040 S    scsi    scsibus110
sd3569     sd       666  0x00000050 S    scsi    scsibus110
scsibus111 scsibus  111  0x07000060 SM   mscsi   fcbr13
pbay111    pbay     111  0x00000100 S    scsi    scsibus111
sd3570     sd       667  0x00000000 S    scsi    scsibus111
sd3571     sd       668  0x00000010 S    scsi    scsibus111
sd3572     sd       669  0x00000020 S    scsi    scsibus111
sd3573     sd       670  0x00000030 S    scsi    scsibus111
sd3574     sd       671  0x00000040 S    scsi    scsibus111
```

Appendix C – Tunable Parameters

```
sd3575    sd      672  0x00000050 S    scsi   scsibus111
fcbr15    fcbr     15  0x00fe3800 SP   fcp    fabric1
scsibus120 scsibus  120 0x00000070 SM   mscsi  fcbr15
pbay120   pbay    120  0x00000100 S    scsi   scsibus120
sd3720    sd      721  0x00000000 S    scsi   scsibus120
sd3721    sd      722  0x00000010 S    scsi   scsibus120
sd3722    sd     1609  0x00000020 S    scsi   scsibus120
sd3723    sd      724  0x00000030 S    scsi   scsibus120
sd3724    sd      725  0x00000040 S    scsi   scsibus120
sd3725    sd      726  0x00000050 S    scsi   scsibus120
scsibus121 scsibus  121 0x01000070 SM   mscsi  fcbr15
pbay121   pbay    121  0x00000100 S    scsi   scsibus121
sd3726    sd      727  0x00000000 S    scsi   scsibus121
sd3727    sd      728  0x00000010 S    scsi   scsibus121
sd3728    sd      729  0x00000020 S    scsi   scsibus121
sd3729    sd      730  0x00000030 S    scsi   scsibus121
sd3730    sd      731  0x00000040 S    scsi   scsibus121
sd3731    sd      732  0x00000050 S    scsi   scsibus121
scsibus122 scsibus  122 0x02000060 SM   mscsi  fcbr15
pbay122   pbay    122  0x00000100 S    scsi   scsibus122
sd3732    sd      733  0x00000000 S    scsi   scsibus122
sd3733    sd      734  0x00000010 S    scsi   scsibus122
sd3734    sd      735  0x00000020 S    scsi   scsibus122
sd3735    sd      736  0x00000030 S    scsi   scsibus122
sd3736    sd      737  0x00000040 S    scsi   scsibus122
sd3737    sd      738  0x00000050 S    scsi   scsibus122
scsibus123 scsibus  123 0x03000060 SM   mscsi  fcbr15
pbay123   pbay    123  0x00000100 S    scsi   scsibus123
sd4720    sd      739  0x00000000 S    scsi   scsibus123
sd4721    sd      740  0x00000010 S    scsi   scsibus123
sd4722    sd      741  0x00000020 S    scsi   scsibus123
sd4723    sd      742  0x00000030 S    scsi   scsibus123
sd4724    sd      743  0x00000040 S    scsi   scsibus123
sd4725    sd      744  0x00000050 S    scsi   scsibus123
scsibus124 scsibus  124 0x04000070 SM   mscsi  fcbr15
pbay124   pbay    124  0x00000100 S    scsi   scsibus124
sd6720    sd      745  0x00000000 S    scsi   scsibus124
sd6721    sd      746  0x00000010 S    scsi   scsibus124
sd6722    sd      747  0x00000020 S    scsi   scsibus124
sd6723    sd      748  0x00000030 S    scsi   scsibus124
sd6724    sd      749  0x00000040 S    scsi   scsibus124
sd6725    sd      750  0x00000050 S    scsi   scsibus124
scsibus125 scsibus  125 0x05000070 SM   mscsi  fcbr15
pbay125   pbay    125  0x00000100 S    scsi   scsibus125
sd6726    sd      751  0x00000000 S    scsi   scsibus125
sd6727    sd      752  0x00000010 S    scsi   scsibus125
sd6728    sd      753  0x00000020 S    scsi   scsibus125
sd6729    sd      754  0x00000030 S    scsi   scsibus125
sd6730    sd      755  0x00000040 S    scsi   scsibus125
sd6731    sd      756  0x00000050 S    scsi   scsibus125
scsibus126 scsibus  126 0x06000060 SM   mscsi  fcbr15
pbay126   pbay    126  0x00000100 S    scsi   scsibus126
sd4726    sd      757  0x00000000 S    scsi   scsibus126
sd4727    sd      758  0x00000010 S    scsi   scsibus126
sd4728    sd      759  0x00000020 S    scsi   scsibus126
sd4729    sd      760  0x00000030 S    scsi   scsibus126
sd4730    sd      761  0x00000040 S    scsi   scsibus126
sd4731    sd      762  0x00000050 S    scsi   scsibus126
scsibus127 scsibus  127 0x07000060 SM   mscsi  fcbr15
pbay127   pbay    127  0x00000100 S    scsi   scsibus127
sd4732    sd      763  0x00000000 S    scsi   scsibus127
sd4733    sd      764  0x00000010 S    scsi   scsibus127
sd4734    sd      765  0x00000020 S    scsi   scsibus127
sd4735    sd      766  0x00000030 S    scsi   scsibus127
sd4736    sd      767  0x00000040 S    scsi   scsibus127
sd4737    sd      768  0x00000050 S    scsi   scsibus127
sdi0      sdi       0  0x00ffffff SM   sdi    ff1
qlc0      qlc       0  0x00000002 L    pci    quad0
scsibus248 scsibus  248 0x00000070 L    mscsi  qlc0
pbay248   pbay    248  0x00000100 L    scsi   scsibus248
sd9000    sd     1489  0x00000020 L    scsi   scsibus248
sd2000    sd     1596  0x00000080 L    scsi   scsibus248
sd2001    sd     1504  0x00000090 L    scsi   scsibus248
sd2002    sd     1505  0x000000a0 L    scsi   scsibus248
qlc1      qlc       1  0x00000004 L    pci    quad0
scsibus249 scsibus  249 0x00000070 L    mscsi  qlc1
pbay249   pbay    249  0x00000100 L    scsi   scsibus249
sd2003    sd     1597  0x00000000 L    scsi   scsibus249
sd2004    sd     1598  0x00000010 L    scsi   scsibus249
sd2005    sd     1498  0x00000020 L    scsi   scsibus249
pci0      pci       0  0x00000003 L    pci    quad0
pe0       pe        0  0x00000003 L    pci    pci0
pe1       pe        1  0x00000003 L    pci    pci0
pe2       pe        2  0x00000003 L    pci    pci0
pe3       pe        3  0x00000003 L    pci    pci0
quad1     quad      1  0x00000001 L    sci    sci0
ff4       ff        4  0x00000000 SP   pci    quad1
fabric0   fabric    0  0x00fe5000 SM   fc     ff4
sdi0      sdi       0  0x00ffffff SM   sdi    ff4
ff5       ff        5  0x00000001 SP   pci    quad1
fabric1   fabric    1  0x00fe5000 SM   fc     ff5
sdi0      sdi       0  0x00ffffff SM   sdi    ff5
qlc2      qlc       2  0x00000002 L    pci    quad1
scsibus250 scsibus  250 0x00000070 L    mscsi  qlc2
pbay250   pbay    250  0x00000100 L    scsi   scsibus250
sd2006    sd     1599  0x00000000 L    scsi   scsibus250
sd2007    sd     1600  0x00000010 L    scsi   scsibus250
sd2008    sd     1601  0x00000020 L    scsi   scsibus250
qlc3      qlc       3  0x00000004 L    pci    quad1
scsibus251 scsibus  251 0x00000070 L    mscsi  qlc3
pbay251   pbay    251  0x00000100 L    scsi   scsibus251
sd2009    sd     1509  0x00000000 L    scsi   scsibus251
sd2010    sd     1510  0x00000010 L    scsi   scsibus251
sd2011    sd     1511  0x00000020 L    scsi   scsibus251
pe4       pe        4  0x00000003 L    pci    quad1
quad2     quad      2  0x00000002 L    sci    sci0
ff10      ff       10  0x00000005 SP   pci    quad2
fabric6   fabric    7  0x00fe5800 SM   fc     ff10
fcbr117   fcbr     54  0x00fe0000 SP   fcp    fabric6
scsibus136 scsibus  136 0x03000060 SM   mscsi  fcbr117
pbay136   pbay    136  0x00000100 S    scsi   scsibus136
sd6048    sd      817  0x00000000 S    scsi   scsibus136
sd6049    sd      818  0x00000010 S    scsi   scsibus136
sd6050    sd      819  0x00000020 S    scsi   scsibus136
sd6051    sd      820  0x00000030 S    scsi   scsibus136
sd6052    sd      821  0x00000040 S    scsi   scsibus136
sd6053    sd      822  0x00000050 S    scsi   scsibus136
scsibus137 scsibus  137 0x02000060 SM   mscsi  fcbr117
pbay137   pbay    137  0x00000100 S    scsi   scsibus137
sd6054    sd      823  0x00000000 S    scsi   scsibus137
sd6055    sd      824  0x00000010 S    scsi   scsibus137
sd6056    sd      825  0x00000020 S    scsi   scsibus137
sd6057    sd      826  0x00000030 S    scsi   scsibus137
sd6058    sd      827  0x00000040 S    scsi   scsibus137
sd6059    sd      828  0x00000050 S    scsi   scsibus137
scsibus138 scsibus  138 0x01000070 SM   mscsi  fcbr117
pbay138   pbay    138  0x00000100 S    scsi   scsibus138
sd6060    sd      829  0x00000000 S    scsi   scsibus138
sd6061    sd      830  0x00000010 S    scsi   scsibus138
sd6062    sd      831  0x00000020 S    scsi   scsibus138
sd6063    sd      832  0x00000030 S    scsi   scsibus138
sd6064    sd      833  0x00000040 S    scsi   scsibus138
sd6065    sd      834  0x00000050 S    scsi   scsibus138
scsibus139 scsibus  139 0x00000070 SM   mscsi  fcbr117
pbay139   pbay    139  0x00000100 S    scsi   scsibus139
sd6066    sd      835  0x00000000 S    scsi   scsibus139
sd6067    sd      836  0x00000010 S    scsi   scsibus139
sd6068    sd      837  0x00000020 S    scsi   scsibus139
sd6069    sd      838  0x00000030 S    scsi   scsibus139
sd6070    sd      839  0x00000040 S    scsi   scsibus139
sd6071    sd      840  0x00000050 S    scsi   scsibus139
scsibus140 scsibus  140 0x07000060 SM   mscsi  fcbr117
pbay140   pbay    140  0x00000100 S    scsi   scsibus140
sd6072    sd      841  0x00000000 S    scsi   scsibus140
sd6073    sd      842  0x00000010 S    scsi   scsibus140
sd6074    sd      843  0x00000020 S    scsi   scsibus140
sd6075    sd      844  0x00000030 S    scsi   scsibus140
sd6076    sd      845  0x00000040 S    scsi   scsibus140
sd6077    sd      846  0x00000050 S    scsi   scsibus140
scsibus141 scsibus  141 0x06000060 SM   mscsi  fcbr117
pbay141   pbay    141  0x00000100 S    scsi   scsibus141
sd6078    sd      847  0x00000000 S    scsi   scsibus141
sd6079    sd      848  0x00000010 S    scsi   scsibus141
sd6080    sd      849  0x00000020 S    scsi   scsibus141
sd6081    sd      850  0x00000030 S    scsi   scsibus141
sd6082    sd      851  0x00000040 S    scsi   scsibus141
sd6083    sd      852  0x00000050 S    scsi   scsibus141
scsibus142 scsibus  142 0x05000070 SM   mscsi  fcbr117
pbay142   pbay    142  0x00000100 S    scsi   scsibus142
sd6084    sd      853  0x00000000 S    scsi   scsibus142
sd6085    sd      854  0x00000010 S    scsi   scsibus142
sd6086    sd      855  0x00000020 S    scsi   scsibus142
sd6087    sd      856  0x00000030 S    scsi   scsibus142
sd6088    sd      857  0x00000040 S    scsi   scsibus142
sd6089    sd      858  0x00000050 S    scsi   scsibus142
scsibus143 scsibus  143 0x04000070 SM   mscsi  fcbr117
pbay143   pbay    143  0x00000100 S    scsi   scsibus143
sd6090    sd      859  0x00000000 S    scsi   scsibus143
sd6091    sd      860  0x00000010 S    scsi   scsibus143
sd6092    sd      861  0x00000020 S    scsi   scsibus143
sd6093    sd      862  0x00000030 S    scsi   scsibus143
sd6094    sd      863  0x00000040 S    scsi   scsibus143
sd6095    sd      864  0x00000050 S    scsi   scsibus143
fcbr116   fcbr     55  0x00fe0800 SP   fcp    fabric6
scsibus128 scsibus  128 0x06000060 SM   mscsi  fcbr116
pbay128   pbay    128  0x00000100 S    scsi   scsibus128
sd6000    sd      769  0x00000000 S    scsi   scsibus128
sd6001    sd      770  0x00000010 S    scsi   scsibus128
```

Appendix C – Tunable Parameters

```
sd6002    sd       771 0x00000020 S    scsi   scsibus128
sd6003    sd       772 0x00000030 S    scsi   scsibus128
sd6004    sd       773 0x00000040 S    scsi   scsibus128
sd6005    sd       774 0x00000050 S    scsi   scsibus128
scsibus129 scsibus  129 0x07000060 SM   mscsi  fcbr116
pbay129   pbay     129 0x00000100 S    scsi   scsibus129
sd6006    sd       775 0x00000000 S    scsi   scsibus129
sd6007    sd       776 0x00000010 S    scsi   scsibus129
sd6008    sd       777 0x00000020 S    scsi   scsibus129
sd6009    sd       778 0x00000030 S    scsi   scsibus129
sd6010    sd       779 0x00000040 S    scsi   scsibus129
sd6011    sd       780 0x00000050 S    scsi   scsibus129
scsibus130 scsibus  130 0x04000070 SM   mscsi  fcbr116
pbay130   pbay     130 0x00000100 S    scsi   scsibus130
sd6012    sd       781 0x00000000 S    scsi   scsibus130
sd6013    sd       782 0x00000010 S    scsi   scsibus130
sd6014    sd       783 0x00000020 S    scsi   scsibus130
sd6015    sd       784 0x00000030 S    scsi   scsibus130
sd6016    sd       785 0x00000040 S    scsi   scsibus130
sd6017    sd       786 0x00000050 S    scsi   scsibus130
scsibus131 scsibus  131 0x05000070 SM   mscsi  fcbr116
pbay131   pbay     131 0x00000100 S    scsi   scsibus131
sd6018    sd       787 0x00000000 S    scsi   scsibus131
sd6019    sd       788 0x00000010 S    scsi   scsibus131
sd6020    sd       789 0x00000020 S    scsi   scsibus131
sd6021    sd       790 0x00000030 S    scsi   scsibus131
sd6022    sd       791 0x00000040 S    scsi   scsibus131
sd6023    sd       792 0x00000050 S    scsi   scsibus131
scsibus132 scsibus  132 0x02000060 SM   mscsi  fcbr116
pbay132   pbay     132 0x00000100 S    scsi   scsibus132
sd6024    sd       793 0x00000000 S    scsi   scsibus132
sd6025    sd       794 0x00000010 S    scsi   scsibus132
sd6026    sd       795 0x00000020 S    scsi   scsibus132
sd6027    sd       796 0x00000030 S    scsi   scsibus132
sd6028    sd       797 0x00000040 S    scsi   scsibus132
sd6029    sd       798 0x00000050 S    scsi   scsibus132
scsibus133 scsibus  133 0x03000060 SM   mscsi  fcbr116
pbay133   pbay     133 0x00000100 S    scsi   scsibus133
sd6030    sd       799 0x00000000 S    scsi   scsibus133
sd6031    sd       800 0x00000010 S    scsi   scsibus133
sd6032    sd       801 0x00000020 S    scsi   scsibus133
sd6033    sd       802 0x00000030 S    scsi   scsibus133
sd6034    sd       803 0x00000040 S    scsi   scsibus133
sd6035    sd       804 0x00000050 S    scsi   scsibus133
scsibus134 scsibus  134 0x00000070 SM   mscsi  fcbr116
pbay134   pbay     134 0x00000100 S    scsi   scsibus134
sd6036    sd       805 0x00000000 S    scsi   scsibus134
sd6037    sd       806 0x00000010 S    scsi   scsibus134
sd6038    sd       807 0x00000020 S    scsi   scsibus134
sd6039    sd       808 0x00000030 S    scsi   scsibus134
sd6040    sd       809 0x00000040 S    scsi   scsibus134
sd6041    sd       810 0x00000050 S    scsi   scsibus134
scsibus135 scsibus  135 0x01000070 SM   mscsi  fcbr116
pbay135   pbay     135 0x00000100 S    scsi   scsibus135
sd6042    sd       811 0x00000000 S    scsi   scsibus135
sd6043    sd       812 0x00000010 S    scsi   scsibus135
sd6044    sd       813 0x00000020 S    scsi   scsibus135
sd6045    sd       814 0x00000030 S    scsi   scsibus135
sd6046    sd       815 0x00000040 S    scsi   scsibus135
sd6047    sd       816 0x00000050 S    scsi   scsibus135
fcbr118   fcbr      56 0x00fe1000 SP   fcp    fabric6
scsibus144 scsibus  144 0x07000060 SM   mscsi  fcbr118
pbay144   pbay     144 0x00000100 S    scsi   scsibus144
sd6096    sd       865 0x00000000 S    scsi   scsibus144
sd6097    sd       866 0x00000010 S    scsi   scsibus144
sd6098    sd       867 0x00000020 S    scsi   scsibus144
sd6099    sd       868 0x00000030 S    scsi   scsibus144
sd6100    sd       869 0x00000040 S    scsi   scsibus144
sd6101    sd       870 0x00000050 S    scsi   scsibus144
scsibus145 scsibus  145 0x06000060 SM   mscsi  fcbr118
pbay145   pbay     145 0x00000100 S    scsi   scsibus145
sd6102    sd       871 0x00000000 S    scsi   scsibus145
sd6103    sd       872 0x00000010 S    scsi   scsibus145
sd6104    sd       873 0x00000020 S    scsi   scsibus145
sd6105    sd       874 0x00000030 S    scsi   scsibus145
sd6106    sd       875 0x00000040 S    scsi   scsibus145
sd6107    sd       876 0x00000050 S    scsi   scsibus145
scsibus146 scsibus  146 0x05000070 SM   mscsi  fcbr118
pbay146   pbay     146 0x00000100 S    scsi   scsibus146
sd6108    sd       877 0x00000000 S    scsi   scsibus146
sd6109    sd       878 0x00000010 S    scsi   scsibus146
sd6110    sd       879 0x00000020 S    scsi   scsibus146
sd6111    sd       880 0x00000030 S    scsi   scsibus146
sd6112    sd       881 0x00000040 S    scsi   scsibus146
sd6113    sd       882 0x00000050 S    scsi   scsibus146
scsibus147 scsibus  147 0x04000070 SM   mscsi  fcbr118
pbay147   pbay     147 0x00000100 S    scsi   scsibus147
sd6114    sd       883 0x00000000 S    scsi   scsibus147
sd6115    sd       884 0x00000010 S    scsi   scsibus147
sd6116    sd       885 0x00000020 S    scsi   scsibus147
sd6117    sd       886 0x00000030 S    scsi   scsibus147
sd6118    sd       887 0x00000040 S    scsi   scsibus147
sd6119    sd       888 0x00000050 S    scsi   scsibus147
scsibus148 scsibus  148 0x03000060 SM   mscsi  fcbr118
pbay148   pbay     148 0x00000100 S    scsi   scsibus148
sd6120    sd       889 0x00000000 S    scsi   scsibus148
sd6121    sd       890 0x00000010 S    scsi   scsibus148
sd6122    sd       891 0x00000020 S    scsi   scsibus148
sd6123    sd       892 0x00000030 S    scsi   scsibus148
sd6124    sd       893 0x00000040 S    scsi   scsibus148
sd6125    sd       894 0x00000050 S    scsi   scsibus148
scsibus149 scsibus  149 0x02000060 SM   mscsi  fcbr118
pbay149   pbay     149 0x00000100 S    scsi   scsibus149
sd6126    sd       895 0x00000000 S    scsi   scsibus149
sd6127    sd       896 0x00000010 S    scsi   scsibus149
sd6128    sd       897 0x00000020 S    scsi   scsibus149
sd6129    sd       898 0x00000030 S    scsi   scsibus149
sd6130    sd       899 0x00000040 S    scsi   scsibus149
sd6131    sd       900 0x00000050 S    scsi   scsibus149
scsibus150 scsibus  150 0x01000070 SM   mscsi  fcbr118
pbay150   pbay     150 0x00000100 S    scsi   scsibus150
sd6132    sd       901 0x00000000 S    scsi   scsibus150
sd6133    sd       902 0x00000010 S    scsi   scsibus150
sd6134    sd       903 0x00000020 S    scsi   scsibus150
sd6135    sd       904 0x00000030 S    scsi   scsibus150
sd6136    sd       905 0x00000040 S    scsi   scsibus150
sd6137    sd       906 0x00000050 S    scsi   scsibus150
scsibus151 scsibus  151 0x00000070 SM   mscsi  fcbr118
pbay151   pbay     151 0x00000100 S    scsi   scsibus151
sd6138    sd       907 0x00000000 S    scsi   scsibus151
sd6139    sd       908 0x00000010 S    scsi   scsibus151
sd6140    sd       909 0x00000020 S    scsi   scsibus151
sd6141    sd       910 0x00000030 S    scsi   scsibus151
sd6142    sd       911 0x00000040 S    scsi   scsibus151
sd6143    sd       912 0x00000050 S    scsi   scsibus151
fcbr120   fcbr      57 0x00fe1800 SP   fcp    fabric6
scsibus160 scsibus  160 0x03000060 SM   mscsi  fcbr120
pbay160   pbay     160 0x00000100 S    scsi   scsibus160
sd6192    sd       961 0x00000000 S    scsi   scsibus160
sd6193    sd       962 0x00000010 S    scsi   scsibus160
sd6194    sd       963 0x00000020 S    scsi   scsibus160
sd6195    sd       964 0x00000030 S    scsi   scsibus160
sd6196    sd       965 0x00000040 S    scsi   scsibus160
sd6197    sd       966 0x00000050 S    scsi   scsibus160
scsibus161 scsibus  161 0x02000060 SM   mscsi  fcbr120
pbay161   pbay     161 0x00000100 S    scsi   scsibus161
sd6198    sd       967 0x00000000 S    scsi   scsibus161
sd6199    sd       968 0x00000010 S    scsi   scsibus161
sd6200    sd       969 0x00000020 S    scsi   scsibus161
sd6201    sd       970 0x00000030 S    scsi   scsibus161
sd6202    sd       971 0x00000040 S    scsi   scsibus161
sd6203    sd       972 0x00000050 S    scsi   scsibus161
scsibus162 scsibus  162 0x04000070 SM   mscsi  fcbr120
pbay162   pbay     162 0x00000100 S    scsi   scsibus162
sd6204    sd       973 0x00000000 S    scsi   scsibus162
sd6205    sd       974 0x00000010 S    scsi   scsibus162
sd6206    sd       975 0x00000020 S    scsi   scsibus162
sd6207    sd       976 0x00000030 S    scsi   scsibus162
sd6208    sd       977 0x00000040 S    scsi   scsibus162
sd6209    sd       978 0x00000050 S    scsi   scsibus162
scsibus163 scsibus  163 0x05000070 SM   mscsi  fcbr120
pbay163   pbay     163 0x00000100 S    scsi   scsibus163
sd6210    sd       979 0x00000000 S    scsi   scsibus163
sd6211    sd       980 0x00000010 S    scsi   scsibus163
sd6212    sd       981 0x00000020 S    scsi   scsibus163
sd6213    sd       982 0x00000030 S    scsi   scsibus163
sd6214    sd       983 0x00000040 S    scsi   scsibus163
sd6215    sd       984 0x00000050 S    scsi   scsibus163
scsibus164 scsibus  164 0x07000060 SM   mscsi  fcbr120
pbay164   pbay     164 0x00000100 S    scsi   scsibus164
sd6216    sd       985 0x00000000 S    scsi   scsibus164
sd6217    sd       986 0x00000010 S    scsi   scsibus164
sd6218    sd       987 0x00000020 S    scsi   scsibus164
sd6219    sd       988 0x00000030 S    scsi   scsibus164
sd6220    sd       989 0x00000040 S    scsi   scsibus164
sd6221    sd       990 0x00000050 S    scsi   scsibus164
scsibus165 scsibus  165 0x06000060 SM   mscsi  fcbr120
pbay165   pbay     165 0x00000100 S    scsi   scsibus165
sd6222    sd       991 0x00000000 S    scsi   scsibus165
sd6223    sd       992 0x00000010 S    scsi   scsibus165
sd6224    sd       993 0x00000020 S    scsi   scsibus165
sd6225    sd       994 0x00000030 S    scsi   scsibus165
sd6226    sd       995 0x00000040 S    scsi   scsibus165
sd6227    sd       996 0x00000050 S    scsi   scsibus165
scsibus166 scsibus  166 0x01000070 SM   mscsi  fcbr120
pbay166   pbay     166 0x00000100 S    scsi   scsibus166
sd6228    sd       997 0x00000000 S    scsi   scsibus166
sd6229    sd       998 0x00000010 S    scsi   scsibus166
```

Appendix C – Tunable Parameters

```
sd6230    sd       999 0x00000020 S    scsi   scsibus166
sd6231    sd      1000 0x00000030 S    scsi   scsibus166
sd6232    sd      1001 0x00000040 S    scsi   scsibus166
sd6233    sd      1002 0x00000050 S    scsi   scsibus166
scsibus167 scsibus  167 0x00000070 SM   mscsi  fcbr120
pbay167   pbay     167 0x00000100 S    scsi   scsibus167
sd6234    sd      1003 0x00000000 S    scsi   scsibus167
sd6235    sd      1004 0x00000010 S    scsi   scsibus167
sd6236    sd      1005 0x00000020 S    scsi   scsibus167
sd6237    sd      1006 0x00000030 S    scsi   scsibus167
sd6238    sd      1007 0x00000040 S    scsi   scsibus167
sd6239    sd      1008 0x00000050 S    scsi   scsibus167
fcbr119   fcbr     58 0x00fe2000 SP   fcp    fabric6
scsibus152 scsibus  152 0x03000060 SM   mscsi  fcbr119
pbay152   pbay     152 0x00000100 S    scsi   scsibus152
sd6144    sd       913 0x00000000 S    scsi   scsibus152
sd6145    sd       914 0x00000010 S    scsi   scsibus152
sd6146    sd       915 0x00000020 S    scsi   scsibus152
sd6147    sd       916 0x00000030 S    scsi   scsibus152
sd6148    sd       917 0x00000040 S    scsi   scsibus152
sd6149    sd       918 0x00000050 S    scsi   scsibus152
scsibus153 scsibus  153 0x02000060 SM   mscsi  fcbr119
pbay153   pbay     153 0x00000100 S    scsi   scsibus153
sd6150    sd       919 0x00000000 S    scsi   scsibus153
sd6151    sd       920 0x00000010 S    scsi   scsibus153
sd6152    sd       921 0x00000020 S    scsi   scsibus153
sd6153    sd       922 0x00000030 S    scsi   scsibus153
sd6154    sd       923 0x00000040 S    scsi   scsibus153
sd6155    sd       924 0x00000050 S    scsi   scsibus153
scsibus154 scsibus  154 0x00000070 SM   mscsi  fcbr119
pbay154   pbay     154 0x00000100 S    scsi   scsibus154
sd6156    sd       925 0x00000000 S    scsi   scsibus154
sd6157    sd       926 0x00000010 S    scsi   scsibus154
sd6158    sd       927 0x00000020 S    scsi   scsibus154
sd6159    sd       928 0x00000030 S    scsi   scsibus154
sd6160    sd       929 0x00000040 S    scsi   scsibus154
sd6161    sd       930 0x00000050 S    scsi   scsibus154
scsibus155 scsibus  155 0x01000070 SM   mscsi  fcbr119
pbay155   pbay     155 0x00000100 S    scsi   scsibus155
sd6162    sd       931 0x00000000 S    scsi   scsibus155
sd6163    sd       932 0x00000010 S    scsi   scsibus155
sd6164    sd       933 0x00000020 S    scsi   scsibus155
sd6165    sd       934 0x00000030 S    scsi   scsibus155
sd6166    sd       935 0x00000040 S    scsi   scsibus155
sd6167    sd      1501 0x00000050 S    scsi   scsibus155
scsibus156 scsibus  156 0x07000060 SM   mscsi  fcbr119
pbay156   pbay     156 0x00000100 S    scsi   scsibus156
sd6168    sd       937 0x00000000 S    scsi   scsibus156
sd6169    sd       938 0x00000010 S    scsi   scsibus156
sd6170    sd       939 0x00000020 S    scsi   scsibus156
sd6171    sd       940 0x00000030 S    scsi   scsibus156
sd6172    sd       941 0x00000040 S    scsi   scsibus156
sd6173    sd       942 0x00000050 S    scsi   scsibus156
scsibus157 scsibus  157 0x06000060 SM   mscsi  fcbr119
pbay157   pbay     157 0x00000100 S    scsi   scsibus157
sd6174    sd       943 0x00000000 S    scsi   scsibus157
sd6175    sd       944 0x00000010 S    scsi   scsibus157
sd6176    sd       945 0x00000020 S    scsi   scsibus157
sd6177    sd       946 0x00000030 S    scsi   scsibus157
sd6178    sd       947 0x00000040 S    scsi   scsibus157
sd6179    sd       948 0x00000050 S    scsi   scsibus157
scsibus158 scsibus  158 0x04000070 SM   mscsi  fcbr119
pbay158   pbay     158 0x00000100 S    scsi   scsibus158
sd6180    sd       949 0x00000000 S    scsi   scsibus158
sd6181    sd       950 0x00000010 S    scsi   scsibus158
sd6182    sd       951 0x00000020 S    scsi   scsibus158
sd6183    sd       952 0x00000030 S    scsi   scsibus158
sd6184    sd       953 0x00000040 S    scsi   scsibus158
sd6185    sd       954 0x00000050 S    scsi   scsibus158
scsibus159 scsibus  159 0x05000070 SM   mscsi  fcbr119
pbay159   pbay     159 0x00000100 S    scsi   scsibus159
sd6186    sd       955 0x00000000 S    scsi   scsibus159
sd6187    sd       956 0x00000010 S    scsi   scsibus159
sd6188    sd       957 0x00000020 S    scsi   scsibus159
sd6189    sd       958 0x00000030 S    scsi   scsibus159
sd6190    sd       959 0x00000040 S    scsi   scsibus159
sd6191    sd       960 0x00000050 S    scsi   scsibus159
fcbr121   fcbr     59 0x00fe2800 SP   fcp    fabric6
scsibus168 scsibus  168 0x03000060 SM   mscsi  fcbr121
pbay168   pbay     168 0x00000100 S    scsi   scsibus168
sd6240    sd      1009 0x00000000 S    scsi   scsibus168
sd6241    sd      1010 0x00000010 S    scsi   scsibus168
sd6242    sd      1011 0x00000020 S    scsi   scsibus168
sd6243    sd      1012 0x00000030 S    scsi   scsibus168
sd6244    sd      1013 0x00000040 S    scsi   scsibus168
sd6245    sd      1014 0x00000050 S    scsi   scsibus168
scsibus169 scsibus  169 0x02000060 SM   mscsi  fcbr121
pbay169   pbay     169 0x00000100 S    scsi   scsibus169
sd6246    sd      1015 0x00000000 S    scsi   scsibus169
sd6247    sd      1016 0x00000010 S    scsi   scsibus169
sd6248    sd      1017 0x00000020 S    scsi   scsibus169
sd6249    sd      1018 0x00000030 S    scsi   scsibus169
sd6250    sd      1019 0x00000040 S    scsi   scsibus169
sd6251    sd      1020 0x00000050 S    scsi   scsibus169
scsibus170 scsibus  170 0x01000070 SM   mscsi  fcbr121
pbay170   pbay     170 0x00000100 S    scsi   scsibus170
sd6252    sd      1021 0x00000000 S    scsi   scsibus170
sd6253    sd      1022 0x00000010 S    scsi   scsibus170
sd6254    sd      1023 0x00000020 S    scsi   scsibus170
sd6255    sd      1024 0x00000030 S    scsi   scsibus170
sd6256    sd      1025 0x00000040 S    scsi   scsibus170
sd6257    sd      1026 0x00000050 S    scsi   scsibus170
scsibus171 scsibus  171 0x00000070 SM   mscsi  fcbr121
pbay171   pbay     171 0x00000100 S    scsi   scsibus171
sd6258    sd      1027 0x00000000 S    scsi   scsibus171
sd6259    sd      1028 0x00000010 S    scsi   scsibus171
sd6260    sd      1029 0x00000020 S    scsi   scsibus171
sd6261    sd      1030 0x00000030 S    scsi   scsibus171
sd6262    sd      1031 0x00000040 S    scsi   scsibus171
sd6263    sd      1032 0x00000050 S    scsi   scsibus171
scsibus172 scsibus  172 0x07000060 SM   mscsi  fcbr121
pbay172   pbay     172 0x00000100 S    scsi   scsibus172
sd6264    sd      1033 0x00000000 S    scsi   scsibus172
sd6265    sd      1034 0x00000010 S    scsi   scsibus172
sd6266    sd      1035 0x00000020 S    scsi   scsibus172
sd6267    sd      1036 0x00000030 S    scsi   scsibus172
sd6268    sd      1037 0x00000040 S    scsi   scsibus172
sd6269    sd      1038 0x00000050 S    scsi   scsibus172
scsibus173 scsibus  173 0x06000060 SM   mscsi  fcbr121
pbay173   pbay     173 0x00000100 S    scsi   scsibus173
sd6270    sd      1039 0x00000000 S    scsi   scsibus173
sd6271    sd      1040 0x00000010 S    scsi   scsibus173
sd6272    sd      1041 0x00000020 S    scsi   scsibus173
sd6273    sd      1042 0x00000030 S    scsi   scsibus173
sd6274    sd      1043 0x00000040 S    scsi   scsibus173
sd6275    sd      1044 0x00000050 S    scsi   scsibus173
scsibus174 scsibus  174 0x05000070 SM   mscsi  fcbr121
pbay174   pbay     174 0x00000100 S    scsi   scsibus174
sd6276    sd      1045 0x00000000 S    scsi   scsibus174
sd6277    sd      1046 0x00000010 S    scsi   scsibus174
sd6278    sd      1047 0x00000020 S    scsi   scsibus174
sd6279    sd      1048 0x00000030 S    scsi   scsibus174
sd6280    sd      1049 0x00000040 S    scsi   scsibus174
sd6281    sd      1050 0x00000050 S    scsi   scsibus174
scsibus175 scsibus  175 0x04000070 SM   mscsi  fcbr121
pbay175   pbay     175 0x00000100 S    scsi   scsibus175
sd6282    sd      1051 0x00000000 S    scsi   scsibus175
sd6283    sd      1052 0x00000010 S    scsi   scsibus175
sd6284    sd      1053 0x00000020 S    scsi   scsibus175
sd6285    sd      1054 0x00000030 S    scsi   scsibus175
sd6286    sd      1055 0x00000040 S    scsi   scsibus175
sd6287    sd      1056 0x00000050 S    scsi   scsibus175
fcbr115   fcbr     38 0x00fe3000 SP   fcp    fabric6
scsibus120 scsibus  120 0x06000060 SM   mscsi  fcbr115
scsibus121 scsibus  121 0x07000060 SM   mscsi  fcbr115
scsibus122 scsibus  122 0x04000070 SM   mscsi  fcbr115
scsibus123 scsibus  123 0x05000070 SM   mscsi  fcbr115
scsibus124 scsibus  124 0x02000060 SM   mscsi  fcbr115
scsibus125 scsibus  125 0x03000060 SM   mscsi  fcbr115
scsibus126 scsibus  126 0x00000070 SM   mscsi  fcbr115
scsibus127 scsibus  127 0x01000070 SM   mscsi  fcbr115
sdi0      sdi        0 0x00ffffff SM   sdi    ff10
ff66      ff        66 0x00000006 SP   pci    quad2
fabric7   fabric     6 0x00fe5800 SM   fc     ff66
fcbr124   fcbr      46 0x00fe0000 SP   fcp    fabric7
scsibus192 scsibus  192 0x02000060 SM   mscsi  fcbr124
pbay192   pbay     192 0x00000100 S    scsi   scsibus192
sd6288    sd      1153 0x00000000 S    scsi   scsibus192
sd6289    sd      1154 0x00000010 S    scsi   scsibus192
sd6290    sd      1155 0x00000020 S    scsi   scsibus192
sd6291    sd      1156 0x00000030 S    scsi   scsibus192
sd6292    sd      1157 0x00000040 S    scsi   scsibus192
sd6293    sd      1158 0x00000050 S    scsi   scsibus192
scsibus193 scsibus  193 0x07000060 SM   mscsi  fcbr124
pbay193   pbay     193 0x00000100 S    scsi   scsibus193
sd6294    sd      1159 0x00000000 S    scsi   scsibus193
sd6295    sd      1160 0x00000010 S    scsi   scsibus193
sd6296    sd      1161 0x00000020 S    scsi   scsibus193
sd6297    sd      1162 0x00000030 S    scsi   scsibus193
sd6298    sd      1163 0x00000040 S    scsi   scsibus193
sd6299    sd      1164 0x00000050 S    scsi   scsibus193
scsibus194 scsibus  194 0x00000070 SM   mscsi  fcbr124
pbay194   pbay     194 0x00000100 S    scsi   scsibus194
sd6300    sd      1165 0x00000000 S    scsi   scsibus194
sd6301    sd      1166 0x00000010 S    scsi   scsibus194
sd6302    sd      1167 0x00000020 S    scsi   scsibus194
sd6303    sd      1168 0x00000030 S    scsi   scsibus194
sd6304    sd      1169 0x00000040 S    scsi   scsibus194
```

Appendix C – Tunable Parameters

```
sd6305    sd      1170  0x00000050 S    scsi    scsibus194
scsibus195 scsibus   195 0x01000070 SM   mscsi   fcbr124
pbay195   pbay     195  0x00000100 S    scsi    scsibus195
sd6306    sd      1171  0x00000000 S    scsi    scsibus195
sd6307    sd      1172  0x00000010 S    scsi    scsibus195
sd6308    sd      1173  0x00000020 S    scsi    scsibus195
sd6309    sd      1174  0x00000030 S    scsi    scsibus195
sd6310    sd      1175  0x00000040 S    scsi    scsibus195
sd6311    sd      1176  0x00000050 S    scsi    scsibus195
scsibus196 scsibus   196 0x06000060 SM   mscsi   fcbr124
pbay196   pbay     196  0x00000100 S    scsi    scsibus196
sd6312    sd      1177  0x00000000 S    scsi    scsibus196
sd6313    sd      1178  0x00000010 S    scsi    scsibus196
sd6314    sd      1179  0x00000020 S    scsi    scsibus196
sd6315    sd      1180  0x00000030 S    scsi    scsibus196
sd6316    sd      1181  0x00000040 S    scsi    scsibus196
sd6317    sd      1182  0x00000050 S    scsi    scsibus196
scsibus197 scsibus   197 0x03000060 SM   mscsi   fcbr124
pbay197   pbay     197  0x00000100 S    scsi    scsibus197
sd6318    sd      1183  0x00000000 S    scsi    scsibus197
sd6319    sd      1184  0x00000010 S    scsi    scsibus197
sd6320    sd      1185  0x00000020 S    scsi    scsibus197
sd6321    sd      1186  0x00000030 S    scsi    scsibus197
sd6322    sd      1187  0x00000040 S    scsi    scsibus197
sd6323    sd      1188  0x00000050 S    scsi    scsibus197
scsibus198 scsibus   198 0x04000070 SM   mscsi   fcbr124
pbay198   pbay     198  0x00000100 S    scsi    scsibus198
sd6324    sd      1189  0x00000000 S    scsi    scsibus198
sd6325    sd      1190  0x00000010 S    scsi    scsibus198
sd6326    sd      1191  0x00000020 S    scsi    scsibus198
sd6327    sd      1192  0x00000030 S    scsi    scsibus198
sd6328    sd      1193  0x00000040 S    scsi    scsibus198
sd6329    sd      1194  0x00000050 S    scsi    scsibus198
scsibus199 scsibus   199 0x05000070 SM   mscsi   fcbr124
pbay199   pbay     199  0x00000100 S    scsi    scsibus199
sd6330    sd      1195  0x00000000 S    scsi    scsibus199
sd6331    sd      1196  0x00000010 S    scsi    scsibus199
sd6332    sd      1197  0x00000020 S    scsi    scsibus199
sd6333    sd      1198  0x00000030 S    scsi    scsibus199
sd6334    sd      1199  0x00000040 S    scsi    scsibus199
sd6335    sd      1200  0x00000050 S    scsi    scsibus199
fcbr126   fcbr     47  0x00fe0800 SP   fcp     fabric7
scsibus208 scsibus   208 0x07000060 SM   mscsi   fcbr126
pbay208   pbay     208  0x00000100 S    scsi    scsibus208
sd6384    sd      1249  0x00000000 S    scsi    scsibus208
sd6385    sd      1250  0x00000010 S    scsi    scsibus208
sd6386    sd      1251  0x00000020 S    scsi    scsibus208
sd6387    sd      1252  0x00000030 S    scsi    scsibus208
sd6388    sd      1253  0x00000040 S    scsi    scsibus208
sd6389    sd      1254  0x00000050 S    scsi    scsibus208
scsibus209 scsibus   209 0x06000060 SM   mscsi   fcbr126
pbay209   pbay     209  0x00000100 S    scsi    scsibus209
sd6390    sd      1255  0x00000000 S    scsi    scsibus209
sd6391    sd      1256  0x00000010 S    scsi    scsibus209
sd6392    sd      1257  0x00000020 S    scsi    scsibus209
sd6393    sd      1258  0x00000030 S    scsi    scsibus209
sd6394    sd      1259  0x00000040 S    scsi    scsibus209
sd6395    sd      1260  0x00000050 S    scsi    scsibus209
scsibus210 scsibus   210 0x01000070 SM   mscsi   fcbr126
pbay210   pbay     210  0x00000100 S    scsi    scsibus210
sd6396    sd      1261  0x00000000 S    scsi    scsibus210
sd6397    sd      1262  0x00000010 S    scsi    scsibus210
sd6398    sd      1263  0x00000020 S    scsi    scsibus210
sd6399    sd      1264  0x00000030 S    scsi    scsibus210
sd6400    sd      1265  0x00000040 S    scsi    scsibus210
sd6401    sd      1266  0x00000050 S    scsi    scsibus210
scsibus211 scsibus   211 0x00000070 SM   mscsi   fcbr126
pbay211   pbay     211  0x00000100 S    scsi    scsibus211
sd6402    sd      1267  0x00000000 S    scsi    scsibus211
sd6403    sd      1268  0x00000010 S    scsi    scsibus211
sd6404    sd      1269  0x00000020 S    scsi    scsibus211
sd6405    sd      1270  0x00000030 S    scsi    scsibus211
sd6406    sd      1271  0x00000040 S    scsi    scsibus211
sd6407    sd      1272  0x00000050 S    scsi    scsibus211
scsibus212 scsibus   212 0x03000060 SM   mscsi   fcbr126
pbay212   pbay     212  0x00000100 S    scsi    scsibus212
sd6408    sd      1273  0x00000000 S    scsi    scsibus212
sd6409    sd      1274  0x00000010 S    scsi    scsibus212
sd6410    sd      1275  0x00000020 S    scsi    scsibus212
sd6411    sd      1276  0x00000030 S    scsi    scsibus212
sd6412    sd      1277  0x00000040 S    scsi    scsibus212
sd6413    sd      1278  0x00000050 S    scsi    scsibus212
scsibus213 scsibus   213 0x02000060 SM   mscsi   fcbr126
pbay213   pbay     213  0x00000100 S    scsi    scsibus213
sd6414    sd      1279  0x00000000 S    scsi    scsibus213
sd6415    sd      1280  0x00000010 S    scsi    scsibus213
sd6416    sd      1281  0x00000020 S    scsi    scsibus213
sd6417    sd      1282  0x00000030 S    scsi    scsibus213
sd6418    sd      1283  0x00000040 S    scsi    scsibus213
sd6419    sd      1284  0x00000050 S    scsi    scsibus213
scsibus214 scsibus   214 0x05000070 SM   mscsi   fcbr126
pbay214   pbay     214  0x00000100 S    scsi    scsibus214
sd6420    sd      1285  0x00000000 S    scsi    scsibus214
sd6421    sd      1286  0x00000010 S    scsi    scsibus214
sd6422    sd      1287  0x00000020 S    scsi    scsibus214
sd6423    sd      1288  0x00000030 S    scsi    scsibus214
sd6424    sd      1289  0x00000040 S    scsi    scsibus214
sd6425    sd      1290  0x00000050 S    scsi    scsibus214
scsibus215 scsibus   215 0x04000070 SM   mscsi   fcbr126
pbay215   pbay     215  0x00000100 S    scsi    scsibus215
sd6426    sd      1291  0x00000000 S    scsi    scsibus215
sd6427    sd      1292  0x00000010 S    scsi    scsibus215
sd6428    sd      1293  0x00000020 S    scsi    scsibus215
sd6429    sd      1294  0x00000030 S    scsi    scsibus215
sd6430    sd      1295  0x00000040 S    scsi    scsibus215
sd6431    sd      1296  0x00000050 S    scsi    scsibus215
fcbr127   fcbr     48  0x00fe1000 SP   fcp     fabric7
scsibus216 scsibus   216 0x06000060 SM   mscsi   fcbr127
pbay216   pbay     216  0x00000100 S    scsi    scsibus216
sd6432    sd      1297  0x00000000 S    scsi    scsibus216
sd6433    sd      1298  0x00000010 S    scsi    scsibus216
sd6434    sd      1299  0x00000020 S    scsi    scsibus216
sd6435    sd      1300  0x00000030 S    scsi    scsibus216
sd6436    sd      1301  0x00000040 S    scsi    scsibus216
sd6437    sd      1302  0x00000050 S    scsi    scsibus216
scsibus217 scsibus   217 0x07000060 SM   mscsi   fcbr127
pbay217   pbay     217  0x00000100 S    scsi    scsibus217
sd6438    sd      1303  0x00000000 S    scsi    scsibus217
sd6439    sd      1304  0x00000010 S    scsi    scsibus217
sd6440    sd      1305  0x00000020 S    scsi    scsibus217
sd6441    sd      1306  0x00000030 S    scsi    scsibus217
sd6442    sd      1307  0x00000040 S    scsi    scsibus217
sd6443    sd      1308  0x00000050 S    scsi    scsibus217
scsibus218 scsibus   218 0x04000070 SM   mscsi   fcbr127
pbay218   pbay     218  0x00000100 S    scsi    scsibus218
sd6444    sd      1309  0x00000000 S    scsi    scsibus218
sd6445    sd      1347  0x00000010 S    scsi    scsibus218
sd6446    sd      1311  0x00000020 S    scsi    scsibus218
sd6447    sd      1312  0x00000030 S    scsi    scsibus218
sd6448    sd      1313  0x00000040 S    scsi    scsibus218
sd6449    sd      1314  0x00000050 S    scsi    scsibus218
scsibus219 scsibus   219 0x05000070 SM   mscsi   fcbr127
pbay219   pbay     219  0x00000100 S    scsi    scsibus219
sd6450    sd      1315  0x00000000 S    scsi    scsibus219
sd6451    sd      1316  0x00000010 S    scsi    scsibus219
sd6452    sd      1317  0x00000020 S    scsi    scsibus219
sd6453    sd      1318  0x00000030 S    scsi    scsibus219
sd6454    sd      1319  0x00000040 S    scsi    scsibus219
sd6455    sd      1320  0x00000050 S    scsi    scsibus219
scsibus220 scsibus   220 0x02000060 SM   mscsi   fcbr127
pbay220   pbay     220  0x00000100 S    scsi    scsibus220
sd6456    sd      1321  0x00000000 S    scsi    scsibus220
sd6457    sd      1322  0x00000010 S    scsi    scsibus220
sd6458    sd      1323  0x00000020 S    scsi    scsibus220
sd6459    sd      1324  0x00000030 S    scsi    scsibus220
sd6460    sd      1325  0x00000040 S    scsi    scsibus220
sd6461    sd      1326  0x00000050 S    scsi    scsibus220
scsibus221 scsibus   221 0x03000060 SM   mscsi   fcbr127
pbay221   pbay     221  0x00000100 S    scsi    scsibus221
sd6462    sd      1327  0x00000000 S    scsi    scsibus221
sd6463    sd      1328  0x00000010 S    scsi    scsibus221
sd6464    sd      1329  0x00000020 S    scsi    scsibus221
sd6465    sd      1330  0x00000030 S    scsi    scsibus221
sd6466    sd      1331  0x00000040 S    scsi    scsibus221
sd6467    sd      1332  0x00000050 S    scsi    scsibus221
scsibus222 scsibus   222 0x00000070 SM   mscsi   fcbr127
pbay222   pbay     222  0x00000100 S    scsi    scsibus222
sd6468    sd      1333  0x00000000 S    scsi    scsibus222
sd6469    sd      1334  0x00000010 S    scsi    scsibus222
sd6470    sd      1335  0x00000020 S    scsi    scsibus222
sd6471    sd      1336  0x00000030 S    scsi    scsibus222
sd6472    sd      1337  0x00000040 S    scsi    scsibus222
sd6473    sd      1338  0x00000050 S    scsi    scsibus222
scsibus223 scsibus   223 0x01000070 SM   mscsi   fcbr127
pbay223   pbay     223  0x00000100 S    scsi    scsibus223
sd6474    sd      1339  0x00000000 S    scsi    scsibus223
sd6475    sd      1340  0x00000010 S    scsi    scsibus223
sd6476    sd      1341  0x00000020 S    scsi    scsibus223
sd6477    sd      1342  0x00000030 S    scsi    scsibus223
sd6478    sd      1343  0x00000040 S    scsi    scsibus223
sd6479    sd      1344  0x00000050 S    scsi    scsibus223
fcbr125   fcbr     49  0x00fe1800 SP   fcp     fabric7
scsibus200 scsibus   200 0x06000060 SM   mscsi   fcbr125
pbay200   pbay     200  0x00000100 S    scsi    scsibus200
sd6336    sd      1201  0x00000000 S    scsi    scsibus200
sd6337    sd      1202  0x00000010 S    scsi    scsibus200
sd6338    sd      1203  0x00000020 S    scsi    scsibus200
sd6339    sd      1204  0x00000030 S    scsi    scsibus200
```

Appendix C – Tunable Parameters

```
sd6340 sd 1205 0x00000040 S scsi scsibus200
sd6341 sd 1206 0x00000050 S scsi scsibus200
scsibus201 scsibus 201 0x07000060 SM mscsi fcbr125
pbay201 pbay 201 0x00000100 S scsi scsibus201
sd6342 sd 1207 0x00000000 S scsi scsibus201
sd6343 sd 1208 0x00000010 S scsi scsibus201
sd6344 sd 1209 0x00000020 S scsi scsibus201
sd6345 sd 1210 0x00000030 S scsi scsibus201
sd6346 sd 1211 0x00000040 S scsi scsibus201
sd6347 sd 1212 0x00000050 S scsi scsibus201
scsibus202 scsibus 202 0x04000070 SM mscsi fcbr125
pbay202 pbay 202 0x00000100 S scsi scsibus202
sd6348 sd 1213 0x00000000 S scsi scsibus202
sd6349 sd 1214 0x00000010 S scsi scsibus202
sd6350 sd 1215 0x00000020 S scsi scsibus202
sd6351 sd 1216 0x00000030 S scsi scsibus202
sd6352 sd 1217 0x00000040 S scsi scsibus202
sd6353 sd 1218 0x00000050 S scsi scsibus202
scsibus203 scsibus 203 0x05000070 SM mscsi fcbr125
pbay203 pbay 203 0x00000100 S scsi scsibus203
sd6354 sd 1219 0x00000000 S scsi scsibus203
sd6355 sd 1220 0x00000010 S scsi scsibus203
sd6356 sd 1221 0x00000020 S scsi scsibus203
sd6357 sd 1222 0x00000030 S scsi scsibus203
sd6358 sd 1223 0x00000040 S scsi scsibus203
sd6359 sd 1224 0x00000050 S scsi scsibus203
scsibus204 scsibus 204 0x02000060 SM mscsi fcbr125
pbay204 pbay 204 0x00000100 S scsi scsibus204
sd6360 sd 1225 0x00000000 S scsi scsibus204
sd6361 sd 1226 0x00000010 S scsi scsibus204
sd6362 sd 1227 0x00000020 S scsi scsibus204
sd6363 sd 1228 0x00000030 S scsi scsibus204
sd6364 sd 1229 0x00000040 S scsi scsibus204
sd6365 sd 1230 0x00000050 S scsi scsibus204
scsibus205 scsibus 205 0x03000060 SM mscsi fcbr125
pbay205 pbay 205 0x00000100 S scsi scsibus205
sd6366 sd 1231 0x00000000 S scsi scsibus205
sd6367 sd 1232 0x00000010 S scsi scsibus205
sd6368 sd 1233 0x00000020 S scsi scsibus205
sd6369 sd 1234 0x00000030 S scsi scsibus205
sd6370 sd 1235 0x00000040 S scsi scsibus205
sd6371 sd 1236 0x00000050 S scsi scsibus205
scsibus206 scsibus 206 0x00000070 SM mscsi fcbr125
pbay206 pbay 206 0x00000100 S scsi scsibus206
sd6372 sd 1237 0x00000000 S scsi scsibus206
sd6373 sd 1238 0x00000010 S scsi scsibus206
sd6374 sd 1239 0x00000020 S scsi scsibus206
sd6375 sd 1240 0x00000030 S scsi scsibus206
sd6376 sd 1241 0x00000040 S scsi scsibus206
sd6377 sd 1242 0x00000050 S scsi scsibus206
scsibus207 scsibus 207 0x01000070 SM mscsi fcbr125
pbay207 pbay 207 0x00000100 S scsi scsibus207
sd6378 sd 1243 0x00000000 S scsi scsibus207
sd6379 sd 1244 0x00000010 S scsi scsibus207
sd6380 sd 1245 0x00000020 S scsi scsibus207
sd6381 sd 1246 0x00000030 S scsi scsibus207
sd6382 sd 1247 0x00000040 S scsi scsibus207
sd6383 sd 1248 0x00000050 S scsi scsibus207
fcbr129 fcbr 51 0x00fe2800 SP fcp fabric7
scsibus232 scsibus 232 0x03000060 SM mscsi fcbr129
pbay232 pbay 232 0x00000100 S scsi scsibus232
sd6528 sd 1393 0x00000000 S scsi scsibus232
sd6529 sd 1394 0x00000010 S scsi scsibus232
sd6530 sd 1395 0x00000020 S scsi scsibus232
sd6531 sd 1396 0x00000030 S scsi scsibus232
sd6532 sd 1397 0x00000040 S scsi scsibus232
sd6533 sd 1398 0x00000050 S scsi scsibus232
scsibus233 scsibus 233 0x06000060 SM mscsi fcbr129
pbay233 pbay 233 0x00000100 S scsi scsibus233
sd6534 sd 1399 0x00000000 S scsi scsibus233
sd6535 sd 1400 0x00000010 S scsi scsibus233
sd6536 sd 1401 0x00000020 S scsi scsibus233
sd6537 sd 1402 0x00000030 S scsi scsibus233
sd6538 sd 1403 0x00000040 S scsi scsibus233
sd6539 sd 1404 0x00000050 S scsi scsibus233
scsibus234 scsibus 234 0x01000070 SM mscsi fcbr129
pbay234 pbay 234 0x00000100 S scsi scsibus234
sd6540 sd 1405 0x00000000 S scsi scsibus234
sd6541 sd 1406 0x00000010 S scsi scsibus234
sd6542 sd 1407 0x00000020 S scsi scsibus234
sd6543 sd 1408 0x00000030 S scsi scsibus234
sd6544 sd 1409 0x00000040 S scsi scsibus234
sd6545 sd 1410 0x00000050 S scsi scsibus234
scsibus235 scsibus 235 0x04000070 SM mscsi fcbr129
pbay235 pbay 235 0x00000100 S scsi scsibus235
sd6546 sd 1411 0x00000000 S scsi scsibus235
sd6547 sd 1412 0x00000010 S scsi scsibus235
sd6548 sd 1413 0x00000020 S scsi scsibus235
sd6549 sd 1414 0x00000030 S scsi scsibus235
sd6550 sd 1415 0x00000040 S scsi scsibus235
sd6551 sd 1416 0x00000050 S scsi scsibus235
scsibus236 scsibus 236 0x07000060 SM mscsi fcbr129
pbay236 pbay 236 0x00000100 S scsi scsibus236
sd6552 sd 1417 0x00000000 S scsi scsibus236
sd6553 sd 1418 0x00000010 S scsi scsibus236
sd6554 sd 1419 0x00000020 S scsi scsibus236
sd6555 sd 1420 0x00000030 S scsi scsibus236
sd6556 sd 1421 0x00000040 S scsi scsibus236
sd6557 sd 1422 0x00000050 S scsi scsibus236
scsibus237 scsibus 237 0x02000060 SM mscsi fcbr129
pbay237 pbay 237 0x00000100 S scsi scsibus237
sd6558 sd 1423 0x00000000 S scsi scsibus237
sd6559 sd 1424 0x00000010 S scsi scsibus237
sd6560 sd 1425 0x00000020 S scsi scsibus237
sd6561 sd 1426 0x00000030 S scsi scsibus237
sd6562 sd 1427 0x00000040 S scsi scsibus237
sd6563 sd 1428 0x00000050 S scsi scsibus237
scsibus238 scsibus 238 0x05000070 SM mscsi fcbr129
pbay238 pbay 238 0x00000100 S scsi scsibus238
sd6564 sd 1429 0x00000000 S scsi scsibus238
sd6565 sd 1430 0x00000010 S scsi scsibus238
sd6566 sd 1431 0x00000020 S scsi scsibus238
sd6567 sd 1432 0x00000030 S scsi scsibus238
sd6568 sd 1433 0x00000040 S scsi scsibus238
sd6569 sd 1434 0x00000050 S scsi scsibus238
scsibus239 scsibus 239 0x00000070 SM mscsi fcbr129
pbay239 pbay 239 0x00000100 S scsi scsibus239
sd6570 sd 1435 0x00000000 S scsi scsibus239
sd6571 sd 1436 0x00000010 S scsi scsibus239
sd6572 sd 1437 0x00000020 S scsi scsibus239
sd6573 sd 1438 0x00000030 S scsi scsibus239
sd6574 sd 1439 0x00000040 S scsi scsibus239
sd6575 sd 1440 0x00000050 S scsi scsibus239
fcbr128 fcbr 52 0x00fe3000 SP fcp fabric7
scsibus224 scsibus 224 0x03000060 SM mscsi fcbr128
pbay224 pbay 224 0x00000100 S scsi scsibus224
sd6480 sd 1345 0x00000000 S scsi scsibus224
sd6481 sd 1346 0x00000010 S scsi scsibus224
sd6482 sd 1610 0x00000020 S scsi scsibus224
sd6483 sd 1348 0x00000030 S scsi scsibus224
sd6484 sd 1349 0x00000040 S scsi scsibus224
sd6485 sd 1350 0x00000050 S scsi scsibus224
scsibus225 scsibus 225 0x06000060 SM mscsi fcbr128
pbay225 pbay 225 0x00000100 S scsi scsibus225
sd6486 sd 1351 0x00000000 S scsi scsibus225
sd6487 sd 1352 0x00000010 S scsi scsibus225
sd6488 sd 1353 0x00000020 S scsi scsibus225
sd6489 sd 1354 0x00000030 S scsi scsibus225
sd6490 sd 1355 0x00000040 S scsi scsibus225
sd6491 sd 1356 0x00000050 S scsi scsibus225
scsibus226 scsibus 226 0x01000070 SM mscsi fcbr128
pbay226 pbay 226 0x00000100 S scsi scsibus226
sd6492 sd 1357 0x00000000 S scsi scsibus226
sd6493 sd 1358 0x00000010 S scsi scsibus226
sd6494 sd 1359 0x00000020 S scsi scsibus226
sd6495 sd 1360 0x00000030 S scsi scsibus226
sd6496 sd 1361 0x00000040 S scsi scsibus226
sd6497 sd 1362 0x00000050 S scsi scsibus226
scsibus227 scsibus 227 0x00000070 SM mscsi fcbr128
pbay227 pbay 227 0x00000100 S scsi scsibus227
sd6498 sd 1363 0x00000000 S scsi scsibus227
sd6499 sd 1364 0x00000010 S scsi scsibus227
sd6500 sd 1365 0x00000020 S scsi scsibus227
sd6501 sd 1366 0x00000030 S scsi scsibus227
sd6502 sd 1367 0x00000040 S scsi scsibus227
sd6503 sd 1368 0x00000050 S scsi scsibus227
scsibus228 scsibus 228 0x07000060 SM mscsi fcbr128
pbay228 pbay 228 0x00000100 S scsi scsibus228
sd6504 sd 1369 0x00000000 S scsi scsibus228
sd6505 sd 1370 0x00000010 S scsi scsibus228
sd6506 sd 1371 0x00000020 S scsi scsibus228
sd6507 sd 1372 0x00000030 S scsi scsibus228
sd6508 sd 1373 0x00000040 S scsi scsibus228
sd6509 sd 1374 0x00000050 S scsi scsibus228
scsibus229 scsibus 229 0x02000060 SM mscsi fcbr128
pbay229 pbay 229 0x00000100 S scsi scsibus229
sd6510 sd 1375 0x00000000 S scsi scsibus229
sd6511 sd 1376 0x00000010 S scsi scsibus229
sd6512 sd 1377 0x00000020 S scsi scsibus229
sd6513 sd 1378 0x00000030 S scsi scsibus229
sd6514 sd 1379 0x00000040 S scsi scsibus229
sd6515 sd 1380 0x00000050 S scsi scsibus229
scsibus230 scsibus 230 0x05000070 SM mscsi fcbr128
pbay230 pbay 230 0x00000100 S scsi scsibus230
sd6516 sd 1381 0x00000000 S scsi scsibus230
sd6517 sd 1382 0x00000010 S scsi scsibus230
sd6518 sd 1383 0x00000020 S scsi scsibus230
sd6519 sd 1384 0x00000030 S scsi scsibus230
```

Appendix C – Tunable Parameters

```
sd6520      sd        1385  0x00000040  S      scsi     scsibus230
sd6521      sd        1386  0x00000050  S      scsi     scsibus230
scsibus231  scsibus    231  0x04000070  SM     mscsi    fcbr128
pbay231     pbay       231  0x00000100  S      scsi     scsibus231
sd6522      sd        1387  0x00000000  S      scsi     scsibus231
sd6523      sd        1388  0x00000010  S      scsi     scsibus231
sd6524      sd        1389  0x00000020  S      scsi     scsibus231
sd6525      sd        1390  0x00000030  S      scsi     scsibus231
sd6526      sd        1391  0x00000040  S      scsi     scsibus231
sd6527      sd        1392  0x00000050  S      scsi     scsibus231
sdi0        sdi          0  0x00ffffff  SM     sdi      ff66
qlc4        qlc          4  0x00000002  L      pci      quad2
scsibus252  scsibus    252  0x00000070  L      mscsi    qlc4
pbay252     pbay       252  0x00000100  L      scsi     scsibus252
sd2012      sd        1512  0x00000000  L      scsi     scsibus252
sd2013      sd        1602  0x00000010  L      scsi     scsibus252
sd2014      sd        1603  0x00000020  L      scsi     scsibus252
qlc5        qlc          5  0x00000004  L      pci      quad2
scsibus253  scsibus    253  0x00000070  L      mscsi    qlc5
pbay253     pbay       253  0x00000100  L      scsi     scsibus253
sd2015      sd        1515  0x00000000  L      scsi     scsibus253
sd2016      sd        1516  0x00000010  L      scsi     scsibus253
sd2017      sd        1517  0x00000020  L      scsi     scsibus253
pe5         pe           5  0x00000003  L      pci      quad2
quad3       quad         3  0x00000003  L      sci      sci0
ff68        ff          68  0x00000005  SP     pci      quad3
fabric6     fabric       7  0x00fe5000  SM     fc       ff68
sdi0        sdi          0  0x00ffffff  SM     sdi      ff68
ff15        ff          15  0x00000006  SP     pci      quad3
fabric7     fabric       6  0x00fe5000  SM     fc       ff15
sdi0        sdi          0  0x00ffffff  SM     sdi      ff15
qlc6        qlc          6  0x00000002  L      pci      quad3
scsibus254  scsibus    254  0x00000070  L      mscsi    qlc6
pbay254     pbay       254  0x00000100  L      scsi     scsibus254
sd2018      sd        1518  0x00000000  L      scsi     scsibus254
sd2019      sd        1519  0x00000010  L      scsi     scsibus254
sd2020      sd        1520  0x00000020  L      scsi     scsibus254
qlc7        qlc          7  0x00000004  L      pci      quad3
scsibus255  scsibus    255  0x00000070  L      mscsi    qlc7
pbay255     pbay       255  0x00000100  L      scsi     scsibus255
sd2021      sd        1521  0x00000000  L      scsi     scsibus255
sd2022      sd        1522  0x00000010  L      scsi     scsibus255
sd2023      sd        1523  0x00000020  L      scsi     scsibus255
pe6         pe           6  0x00000003  L      pci      quad3
quad4       quad         4  0x00000004  L      sci      sci0
ff69        ff          69  0x00000000  SP     pci      quad4
fabric0     fabric       0  0x00fe4800  SM     fc       ff69
sdi0        sdi          0  0x00ffffff  SM     sdi      ff69
ff17        ff          17  0x00000001  SP     pci      quad4
fabric1     fabric       1  0x00fe4800  SM     fc       ff17
sdi0        sdi          0  0x00ffffff  SM     sdi      ff17
qlc8        qlc          8  0x00000002  L      pci      quad4
scsibus256  scsibus    256  0x00000070  L      mscsi    qlc8
pbay256     pbay       256  0x00000100  L      scsi     scsibus256
sd2024      sd        1524  0x00000000  L      scsi     scsibus256
sd2025      sd        1525  0x00000010  L      scsi     scsibus256
sd2026      sd        1526  0x00000020  L      scsi     scsibus256
qlc9        qlc          9  0x00000004  L      pci      quad4
scsibus257  scsibus    257  0x00000070  L      mscsi    qlc9
pbay257     pbay       257  0x00000100  L      scsi     scsibus257
sd2027      sd        1527  0x00000000  L      scsi     scsibus257
sd2028      sd        1528  0x00000010  L      scsi     scsibus257
sd2029      sd        1529  0x00000020  L      scsi     scsibus257
pe7         pe           7  0x00000003  L      pci      quad4
quad5       quad         5  0x00000005  L      sci      sci0
ff71        ff          71  0x00000000  SP     pci      quad5
fabric0     fabric       0  0x00fe4000  SM     fc       ff71
sdi0        sdi          0  0x00ffffff  SM     sdi      ff71
ff21        ff          21  0x00000001  SP     pci      quad5
fabric1     fabric       1  0x00fe4000  SM     fc       ff21
sdi0        sdi          0  0x00ffffff  SM     sdi      ff21
qlc10       qlc         10  0x00000002  L      pci      quad5
scsibus258  scsibus    258  0x00000070  L      mscsi    qlc10
pbay258     pbay       258  0x00000100  L      scsi     scsibus258
sd2030      sd        1530  0x00000000  L      scsi     scsibus258
sd2031      sd        1531  0x00000010  L      scsi     scsibus258
sd2032      sd        1532  0x00000020  L      scsi     scsibus258
qlc11       qlc         11  0x00000004  L      pci      quad5
scsibus259  scsibus    259  0x00000070  L      mscsi    qlc11
pbay259     pbay       259  0x00000100  L      scsi     scsibus259
sd2033      sd        1533  0x00000000  L      scsi     scsibus259
sd2034      sd        1534  0x00000010  L      scsi     scsibus259
sd2035      sd        1535  0x00000020  L      scsi     scsibus259
pe8         pe           8  0x00000003  L      pci      quad5
quad6       quad         6  0x00000006  L      sci      sci0
ff26        ff          26  0x00000005  SP     pci      quad6
fabric6     fabric       7  0x00fe4800  SM     fc       ff26
sdi0        sdi          0  0x00ffffff  SM     sdi      ff26
ff27        ff          27  0x00000006  SP     pci      quad6
fabric7     fabric       6  0x00fe4800  SM     fc       ff27
sdi0        sdi          0  0x00ffffff  SM     sdi      ff27
qlc12       qlc         12  0x00000002  L      pci      quad6
scsibus260  scsibus    260  0x00000070  L      mscsi    qlc12
pbay260     pbay       260  0x00000100  L      scsi     scsibus260
sd2036      sd        1536  0x00000000  L      scsi     scsibus260
sd2037      sd        1537  0x00000010  L      scsi     scsibus260
sd2038      sd        1538  0x00000020  L      scsi     scsibus260
qlc13       qlc         13  0x00000004  L      pci      quad6
scsibus261  scsibus    261  0x00000070  L      mscsi    qlc13
pbay261     pbay       261  0x00000100  L      scsi     scsibus261
sd2039      sd        1539  0x00000000  L      scsi     scsibus261
sd2040      sd        1540  0x00000010  L      scsi     scsibus261
sd2041      sd        1541  0x00000020  L      scsi     scsibus261
pe9         pe           9  0x00000003  L      pci      quad6
quad7       quad         7  0x00000007  L      sci      sci0
ff30        ff          30  0x00000005  SP     pci      quad7
fabric6     fabric       7  0x00fe4000  SM     fc       ff30
sdi0        sdi          0  0x00ffffff  SM     sdi      ff30
ff31        ff          31  0x00000006  SP     pci      quad7
fabric7     fabric       6  0x00fe4000  SM     fc       ff31
sdi0        sdi          0  0x00ffffff  SM     sdi      ff31
qlc14       qlc         14  0x00000002  L      pci      quad7
scsibus262  scsibus    262  0x00000070  L      mscsi    qlc14
pbay262     pbay       262  0x00000100  L      scsi     scsibus262
sd2042      sd        1542  0x00000000  L      scsi     scsibus262
sd2043      sd        1543  0x00000010  L      scsi     scsibus262
sd2044      sd        1544  0x00000020  L      scsi     scsibus262
qlc15       qlc         15  0x00000004  L      pci      quad7
scsibus263  scsibus    263  0x00000070  L      mscsi    qlc15
pbay263     pbay       263  0x00000100  L      scsi     scsibus263
sd2045      sd        1545  0x00000000  L      scsi     scsibus263
sd2046      sd        1546  0x00000010  L      scsi     scsibus263
sd2047      sd        1547  0x00000020  L      scsi     scsibus263
pe10        pe          10  0x00000003  L      pci      quad7
quad8       quad         8  0x00000008  L      sci      sci0
ff72        ff          72  0x00000000  SP     pci      quad8
fabric0     fabric       0  0x00fe7800  SM     fc       ff72
sdi0        sdi          0  0x00ffffff  SM     sdi      ff72
ff73        ff          73  0x00000001  SP     pci      quad8
fabric1     fabric       1  0x00fe7800  SM     fc       ff73
sdi0        sdi          0  0x00ffffff  SM     sdi      ff73
qlc16       qlc         16  0x00000002  L      pci      quad8
scsibus264  scsibus    264  0x00000070  L      mscsi    qlc16
pbay264     pbay       264  0x00000100  L      scsi     scsibus264
sd2048      sd        1548  0x00000000  L      scsi     scsibus264
sd2049      sd        1549  0x00000010  L      scsi     scsibus264
sd2050      sd        1550  0x00000020  L      scsi     scsibus264
qlc17       qlc         17  0x00000004  L      pci      quad8
scsibus265  scsibus    265  0x00000070  L      mscsi    qlc17
pbay265     pbay       265  0x00000100  L      scsi     scsibus265
sd2051      sd        1551  0x00000000  L      scsi     scsibus265
sd2052      sd        1552  0x00000010  L      scsi     scsibus265
sd2053      sd        1553  0x00000020  L      scsi     scsibus265
pe11        pe          11  0x00000003  L      pci      quad8
quad9       quad         9  0x00000009  L      sci      sci0
ff75        ff          75  0x00000000  SP     pci      quad9
fabric0     fabric       0  0x00fe7000  SM     fc       ff75
sdi0        sdi          0  0x00ffffff  SM     sdi      ff75
ff76        ff          76  0x00000001  SP     pci      quad9
fabric1     fabric       1  0x00fe7000  SM     fc       ff76
sdi0        sdi          0  0x00ffffff  SM     sdi      ff76
qlc18       qlc         18  0x00000002  L      pci      quad9
scsibus266  scsibus    266  0x00000070  L      mscsi    qlc18
pbay266     pbay       266  0x00000100  L      scsi     scsibus266
sd2054      sd        1554  0x00000000  L      scsi     scsibus266
sd2055      sd        1555  0x00000010  L      scsi     scsibus266
sd2056      sd        1556  0x00000020  L      scsi     scsibus266
qlc19       qlc         19  0x00000004  L      pci      quad9
scsibus267  scsibus    267  0x00000070  L      mscsi    qlc19
pbay267     pbay       267  0x00000100  L      scsi     scsibus267
sd2057      sd        1557  0x00000000  L      scsi     scsibus267
sd2058      sd        1558  0x00000010  L      scsi     scsibus267
sd2059      sd        1559  0x00000020  L      scsi     scsibus267
pe12        pe          12  0x00000003  L      pci      quad9
quad10      quad        10  0x0000000a  L      sci      sci0
ff80        ff          80  0x00000001  SP     pci      quad10
fabric6     fabric       7  0x00fe7800  SM     fc       ff80
sdi0        sdi          0  0x00ffffff  SM     sdi      ff80
ff43        ff          43  0x00000006  SP     pci      quad10
fabric7     fabric       6  0x00fe7800  SM     fc       ff43
sdi0        sdi          0  0x00ffffff  SM     sdi      ff43
qlc20       qlc         20  0x00000002  L      pci      quad10
scsibus268  scsibus    268  0x00000070  L      mscsi    qlc20
pbay268     pbay       268  0x00000100  L      scsi     scsibus268
sd2060      sd        1560  0x00000000  L      scsi     scsibus268
sd2061      sd        1561  0x00000010  L      scsi     scsibus268
sd2062      sd        1562  0x00000020  L      scsi     scsibus268
qlc21       qlc         21  0x00000004  L      pci      quad10
```

```
scsibus269  scsibus    269  0x00000070  L     mscsi    qlc21
pbay269     pbay       269  0x00000100  L     scsi     scsibus269
sd2063      sd        1563  0x00000000  L     scsi     scsibus269
sd2064      sd        1564  0x00000010  L     scsi     scsibus269
sd2065      sd        1565  0x00000020  L     scsi     scsibus269
pe13        pe          13  0x00000003  L     pci      quad10
quad11      quad        11  0x0000000b  L     sci      sci0
ff46        ff          46  0x00000005  SP    pci      quad11
fabric6     fabric       7  0x00fe7000  SM    fc       ff46
sdi0        sdi          0  0x00ffffff  SM    sdi      ff46
ff47        ff          47  0x00000006  SP    pci      quad11
fabric7     fabric       6  0x00fe7000  SM    fc       ff47
sdi0        sdi          0  0x00ffffff  SM    sdi      ff47
qlc22       qlc         22  0x00000002  L     pci      quad11
scsibus270  scsibus    270  0x00000070  L     mscsi    qlc22
pbay270     pbay       270  0x00000100  L     scsi     scsibus270
sd2066      sd        1604  0x00000000  L     scsi     scsibus270
sd2067      sd        1567  0x00000010  L     scsi     scsibus270
sd2068      sd        1568  0x00000020  L     scsi     scsibus270
qlc23       qlc         23  0x00000004  L     pci      quad11
scsibus271  scsibus    271  0x00000070  L     mscsi    qlc23
pbay271     pbay       271  0x00000100  L     scsi     scsibus271
sd2069      sd        1569  0x00000000  L     scsi     scsibus271
sd2070      sd        1570  0x00000010  L     scsi     scsibus271
sd2071      sd        1571  0x00000020  L     scsi     scsibus271
pe14        pe          14  0x00000003  L     pci      quad11
quad12      quad        12  0x0000000c  L     sci      sci0
ff82        ff          82  0x00000000  SP    pci      quad12
fabric0     fabric       0  0x00fe6800  SM    fc       ff82
sdi0        sdi          0  0x00ffffff  SM    sdi      ff82
ff49        ff          49  0x00000001  SP    pci      quad12
fabric1     fabric       1  0x00fe6800  SM    fc       ff49
sdi0        sdi          0  0x00ffffff  SM    sdi      ff49
qlc24       qlc         24  0x00000002  L     pci      quad12
scsibus272  scsibus    272  0x00000070  L     mscsi    qlc24
pbay272     pbay       272  0x00000100  L     scsi     scsibus272
sd2072      sd        1605  0x00000000  L     scsi     scsibus272
sd2073      sd        1573  0x00000010  L     scsi     scsibus272
sd2074      sd        1574  0x00000020  L     scsi     scsibus272
qlc25       qlc         25  0x00000004  L     pci      quad12
scsibus273  scsibus    273  0x00000070  L     mscsi    qlc25
pbay273     pbay       273  0x00000100  L     scsi     scsibus273
sd2075      sd        1606  0x00000000  L     scsi     scsibus273
sd2076      sd        1607  0x00000010  L     scsi     scsibus273
sd2077      sd        1608  0x00000020  L     scsi     scsibus273
pe15        pe          15  0x00000003  L     pci      quad12
quad13      quad        13  0x0000000d  L     sci      sci0
ff84        ff          84  0x00000000  SP    pci      quad13
fabric0     fabric       0  0x00fe6000  SM    fc       ff84
sdi0        sdi          0  0x00ffffff  SM    sdi      ff84
ff85        ff          85  0x00000001  SP    pci      quad13
fabric1     fabric       1  0x00fe6000  SM    fc       ff85
sdi0        sdi          0  0x00ffffff  SM    sdi      ff85
qlc26       qlc         26  0x00000002  L     pci      quad13
scsibus274  scsibus    274  0x00000070  L     mscsi    qlc26
pbay274     pbay       274  0x00000100  L     scsi     scsibus274
sd2078      sd        1578  0x00000000  L     scsi     scsibus274
sd2079      sd        1579  0x00000010  L     scsi     scsibus274
sd2080      sd        1580  0x00000020  L     scsi     scsibus274
qlc27       qlc         27  0x00000004  L     pci      quad13
scsibus275  scsibus    275  0x00000070  L     mscsi    qlc27
pbay275     pbay       275  0x00000100  L     scsi     scsibus275
sd2081      sd        1581  0x00000000  L     scsi     scsibus275
sd2082      sd        1582  0x00000010  L     scsi     scsibus275
sd2083      sd        1583  0x00000020  L     scsi     scsibus275
pe16        pe          16  0x00000003  L     pci      quad13
quad14      quad        14  0x0000000e  L     sci      sci0
ff90        ff          90  0x00000005  SP    pci      quad14
fabric6     fabric       7  0x00fe6800  SM    fc       ff90
sdi0        sdi          0  0x00ffffff  SM    sdi      ff90
ff91        ff          91  0x00000006  SP    pci      quad14
fabric7     fabric       6  0x00fe6800  SM    fc       ff91
sdi0        sdi          0  0x00ffffff  SM    sdi      ff91
qlc28       qlc         28  0x00000002  L     pci      quad14
scsibus276  scsibus    276  0x00000070  L     mscsi    qlc28
pbay276     pbay       276  0x00000100  L     scsi     scsibus276
sd2084      sd        1584  0x00000000  L     scsi     scsibus276
sd2085      sd        1585  0x00000010  L     scsi     scsibus276
sd2086      sd        1586  0x00000020  L     scsi     scsibus276
qlc29       qlc         29  0x00000004  L     pci      quad14
scsibus277  scsibus    277  0x00000070  L     mscsi    qlc29
pbay277     pbay       277  0x00000100  L     scsi     scsibus277
sd2087      sd        1587  0x00000000  L     scsi     scsibus277
sd2088      sd        1588  0x00000010  L     scsi     scsibus277
sd2089      sd        1589  0x00000020  L     scsi     scsibus277
pe17        pe          17  0x00000003  L     pci      quad14
quad15      quad        15  0x0000000f  L     sci      sci0
ff62        ff          62  0x00000005  SP    pci      quad15
fabric6     fabric       7  0x00fe6000  SM    fc       ff62
sdi0        sdi          0  0x00ffffff  SM    sdi      ff62
ff93        ff          93  0x00000006  SP    pci      quad15
fabric7     fabric       6  0x00fe6000  SM    fc       ff93
sdi0        sdi          0  0x00ffffff  SM    sdi      ff93
qlc30       qlc         30  0x00000002  L     pci      quad15
scsibus278  scsibus    278  0x00000070  L     mscsi    qlc30
pbay278     pbay       278  0x00000100  L     scsi     scsibus278
sd2090      sd        1590  0x00000000  L     scsi     scsibus278
sd2091      sd        1591  0x00000010  L     scsi     scsibus278
sd2092      sd        1592  0x00000020  L     scsi     scsibus278
qlc31       qlc         31  0x00000004  L     pci      quad15
scsibus279  scsibus    279  0x00000070  L     mscsi    qlc31
pbay279     pbay       279  0x00000100  L     scsi     scsibus279
sd2093      sd        1593  0x00000000  L     scsi     scsibus279
sd2094      sd        1594  0x00000010  L     scsi     scsibus279
sd2095      sd        1595  0x00000020  L     scsi     scsibus279
pe18        pe          18  0x00000003  L     pci      quad15
usclk       pseudo       1
pmap        pseudo      32
kl          pseudo       -
log         pseudo       5
mem         pseudo       -
gentty      pseudo       -
cmpt        pseudo       -
vtoc        pseudo       -
pty         pseudo     100
ptycompat   pseudo     100
strpipe     pseudo     300
cn          pseudo       -
clone       pseudo       -
ticlts      pseudo       -
ticotsord   pseudo       -
rtc         pseudo       -
vcpt        pseudo       -
fastlantest pseudo       4
bpf         pseudo       8
echo        pseudo       -
llc2        pseudo       -
ted         pseudo       0
vol         pseudo       -
vols        pseudo       -
volobj      pseudo       -
udp         pseudo       -
ip          pseudo       -
tcpmux      pseudo       -
```

Client Configuration Parameters

Microsoft Windows NT Server 4.0 Tunable Parameters

The client configuration parameters were modified as specified below.

Microsoft Windows NT Server Version 4.0 Service Pack 3 Configuration.

No Windows NT parameters were changed on the client machines.
The following services were disabled:

Gopher Publishing Service
License Logging Service
FTP Publishing Serivce
Network DDE
Network DDE DSDM
Net Login
OracleClientCache80
Plug and Play
Directory Replicator
RPC Locator
RPC Service
Schedule
Spooler
Telephony Service
UPS

Appendix C – Tunable Parameters

Microsoft Windows NT Configuration

```
Microsoft Diagnostics Report For \\CLIENT1
------------------------------------------------------------------------

OS Version Report
------------------------------------------------------------------------
Microsoft (R) Windows NT (TM) Server
Version 4.0 (Build 1381: Service Pack 3) x86 Multiprocessor Free
Registered Owner: sqnt, sqnt
Product Number: 70234-811-9559894-61067
------------------------------------------------------------------------

System Report
------------------------------------------------------------------------
System: AT/AT COMPATIBLE
Hardware Abstraction Layer: MPS 1.4 - APIC platform
BIOS Date: 08/15/98
BIOS Version: AC450NX - PhoenixBIOS 4.0 Releas

Processor list:
  0:  x86 Family 6 Model 5 Stepping 2 GenuineIntel ~400 Mhz
  1:  x86 Family 6 Model 5 Stepping 2 GenuineIntel ~400 Mhz
  2:  x86 Family 6 Model 5 Stepping 2 GenuineIntel ~400 Mhz
  3:  x86 Family 6 Model 5 Stepping 2 GenuineIntel ~400 Mhz
------------------------------------------------------------------------

Video Display Report
------------------------------------------------------------------------
BIOS Date: 05/22/96
BIOS Version: CL-GD5436/46 PCI VGA BIOS Version 1.25

Adapter:
  Setting: 1024 x 768 x 256
        70 Hz
  Type: cirrus compatible display adapter
  String: Cirrus Logic Compatible
  Memory: 2 MB
  Chip Type: Cirrus Logic 5446
  DAC Type: Integrated RAMDAC
Driver:
  Vendor: Microsoft Corporation
  File(s): cirrus.sys, vga.dll, cirrus.dll, vga256.dll, vga64K.dll
  Version: 4.00, 4.0.0


Drives Report
------------------------------------------------------------------------
C:\  (Local - NTFS)  Total: 2,096,450KB, Free: 1,845,619KB
  Serial Number: DC78 - BA95
  Bytes per cluster: 512
  Sectors per cluster: 08
  Serial Number: 806D - 3BC4
  Bytes per cluster: 512
  Sectors per cluster: 8
  Filename length: 255
D:\  (Local - NTFS) ARCH Total: 6,787,460KB, Free: 5,733,300KB
  Serial Number: 806D - 3BC4
  Bytes per cluster: 512
  Sectors per cluster: 8
  Filename length: 255
E:\  (CDROM - CDFS) O8EENT804 Total: 644,516KB, Free: 0KB
  Serial Number: E2AA - 797
  Bytes per cluster: 2048
  Sectors per cluster: 1
  Filename length: 221

Memory Report
------------------------------------------------------------------------
Handles: 2,101
Threads: 139
Processes: 18

Physical Memory (K)
  Total: 1,047,976
  Available: 976,024
  File Cache: 12,412

Kernel Memory (K)
  Total: 10,812
  Paged: 5,756
  Nonpaged: 5,056

Commit Charge (K)
  Total: 28,900
  Limit: 2,048,488
  Peak: 44,400

Pagefile Space (K)
  Total: 1,048,576
  Total in use: 5,460
  Peak: 5,792

  D:\pagefile.sys
    Total: 1,048,576
    Total in use: 5,460
    Peak: 5,792


Services Report
------------------------------------------------------------------------
Alerter                          Running   (Automatic)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    LanmanWorkstation
Computer Browser                 Running   (Automatic)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    LanmanWorkstation
    LanmanServer
    LmHosts
ClipBook Server                  Stopped   (Manual)
  C:\WINNT\system32\clipsrv.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process
  Service Dependencies:
    NetDDE
DHCP Client (TDI)                Stopped   (Disabled)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    Tcpip
    Afd
    NetBT
EventLog (Event log)             Running   (Automatic)
  C:\WINNT\system32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
Gopher Publishing Service        Stopped   (Disabled)
  C:\WINNT\System32\inetsrv\inetinfo.exe
  Service Account Name: LocalSystem
  Error Severity: Ignore
  Service Flags: Shared Process
  Service Dependencies:
    RPCSS
    NTLMSSP
Server                           Running   (Automatic)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Group Dependencies:
    TDI
Workstation (NetworkProvider)    Running   (Automatic)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Group Dependencies:
    TDI
License Logging Service          Stopped   (Disabled)
  C:\WINNT\System32\llssrv.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process
TCP/IP NetBIOS Helper            Running   (Automatic)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
```

Appendix C – Tunable Parameters

Group Dependencies:
NetworkProvider
Messenger Running (Automatic)
C:\WINNT\System32\services.exe
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Shared Process
Service Dependencies:
LanmanWorkstation
NetBios
FTP Publishing Service Stopped (Disabled)
C:\WINNT\System32\inetsrv\inetinfo.exe
Service Account Name: LocalSystem
Error Severity: Ignore
Service Flags: Shared Process
Service Dependencies:
RPCSS
NTLMSSP
Network DDE (NetDDEGroup) Stopped (Manual)
C:\WINNT\system32\netdde.exe
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Shared Process
Service Dependencies:
NetDDEDSDM
Network DDE DSDM Stopped (Manual)
C:\WINNT\system32\netdde.exe
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Shared Process
Net Logon (RemoteValidation) Stopped (Manual)
C:\WINNT\System32\lsass.exe
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Shared Process
Service Dependencies:
LanmanWorkstation
LmHosts
NT LM Security Support Provider Running (Manual)
C:\WINNT\System32\SERVICES.EXE
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Shared Process
OracleClientCache80 Stopped (Manual)
C:\orant\BIN\ONRSD80.EXE
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Own Process
Plug and Play (PlugPlay) Running (Automatic)
C:\WINNT\system32\services.exe
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Shared Process
Directory Replicator Stopped (Manual)
C:\WINNT\System32\lmrepl.exe
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Own Process
Service Dependencies:
LanmanWorkstation
LanmanServer
Remote Procedure Call (RPC) Locator Stopped (Manual)
C:\WINNT\System32\LOCATOR.EXE
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Own Process
Service Dependencies:
LanmanWorkstation
Rdr
Remote Procedure Call (RPC) Service Running (Automatic)
C:\WINNT\system32\RpcSs.exe
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Own Process
Schedule Stopped (Manual)
C:\WINNT\System32\AtSvc.Exe
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Own Process
Spooler (SpoolerGroup) Stopped (Disabled)
C:\WINNT\system32\spoolss.exe
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Own Process, Interactive
Telephony Service Stopped (Manual)
C:\WINNT\system32\tapisrv.exe
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Own Process
TUXEDO IPC HELPER Running (Automatic)
c:\TUXEDO\bin\tuxipc.exe
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Own Process
UPS Stopped (Manual)
C:\WINNT\System32\ups.exe
Service Account Name: LocalSystem
Error Severity: Normal
Service Flags: Own Process
W3SVC Running (Manual)
c:\winnt\system32\inetsrv\inetinfo.exe
Service Account Name: LocalSystem
Error Severity: Ignore
Service Flags: Own Process
World Wide Web Publishing Service Running (Manual)
C:\WINNT\System32\inetsrv\inetinfo.exe
Service Account Name: LocalSystem
Error Severity: Ignore
Service Flags: Shared Process
Service Dependencies:
RPCSS
NTLMSSP

Drivers Report

Abiosdsk (Primary disk) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
AFD Networking Support Environment (TDI) Running (Automatic)
C:\WINNT\System32\drivers\afd.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Aha154x (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Aha174x (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
aic78xx (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Always (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
ami0nt (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
amsint (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Arrow (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
atapi (SCSI miniport) Running (Boot)
C:\WINNT\System32\DRIVERS\atapi.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Atdisk (Primary disk) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
ati (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Beep (Base) Running (System)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
BusLogic (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Busmouse (Pointer Port) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Cdaudio (Filter) Stopped (System)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Cdfs (File system) Running (Disabled)
Error Severity: Normal
Service Flags: File System Driver, Shared Process
Group Dependencies:
SCSI CDROM Class
Cdrom (SCSI CDROM Class) Running (System)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Group Dependencies:
SCSI miniport
Changer (Filter) Stopped (System)

Appendix C – Tunable Parameters

Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
cirrus (Video) Running (System)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Cpqarray (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
cpqfws2e (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
dac960nt (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
DEC DC21X4 Adapter Driver (NDIS) Running (Automatic)
C:\WINNT\System32\drivers\DC21X4.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
dce376nt (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Delldsa (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Dell_DGX (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Disk (SCSI Class) Running (Boot)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Group Dependencies:
SCSI miniport
Diskperf (Filter) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
DptScsi (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
dtc329x (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
et4000 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Fastfat (Boot file system) Running (Disabled)
Error Severity: Normal
Service Flags: File System Driver, Shared Process
Fd16_700 (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Fd7000ex (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Fd8xx (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
flashpnt (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Floppy (Primary disk) Running (System)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Ftdisk (Filter) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
i8042 Keyboard and PS/2 Mouse Port Driver (Keyboard Port) Running (System)
System32\DRIVERS\i8042prt.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Inport (Pointer Port) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Jazzg300 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Jazzg364 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Jzvxl484 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Keyboard Class Driver (Keyboard Class) Running (System)
System32\DRIVERS\kbdclass.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
KSecDD (Base) Running (System)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
mga (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
mga_mil (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
mitsumi (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
mkecr5xx (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Modem (Extended base) Stopped (Manual)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Mouse Class Driver (Pointer Class) Running (System)
System32\DRIVERS\mouclass.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Msfs (File system) Running (System)
Error Severity: Normal
Service Flags: File System Driver, Shared Process
Mup (Network) Running (Manual)
C:\WINNT\System32\drivers\mup.sys
Error Severity: Normal
Service Flags: File System Driver, Shared Process
Ncr53c9x (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
ncr77c22 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Ncrc700 (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Ncrc710 (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Microsoft NDIS System Driver (NDIS) Running (System)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
NetBIOS Interface (NetBIOSGroup) Running (Manual)
C:\WINNT\System32\drivers\netbios.sys
Error Severity: Normal
Service Flags: File System Driver, Shared Process
Group Dependencies:
TDI
WINS Client(TCP/IP) (PNP_TDI) Running (Automatic)
C:\WINNT\System32\drivers\netbt.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Service Dependencies:
Tcpip
NetDetect Stopped (Manual)
C:\WINNT\system32\drivers\netdtect.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Npfs (File system) Running (System)
Error Severity: Normal
Service Flags: File System Driver, Shared Process
Ntfs (File system) Running (Disabled)
Error Severity: Normal
Service Flags: File System Driver, Shared Process
Null (Base) Running (System)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Oliscsi (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Parallel (Extended base) Running (Automatic)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Service Dependencies:
Parport
Group Dependencies:
Parallel arbitrator
Parport (Parallel arbitrator) Running (Automatic)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
ParVdm (Extended base) Running (Automatic)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Service Dependencies:
Parport
Group Dependencies:
Parallel arbitrator
PCIDump (PCI Configuration) Stopped (System)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Pcmcia (System Bus Extender) Stopped (Disabled)

Appendix C – Tunable Parameters

```
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
PnP ISA Enabler Driver (Base)              Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
psidisp (Video)                            Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Ql10wnt (SCSI miniport)                    Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
qv (Video)                                 Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Rdr (Network)                              Running   (Manual)
   C:\WINNT\System32\drivers\rdr.sys
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
s3 (Video)                                 Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Scsiprnt (Extended base)                   Stopped   (Automatic)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
     SCSI miniport
Scsiscan (SCSI Class)                      Running   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
     SCSI miniport
Serial (Extended base)                     Running   (Automatic)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Sermouse (Pointer Port)                    Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Sfloppy (Primary disk)                     Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
     SCSI miniport
Simbad (Filter)                            Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
slcd32 (SCSI miniport)                     Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Sparrow (SCSI miniport)                    Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Spock (SCSI miniport)                      Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Srv (Network)                              Running   (Manual)
   C:\WINNT\System32\drivers\srv.sys
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
symc810 (SCSI miniport)                    Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
sym_hi (SCSI miniport)                     Running   (Boot)
   C:\WINNT\system32\drivers\sym_hi.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
T128 (SCSI miniport)                       Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
T13B (SCSI miniport)                       Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
TCP/IP Service (PNP_TDI)                   Running   (Automatic)
   C:\WINNT\System32\drivers\tcpip.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
tga (Video)                                Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
tmv1 (SCSI miniport)                       Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Ultra124 (SCSI miniport)                   Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Ultra14f (SCSI miniport)                   Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Ultra24f (SCSI miniport)                   Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
v7vram (Video)                             Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
VgaSave (Video Save)                       Stopped   (System)
   C:\WINNT\System32\drivers\vga.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
VgaStart (Video Init)                      Stopped   (System)
   C:\WINNT\System32\drivers\vga.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Wd33c93 (SCSI miniport)                    Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
wd90c24a (Video)                           Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
wdvga (Video)                              Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
weitekp9 (Video)                           Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Xga (Video)                                Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
```

IRQ and Port Report

Devices	Vector	Level	Affinity
MPS 1.4 - APIC platform	8	8	0x0000000f
MPS 1.4 - APIC platform	0	0	0x0000000f
MPS 1.4 - APIC platform	1	1	0x0000000f
MPS 1.4 - APIC platform	2	2	0x0000000f
MPS 1.4 - APIC platform	3	3	0x0000000f
MPS 1.4 - APIC platform	4	4	0x0000000f
MPS 1.4 - APIC platform	5	5	0x0000000f
MPS 1.4 - APIC platform	6	6	0x0000000f
MPS 1.4 - APIC platform	7	7	0x0000000f
MPS 1.4 - APIC platform	8	8	0x0000000f
MPS 1.4 - APIC platform	9	9	0x0000000f
MPS 1.4 - APIC platform	10	10	0x0000000f
MPS 1.4 - APIC platform	11	11	0x0000000f
MPS 1.4 - APIC platform	12	12	0x0000000f
MPS 1.4 - APIC platform	13	13	0x0000000f
MPS 1.4 - APIC platform	14	14	0x0000000f
MPS 1.4 - APIC platform	15	15	0x0000000f
MPS 1.4 - APIC platform	16	16	0x0000000f
MPS 1.4 - APIC platform	17	17	0x0000000f
MPS 1.4 - APIC platform	18	18	0x0000000f
MPS 1.4 - APIC platform	19	19	0x0000000f
MPS 1.4 - APIC platform	20	20	0x0000000f
MPS 1.4 - APIC platform	21	21	0x0000000f
MPS 1.4 - APIC platform	22	22	0x0000000f
MPS 1.4 - APIC platform	23	23	0x0000000f
MPS 1.4 - APIC platform	24	24	0x0000000f
MPS 1.4 - APIC platform	25	25	0x0000000f
MPS 1.4 - APIC platform	26	26	0x0000000f
MPS 1.4 - APIC platform	27	27	0x0000000f
MPS 1.4 - APIC platform	28	28	0x0000000f
MPS 1.4 - APIC platform	29	29	0x0000000f
MPS 1.4 - APIC platform	30	30	0x0000000f
MPS 1.4 - APIC platform	31	31	0x0000000f
MPS 1.4 - APIC platform	32	32	0x0000000f
MPS 1.4 - APIC platform	33	33	0x0000000f
MPS 1.4 - APIC platform	34	34	0x0000000f
MPS 1.4 - APIC platform	35	35	0x0000000f
MPS 1.4 - APIC platform	36	36	0x0000000f
MPS 1.4 - APIC platform	37	37	0x0000000f
MPS 1.4 - APIC platform	38	38	0x0000000f
MPS 1.4 - APIC platform	39	39	0x0000000f
MPS 1.4 - APIC platform	40	40	0x0000000f
MPS 1.4 - APIC platform	41	41	0x0000000f
MPS 1.4 - APIC platform	42	42	0x0000000f
MPS 1.4 - APIC platform	43	43	0x0000000f
MPS 1.4 - APIC platform	44	44	0x0000000f
MPS 1.4 - APIC platform	45	45	0x0000000f
MPS 1.4 - APIC platform	46	46	0x0000000f
MPS 1.4 - APIC platform	47	47	0x0000000f
MPS 1.4 - APIC platform	61	61	0x0000000f
MPS 1.4 - APIC platform	65	65	0x0000000f
MPS 1.4 - APIC platform	80	80	0x0000000f
MPS 1.4 - APIC platform	193	193	0x0000000f
MPS 1.4 - APIC platform	225	225	0x0000000f
MPS 1.4 - APIC platform	253	253	0x0000000f

Appendix C – Tunable Parameters

```
MPS 1.4 - APIC platform        254   254 0x0000000f
MPS 1.4 - APIC platform        255   255 0x0000000f
i8042prt                         1     1 0xffffffff
i8042prt                        12    12 0xffffffff
Serial                           4     4 0x00000000
Serial                           3     3 0x00000000
DC21X4                          20    20 0x00000000
DC21X4                          20    20 0x00000000
DC21X4                          24    24 0x00000000
DC21X4                          28    28 0x00000000
DC21X4                          16    16 0x00000000
DC21X4                          20    20 0x00000000
DC21X4                          24    24 0x00000000
DC21X4                          28    28 0x00000000
DC21X4                          16    16 0x00000000
DC21X4                          24    24 0x00000000
DC21X4                          28    28 0x00000000
DC21X4                          16    16 0x00000000
DC21X4                          20    20 0x00000000
DC21X4                          24    24 0x00000000
DC21X4                          28    28 0x00000000
DC21X4                          16    16 0x00000000
Floppy                           6     6 0x00000000
atapi                            0    14 0x00000000
sym_hi                          40    40 0x00000000
sym_hi                          41    41 0x00000000
-------------------------------------------------------------------------
Devices                    Physical Address  Length
-------------------------------------------------------------------------
MPS 1.4 - APIC platform        0x00000000  0x0000000010
MPS 1.4 - APIC platform        0x00000020  0x0000000002
MPS 1.4 - APIC platform        0x00000040  0x0000000004
MPS 1.4 - APIC platform        0x00000048  0x0000000004
MPS 1.4 - APIC platform        0x00000061  0x0000000001
MPS 1.4 - APIC platform        0x00000070  0x0000000002
MPS 1.4 - APIC platform        0x00000080  0x0000000010
MPS 1.4 - APIC platform        0x00000092  0x0000000001
MPS 1.4 - APIC platform        0x000000a0  0x0000000002
MPS 1.4 - APIC platform        0x000000c0  0x0000000010
MPS 1.4 - APIC platform        0x000000f0  0x0000000010
i8042prt                    0x00000060  0x0000000001
i8042prt                    0x00000064  0x0000000001
Parport                     0x00000378  0x0000000003
Serial                      0x000003f8  0x0000000007
Serial                      0x000002f8  0x0000000007
DC21X4                         0x00003080  0x0000000080
DC21X4                         0x00005080  0x0000000080
DC21X4                         0x00005400  0x0000000080
DC21X4                         0x00005480  0x0000000080
DC21X4                         0x00006000  0x0000000080
DC21X4                         0x00006080  0x0000000080
DC21X4                         0x00006400  0x0000000080
DC21X4                         0x00006480  0x0000000080
DC21X4                         0x00003000  0x0000000080
DC21X4                         0x00003400  0x0000000080
DC21X4                         0x00003480  0x0000000080
DC21X4                         0x00004000  0x0000000080
DC21X4                         0x00004080  0x0000000080
DC21X4                         0x00004400  0x0000000080
DC21X4                         0x00004480  0x0000000080
DC21X4                         0x00005000  0x0000000080
Floppy                      0x000003f0  0x0000000006
Floppy                      0x000003f7  0x0000000001
atapi                       0x000001f0  0x0000000008
atapi                       0x000003f6  0x0000000001
sym_hi                         0x00002000  0x0000000100
sym_hi                         0x00002400  0x0000000100
cirrus                      0x000003b0  0x000000000c
cirrus                      0x000003c0  0x0000000020
```

DMA and Memory Report

```
-------------------------------------------------------------------------
Devices                    Channel    Port
-------------------------------------------------------------------------
Floppy                           2     0
-------------------------------------------------------------------------
Devices                    Physical Address  Length
-------------------------------------------------------------------------
MPS 1.4 - APIC platform        0xfec10000  0x00000400
MPS 1.4 - APIC platform        0xfee00000  0x00000400
sym_hi                      0xfc004000  0x00000400
sym_hi                      0xfc000000  0x00002000
sym_hi                      0xfc004400  0x00000400
sym_hi                      0xfc002000  0x00002000
cirrus                      0x000a0000  0x00020000
cirrus                      0xfd000000  0x01000000
```

Environment Report

```
-------------------------------------------------------------------------
```

System Environment Variables

```
  ComSpec=C:\WINNT\system32\cmd.exe
  NUMBER_OF_PROCESSORS=4
  OS=Windows_NT
  Os2LibPath=C:\WINNT\system32\os2\dll;
  Path=C:\orant\bin;C:\WINNT\system32;C:\WINNT;c:\tuxedo\bin
  PROCESSOR_ARCHITECTURE=x86
  PROCESSOR_IDENTIFIER=x86 Family 6 Model 5 Stepping 2, GenuineIntel
  PROCESSOR_LEVEL=6
  PROCESSOR_REVISION=0502
  TMCONTEXTS=1
  TUXCONFIG=c:\tpcc\tuxconfig
  TUXDIR=c:\tuxedo
  windir=C:\WINNT
```

Environment Variables for Current User

```
  TEMP=C:\TEMP
  TMP=C:\TEMP
```

Network Report

```
-------------------------------------------------------------------------
Your Access Level: Admin & Local
Workgroup or Domain: TPCC
Network Version: 4.0
LanRoot: TPCC
Logged On Users: 1
Current User (1): Administrator
  Logon Domain: CLIENT1
  Logon Server: CLIENT1

Transport: NetBT_DC21X42, 00-00-BC-11-A8-32, VC's: 0, Wan: Wan
Transport: NetBT_DC21X410, 00-00-BC-11-A6-CE, VC's: 0, Wan: Wan
Transport: NetBT_DC21X411, 00-00-BC-11-A6-CD, VC's: 0, Wan: Wan
Transport: NetBT_DC21X412, 00-00-BC-11-A6-CC, VC's: 0, Wan: Wan
Transport: NetBT_DC21X413, 00-00-BC-11-A5-0F, VC's: 0, Wan: Wan
Transport: NetBT_DC21X414, 00-00-BC-11-A5-0E, VC's: 0, Wan: Wan
Transport: NetBT_DC21X415, 00-00-BC-11-A5-0D, VC's: 0, Wan: Wan
Transport: NetBT_DC21X416, 00-00-BC-11-A5-0C, VC's: 0, Wan: Wan
Transport: NetBT_DC21X41, 00-00-BC-11-A8-33, VC's: 0, Wan: Wan
Transport: NetBT_DC21X43, 00-00-BC-11-A8-31, VC's: 0, Wan: Wan
Transport: NetBT_DC21X44, 00-00-BC-11-A8-30, VC's: 0, Wan: Wan
Transport: NetBT_DC21X45, 00-00-BC-11-A5-13, VC's: 0, Wan: Wan
Transport: NetBT_DC21X46, 00-00-BC-11-A5-12, VC's: 0, Wan: Wan
Transport: NetBT_DC21X47, 00-00-BC-11-A5-11, VC's: 0, Wan: Wan
Transport: NetBT_DC21X48, 00-00-BC-11-A5-10, VC's: 0, Wan: Wan
Transport: NetBT_DC21X49, 00-00-BC-11-A6-CF, VC's: 0, Wan: Wan

Character Wait: 3,600
Collection Time: 250
Maximum Collection Count: 16
Keep Connection: 600
Maximum Commands: 5
Session Time Out: 45
Character Buffer Size: 512
Maximum Threads: 17
Lock Quota: 6,144
Lock Increment: 10
Maximum Locks: 500
Pipe Increment: 10
Maximum Pipes: 500
Cache Time Out: 40
Dormant File Limit: 45
Read Ahead Throughput: 4,294,967,295
Mailslot Buffers: 3
Server Announce Buffers: 20
Illegal Datagrams: 5
Datagram Reset Frequency: 60
Log Election Packets: False
Use Opportunistic Locking: True
Use Unlock Behind: True
Use Close Behind: True
Buffer Pipes: True
Use Lock, Read, Unlock: True
Use NT Caching: True
Use Raw Read: True
Use Raw Write: True
Use Write Raw Data: True
Use Encryption: True
```

Buffer Deny Write Files: True
Buffer Read Only Files: True
Force Core Creation: True
512 Byte Max Transfer: False
Bytes Received: 69,360
SMB's Received: 816
Paged Read Bytes Requested: 0
Non Paged Read Bytes Requested: 0
Cache Read Bytes Requested: 0
Network Read Bytes Requested: 0
Bytes Transmitted: 88,672
SMB's Transmitted: 816
Paged Read Bytes Requested: 0
Non Paged Read Bytes Requested: 82,824
Cache Read Bytes Requested: 0
Network Read Bytes Requested: 0
Initally Failed Operations: 0
Failed Completion Operations: 0
Read Operations: 0
Random Read Operations: 0
Read SMB's: 0
Large Read SMB's: 0
Small Read SMB's: 0
Write Operations: 1,428
Random Write Operations: 0
Write SMB's: 0
Large Write SMB's: 0
Small Write SMB's: 0
Raw Reads Denied: 0
Raw Writes Denied: 0
Network Errors: 0
Sessions: 136
Failed Sessions: 0
Reconnects: 0
Core Connects: 0
LM 2.0 Connects: 0
LM 2.x Connects: 0
Windows NT Connects: 136
Server Disconnects: 0
Hung Sessions: 0
Use Count: 392
Failed Use Count: 120
Current Commands: 0
Server File Opens: 0
Server Device Opens: 0
Server Jobs Queued: 0
Server Session Opens: 0
Server Sessions Timed Out: 0
Server Sessions Errored Out: 0
Server Password Errors: 0
Server Permission Errors: 0
Server System Errors: 0
Server Bytes Sent: 0
Server Bytes Received: 0
Server Average Response Time: 0
Server Request Buffers Needed: 0
Server Big Buffers Needed: 0

Microsoft Internet Information Server Registry Parameters

Key Name: SYSTEM\CurrentControlSet\Services\InetInfo\Parameters
Class Name: <NO CLASS>
Last Write Time: 9/27/98 - 10:46 PM
Value 0
Name: BandwidthLevel
Type: REG_DWORD
Data: 0xffffffff

Value 1
Name: ListenBackLog
Type: REG_DWORD
Data: 0x19

Value 2
Name: MaxPoolThreads
Type: REG_DWORD
Data: 0x1

Value 3
Name: PoolThreadLimit
Type: REG_DWORD
Data: 0x64

Key Name: SYSTEM\CurrentControlSet\Services\InetInfo\Parameters\Filter
Class Name: <NO CLASS>
Last Write Time: 9/9/98 - 7:16 AM
Value 0
Name: FilterType
Type: REG_DWORD
Data: 0

Value 1
Name: NumDenySites
Type: REG_DWORD
Data: 0

Value 2
Name: NumGrantSites
Type: REG_DWORD
Data: 0

Key Name:
SYSTEM\CurrentControlSet\Services\InetInfo\Parameters\MimeMap
Class Name: <NO CLASS>
Last Write Time: 9/9/98 - 7:17 AM
Value 0
Name: application/envoy,evy,,5
Type: REG_SZ
Data:

Value 1
Name: application/mac-binhex40,hqx,,4
Type: REG_SZ
Data:

Value 2
Name: application/msword,doc,,5
Type: REG_SZ
Data:

Value 3
Name: application/msword,dot,,5
Type: REG_SZ
Data:

Value 4
Name: application/octet-stream,*,,5
Type: REG_SZ
Data:

Value 5
Name: application/octet-stream,bin,,5
Type: REG_SZ
Data:

Value 6
Name: application/octet-stream,exe,,5
Type: REG_SZ
Data:

Value 7
Name: application/oda,oda,,5
Type: REG_SZ
Data:

Value 8
Name: application/pdf,pdf,,5
Type: REG_SZ
Data:

Value 9
Name: application/postscript,ai,,5
Type: REG_SZ
Data:

Value 10
Name: application/postscript,eps,,5
Type: REG_SZ
Data:

Value 11
Name: application/postscript,ps,,5
Type: REG_SZ
Data:

Value 12
Name: application/rtf,rtf,,5
Type: REG_SZ
Data:

Value 13
Name: application/winhlp,hlp,,5

Type: REG_SZ
Data:

Value 14
Name: application/x-bcpio,bcpio,,5
Type: REG_SZ
Data:

Value 15
Name: application/x-cpio,cpio,,5
Type: REG_SZ
Data:

Value 16
Name: application/x-csh,csh,,5
Type: REG_SZ
Data:

Value 17
Name: application/x-director,dcr,,5
Type: REG_SZ
Data:

Value 18
Name: application/x-director,dir,,5
Type: REG_SZ
Data:

Value 19
Name: application/x-director,dxr,,5
Type: REG_SZ
Data:

Value 20
Name: application/x-dvi,dvi,,5
Type: REG_SZ
Data:

Value 21
Name: application/x-gtar,gtar,,9
Type: REG_SZ
Data:

Value 22
Name: application/x-hdf,hdf,,5
Type: REG_SZ
Data:

Value 23
Name: application/x-latex,latex,,5
Type: REG_SZ
Data:

Value 24
Name: application/x-msaccess,mdb,,5
Type: REG_SZ
Data:

Value 25
Name: application/x-mscardfile,crd,,5
Type: REG_SZ
Data:

Value 26
Name: application/x-msclip,clp,,5
Type: REG_SZ
Data:

Value 27
Name: application/x-msexcel,xla,,5
Type: REG_SZ
Data:

Value 28
Name: application/x-msexcel,xlc,,5
Type: REG_SZ
Data:

Value 29
Name: application/x-msexcel,xlm,,5
Type: REG_SZ
Data:

Value 30
Name: application/x-msexcel,xls,,5
Type: REG_SZ
Data:

Value 31
Name: application/x-msexcel,xlt,,5
Type: REG_SZ
Data:

Value 32
Name: application/x-msexcel,xlw,,5
Type: REG_SZ
Data:

Value 33
Name: application/x-msmediaview,m13,,5
Type: REG_SZ
Data:

Value 34
Name: application/x-msmediaview,m14,,5
Type: REG_SZ
Data:

Value 35
Name: application/x-msmetafile,wmf,,5
Type: REG_SZ
Data:

Value 36
Name: application/x-msmoney,mny,,5
Type: REG_SZ
Data:

Value 37
Name: application/x-mspowerpoint,ppt,,5
Type: REG_SZ
Data:

Value 38
Name: application/x-msproject,mpp,,5
Type: REG_SZ
Data:

Value 39
Name: application/x-mspublisher,pub,,5
Type: REG_SZ
Data:

Value 40
Name: application/x-msterminal,trm,,5
Type: REG_SZ
Data:

Value 41
Name: application/x-msworks,wks,,5
Type: REG_SZ
Data:

Value 42
Name: application/x-mswrite,wri,,5
Type: REG_SZ
Data:

Value 43
Name: application/x-netcdf,cdf,,5
Type: REG_SZ
Data:

Value 44
Name: application/x-netcdf,nc,,5
Type: REG_SZ
Data:

Value 45
Name: application/x-perfmon,pma,,5
Type: REG_SZ
Data:

Value 46
Name: application/x-perfmon,pmc,,5
Type: REG_SZ
Data:

Value 47
Name: application/x-perfmon,pml,,5
Type: REG_SZ
Data:

Value 48
Name: application/x-perfmon,pmr,,5
Type: REG_SZ
Data:

Value 49
Name: application/x-perfmon,pmw,,5
Type: REG_SZ
Data:

Value 50
Name: application/x-sh,sh,,5
Type: REG_SZ
Data:

Value 51
Name: application/x-shar,shar,,5
Type: REG_SZ
Data:

Value 52
Name: application/x-sv4cpio,sv4cpio,,5
Type: REG_SZ
Data:

Value 53
Name: application/x-sv4crc,sv4crc,,5
Type: REG_SZ
Data:

Value 54
Name: application/x-tar,tar,,5
Type: REG_SZ
Data:

Value 55
Name: application/x-tcl,tcl,,5
Type: REG_SZ
Data:

Value 56
Name: application/x-tex,tex,,5
Type: REG_SZ
Data:

Value 57
Name: application/x-texinfo,texi,,5
Type: REG_SZ
Data:

Value 58
Name: application/x-texinfo,texinfo,,5
Type: REG_SZ
Data:

Value 59
Name: application/x-troff,roff,,5
Type: REG_SZ
Data:

Value 60
Name: application/x-troff,t,,5
Type: REG_SZ
Data:

Value 61
Name: application/x-troff,tr,,5
Type: REG_SZ
Data:

Value 62
Name: application/x-troff-man,man,,5
Type: REG_SZ
Data:

Value 63
Name: application/x-troff-me,me,,5
Type: REG_SZ
Data:

Value 64
Name: application/x-troff-ms,ms,,5
Type: REG_SZ
Data:

Value 65
Name: application/x-ustar,ustar,,5
Type: REG_SZ
Data:

Value 66
Name: application/x-wais-source,src,,7
Type: REG_SZ
Data:

Value 67
Name: application/zip,zip,,9
Type: REG_SZ
Data:

Value 68
Name: audio/basic,au,,<
Type: REG_SZ
Data:

Value 69
Name: audio/basic,snd,,<
Type: REG_SZ
Data:

Value 70
Name: audio/x-aiff,aif,,<
Type: REG_SZ
Data:

Value 71
Name: audio/x-aiff,aifc,,<
Type: REG_SZ
Data:

Value 72
Name: audio/x-aiff,aiff,,<
Type: REG_SZ
Data:

Value 73
Name: audio/x-pn-realaudio,ram,,<
Type: REG_SZ
Data:

Value 74
Name: audio/x-wav,wav,,<
Type: REG_SZ
Data:

Value 75
Name: image/bmp,bmp,,:
Type: REG_SZ
Data:

Value 76
Name: image/cis-cod,cod,,5
Type: REG_SZ
Data:

Value 77
Name: image/gif,gif,,g
Type: REG_SZ
Data:

Value 78
Name: image/ief,ief,,:
Type: REG_SZ
Data:

Value 79
Name: image/jpeg,jpe,,:
Type: REG_SZ
Data:

Value 80
Name: image/jpeg,jpeg,,:
Type: REG_SZ
Data:

Value 81
Name: image/jpeg,jpg,,:
Type: REG_SZ
Data:

Value 82
Name: image/tiff,tif,,:
Type: REG_SZ
Data:

Value 83
Name: image/tiff,tiff,,:
Type: REG_SZ
Data:

Value 84
Name: image/x-cmu-raster,ras,,:
Type: REG_SZ

Data:

Value 85
Name: image/x-cmx,cmx,,5
Type: REG_SZ
Data:

Value 86
Name: image/x-portable-anymap,pnm,,:
Type: REG_SZ
Data:

Value 87
Name: image/x-portable-bitmap,pbm,,:
Type: REG_SZ
Data:

Value 88
Name: image/x-portable-graymap,pgm,,:
Type: REG_SZ
Data:

Value 89
Name: image/x-portable-pixmap,ppm,,:
Type: REG_SZ
Data:

Value 90
Name: image/x-rgb,rgb,,:
Type: REG_SZ
Data:

Value 91
Name: image/x-xbitmap,xbm,,:
Type: REG_SZ
Data:

Value 92
Name: image/x-xpixmap,xpm,,:
Type: REG_SZ
Data:

Value 93
Name: image/x-xwindowdump,xwd,,:
Type: REG_SZ
Data:

Value 94
Name: text/html,htm,,h
Type: REG_SZ
Data:

Value 95
Name: text/html,html,,h
Type: REG_SZ
Data:

Value 96
Name: text/html,stm,,h
Type: REG_SZ
Data:

Value 97
Name: text/plain,bas,,0
Type: REG_SZ
Data:

Value 98
Name: text/plain,c,,0
Type: REG_SZ
Data:

Value 99
Name: text/plain,h,,0
Type: REG_SZ
Data:

Value 100
Name: text/plain,txt,,0
Type: REG_SZ
Data:

Value 101
Name: text/richtext,rtx,,0
Type: REG_SZ
Data:

Value 102
Name: text/tab-separated-values,tsv,,0
Type: REG_SZ
Data:

Value 103
Name: text/x-setext,etx,,0
Type: REG_SZ
Data:

Value 104
Name: video/mpeg,mpe,,;
Type: REG_SZ
Data:

Value 105
Name: video/mpeg,mpeg,,;
Type: REG_SZ
Data:

Value 106
Name: video/mpeg,mpg,,;
Type: REG_SZ
Data:

Value 107
Name: video/quicktime,mov,,;
Type: REG_SZ
Data:

Value 108
Name: video/quicktime,qt,,;
Type: REG_SZ
Data:

Value 109
Name: video/x-msvideo,avi,,<
Type: REG_SZ
Data:

Value 110
Name: video/x-sgi-movie,movie,,<
Type: REG_SZ
Data:

Value 111
Name: x-world/x-vrml,flr,,5
Type: REG_SZ
Data:

Value 112
Name: x-world/x-vrml,wrl,,5
Type: REG_SZ
Data:

Value 113
Name: x-world/x-vrml,wrz,,5
Type: REG_SZ
Data:

Value 114
Name: x-world/x-vrml,xaf,,5
Type: REG_SZ
Data:

Value 115
Name: x-world/x-vrml,xof,,5
Type: REG_SZ
Data:

World Wide Web Service Registry Parameters

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters
Class Name: <NO CLASS>
Last Write Time: 9/25/98 - 12:09 AM
Value 0
Name: AccessDeniedMessage
Type: REG_SZ
Data: Error: Access is Denied.

Value 1
Name: AdminEmail
Type: REG_SZ
Data: Admin@corp.com

Value 2

Name: AdminName
Type: REG_SZ
Data: Administrator

Value 3
Name: AnonymousUserName
Type: REG_SZ
Data: IUSR_CLIENT

Value 4
Name: Authorization
Type: REG_DWORD
Data: 0x5

Value 5
Name: CacheExtensions
Type: REG_DWORD
Data: 0x1

Value 6
Name: CheckForWAISDB
Type: REG_DWORD
Data: 0

Value 7
Name: ConnectionTimeOut
Type: REG_DWORD
Data: 0x7fff

Value 8
Name: DebugFlags
Type: REG_DWORD
Data: 0x8

Value 9
Name: Default Load File
Type: REG_SZ
Data: Default.htm

Value 10
Name: Dir Browse Control
Type: REG_DWORD
Data: 0x4000001e

Value 11
Name: Filter DLLs
Type: REG_SZ
Data: C:\WINNT\System32\inetsrv\sspifilt.dll

Value 12
Name: GlobalExpire
Type: REG_DWORD
Data: 0xffffffff

Value 13
Name: InstallPath
Type: REG_SZ
Data: C:\WINNT\System32\inetsrv

Value 14
Name: LogFileDirectory
Type: REG_EXPAND_SZ
Data: %SystemRoot%\System32\LogFiles

Value 15
Name: LogFileFormat
Type: REG_DWORD
Data: 0

Value 16
Name: LogFilePeriod
Type: REG_DWORD
Data: 0x1

Value 17
Name: LogFileTruncateSize
Type: REG_DWORD
Data: 0x1388000

Value 18
Name: LogSqlDataSource
Type: REG_SZ
Data: HTTPLOG

Value 19
Name: LogSqlPassword
Type: REG_SZ
Data: sqllog

Value 20
Name: LogSqlTableName
Type: REG_SZ
Data: Internetlog

Value 21
Name: LogSqlUserName
Type: REG_SZ
Data: InternetAdmin

Value 22
Name: LogType
Type: REG_DWORD
Data: 0

Value 23
Name: MajorVersion
Type: REG_DWORD
Data: 0x2

Value 24
Name: MaxConnections
Type: REG_DWORD
Data: 0x186a0

Value 25
Name: MinorVersion
Type: REG_DWORD
Data: 0

Value 26
Name: NTAuthenticationProviders
Type: REG_SZ
Data: NTLM

Value 27
Name: ScriptTimeout
Type: REG_DWORD
Data: 0x384

Value 28
Name: SecurePort
Type: REG_DWORD
Data: 0x1bb

Value 29
Name: ServerComment
Type: REG_SZ
Data:

Value 30
Name: ServerSideIncludesEnabled
Type: REG_DWORD
Data: 0x1

Value 31
Name: ServerSideIncludesExtension
Type: REG_SZ
Data: .stm

Key Name:
SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map
Class Name: <NO CLASS>
Last Write Time: 9/9/98 - 7:17 AM
Value 0
Name: .idc
Type: REG_SZ
Data: C:\WINNT\System32\inetsrv\httpodbc.dll

Key Name:
SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots
Class Name: <NO CLASS>
Last Write Time: 9/9/98 - 7:17 AM
Value 0
Name: /
Type: REG_SZ
Data: C:\InetPub\wwwroot,,1

Value 1
Name: /iisadmin
Type: REG_SZ
Data: C:\WINNT\System32\inetsrv\iisadmin,,1

Value 2
Name: /Scripts
Type: REG_SZ
Data: C:\InetPub\scripts,,4

Appendix C – Tunable Parameters

TPCC Application Registry Parameters

TPCC Application Registry Parameters

Key Name: SOFTWARE\TPCC
Class Name: <NO CLASS>
Last Write Time: 9/30/98 - 10:36 PM
Value 0
Name: DatabaseName
Type: REG_SZ
Data: tpcc

Value 1
Name: DatabasePassword
Type: REG_SZ
Data: tpcc

Value 2
Name: DatabaseServer
Type: REG_SZ
Data: TPCC1

Value 3
Name: DatabaseUser
Type: REG_SZ
Data: tpcc

Value 4
Name: DLLPath
Type: REG_SZ
Data: http:/scripts/tpcc/tpcc.dll

Value 5
Name: LogPath
Type: REG_EXPAND_SZ
Data: c:\logs

Value 6
Name: MaxUsersThisClient
Type: REG_DWORD
Data: 0xfde8

Value 7
Name: NumberOfWarehousesTotal
Type: REG_DWORD
Data: 0x2710

Tuxedo Configuration File

```
*RESOURCES
IPCKEY                  150000

DOMAINID   CLIENT1
MASTER                  CLIENT1
MAXACCESSERS            300
MAXSERVERS              65
MAXSERVICES             350
MODEL                   SHM
LDBAL                   N
SCANUNIT   15
BLOCKTIME 60
BBLQUERY   60

*MACHINES
DEFAULT:

"CLIENT1"    LMID=CLIENT1
                        APPDIR="c:\tpcc"
                        TUXCONFIG="c:\tpcc\tuxconfig1"
                        TUXDIR="c:\tuxedo"
                    ULOGPFX="c:\tpcc\ulog1"
                    TYPE="WinNT"
                    UID= 0
                    GID= 0
*GROUPS
GROUPNO
                LMID=CLIENT1            GRPNO=1     OPENINFO=NONE
GROUPOTHERS
                LMID=CLIENT1            GRPNO=2     OPENINFO=NONE

*SERVERS
DEFAULT:
tmall               SRVGRP=GROUPNO SRVID=100
                        MIN=30 MAX=50
                        CLOPT="-A"
                        RQADDR=all REPLYQ=Y
*SERVICES
```

RTE Input Parameters

Rte8000w.cfg

```
Sequent
Oracle 804 3mins login, 40.40mins rampup, 3hrs runtime, 8704w, 8000w active
483 33 10800 5 1 1 1
trans_params_tpcc
/scripts/tpcc/tpcc.dll 87 8704
slave1 client1_1            1 83
slave2 client1_2            84 167
slave3 client1_3            168 250
slave4 client1_4            251 333
slave5 client1_5            334 417
slave6 client1_6            418 500
slave7 client1_7            545 627
slave8 client1_8            628 711
slave9 client1_9            712 794
slave10 client1_10          795 877
slave11 client1_11          878 961
slave12 client1_12          962 1044
slave13 client2_1           1089 1171
slave14 client2_2           1172 1255
slave15 client2_3           1256 1338
slave16 client2_4           1339 1421
slave17 client2_5           1422 1505
slave18 client2_6           1506 1588
slave19 client2_7           1633 1715
slave20 client2_8           1716 1799
slave21 client2_9           1800 1882
slave22 client2_10          1883 1965
slave23 client2_11          1966 2049
slave24 client2_12          2050 2132
slave25 client3_1           2177 2259
slave26 client3_2           2260 2343
slave27 client3_3           2344 2426
slave28 client3_4           2427 2509
slave29 client3_5           2510 2593
slave30 client3_6           2594 2676
slave31 client3_7           2721 2803
slave32 client3_8           2804 2887
slave33 client3_9           2888 2970
slave34 client3_10          2971 3053
slave35 client3_11          3054 3137
slave36 client3_12          3138 3220
slave37 client4_1           3265 3347
slave38 client4_2           3348 3431
slave39 client4_3           3432 3514
slave40 client4_4           3515 3597
slave41 client4_5           3598 3681
slave42 client4_6           3682 3764
slave43 client4_7           3809 3891
slave44 client4_8           3892 3975
slave45 client4_9           3976 4058
slave46 client4_10          4059 4141
slave47 client4_11          4142 4225
slave48 client4_12          4226 4308
slave49 client5_1           4353 4435
slave50 client5_2           4436 4519
slave51 client5_3           4520 4602
slave52 client5_4           4603 4685
slave53 client5_5           4686 4769
slave54 client5_6           4770 4852
slave55 client5_7           4897 4979
slave56 client5_8           4980 5063
slave57 client5_9           5064 5146
slave58 client5_10          5147 5229
slave59 client5_11          5230 5313
slave60 client5_12          5314 5396
slave61 client6_1           5441 5523
slave62 client6_2           5524 5607
slave63 client6_3           5608 5690
slave64 client6_4           5691 5773
slave65 client6_5           5774 5857
slave66 client6_6           5858 5940
slave67 client6_7           5985 6067
```

slave68 client6_8 6068 6151
slave69 client6_9 6152 6234
slave70 client6_10 6235 6317
slave71 client6_11 6318 6401
slave72 client6_12 6402 6484
slave73 client7_1 6529 6611
slave74 client7_2 6612 6695
slave75 client7_3 6696 6778
slave76 client7_4 6779 6861
slave77 client7_5 6862 6945
slave78 client7_6 6946 7028
slave79 client7_7 7073 7155
slave80 client7_8 7156 7239
slave81 client7_9 7240 7322
slave82 client7_10 7323 7405
slave83 client7_11 7406 7489
slave84 client7_12 7490 7572
slave85 client8_1 7617 7699
slave86 client8_2 7700 7783
slave87 client8_3 7784 7866
slave88 client8_4 7867 7949
slave89 client8_5 7950 8033
slave90 client8_6 8034 8116
slave91 client8_7 8161 8243
slave92 client8_8 8244 8327
slave93 client8_9 8328 8410
slave94 client8_10 8411 8493
slave95 client8_11 8494 8577
slave96 client8_12 8578 8660

trans_params_tpcc

NewOrder 44.80 12.02 18.88 0.10 5.00 0.10
Payment 43.05 12.03 3.88 0.10 5.00 0.10
OrderStatus 4.05 10.08 2.88 0.10 5.00 0.10
Delivery 4.05 5.03 2.88 0.10 5.00 0.10
StockLevel 4.05 5.03 2.88 0.10 20.00 0.10

Appendix D – Price Quotes

10/12/98 MON 14:32 FAX 1 415 908 4820 BEA SYSTEMS 002

BEA

ENTERPRISE MIDDLEWARE SOLUTIONS

BEA SYSTEMS, INC.

Quotation for:

Tommy Tse

Sequent Computer Systems
15450 SW Koll Parkway

Beaverton, OR 97006

Phone: 503-578-5077
Fax: 503-578-3811

Please make purchase order to:

BEA Systems, Inc.
585 Moffett Park Boulevard
Sunnyvale, CA 94809

Sales Contact:
Bill Dana

Phone: 415-908-4802
Fax: 415-908-4820

Quote #BD101289-Revision 1.0: This quote expires 12/30/98.

Sequent Computer Systems

Tuxedo CFS Pricing

Item #	Product Description	List Price	Qnty	Extended Price
1	BEA Tuxedo CFS Tier 2 Unlimited User License	$ 12,000	1	$ 12,000
	- (4 CPU Intel NT Server)			
	Product Total			$ 12,000
	Annual Maintenance (5 x 8)			$ 1,800
	Annual Maintenance (7 x 24)			2,640

Y2K

All software contained in this quotation meets year 2000 compliance therefore, will function normally after January 1, 2000.

525 Market Street
Suite 3450
San Francisco, CA 94105

BEA Systems, Inc.
Price Quotation

10/12/98

10-12-98 14:35 RECEIVED FROM:1 415 908 4828 P-02

Appendix D – Price Quotes

10/12/98 MON 14:32 FAX 1 415 908 4820 BEA SYSTEMS 002

BEA

ENTERPRISE MIDDLEWARE SOLUTIONS

BEA SYSTEMS, INC.

Quotation for:

Tommy Tse

Sequent Computer Systems
15450 SW Koll Parkway

Beaverton, OR 97006

Phone: 503-578-5077
Fax: 503-578-3811

Please make purchase order to:

BEA Systems, Inc.
585 Moffett Park Boulevard
Sunnyvale, CA 94809

Sales Contact:
Bill Dana

Phone: 415-908-4802
Fax: 415-908-4820

Quote #BD101289-Revision 1.0: This quote expires 12/30/98.

Sequent Computer Systems

Tuxedo CFS Pricing

Item #	Product Description	List Price	Qnty	Extended Price
1	BEA Tuxedo CFS Tier 2 Unlimited User License	$ 12,000	1	$ 12,000
	- (4 CPU Intel NT Server)			
	Product Total			$ 12,000
	Annual Maintenance (5 x 8)			$ 1,800
	Annual Maintenance (7 x 24)			2,640

Y2K

All software contained in this quotation meets year 2000 compliance therefore, will function normally after January 1, 2000.

525 Market Street
Suite 3450
San Francisco, CA 94105

BEA Systems, Inc.
Price Quotation

10/12/98

10-12-98 14:35 RECEIVED FROM:1 415 908 4828 P-02