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

TPC Benchmark™C
Full Disclosure Report

First Edition
13-Oct-1998

First Printing, *13-Oct-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, 13-Oct-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	$6,222,683	48,793.40	$127.53/tpmC	HW: 15-Mar-1999. SW: Currently available.

1.2 Auditor

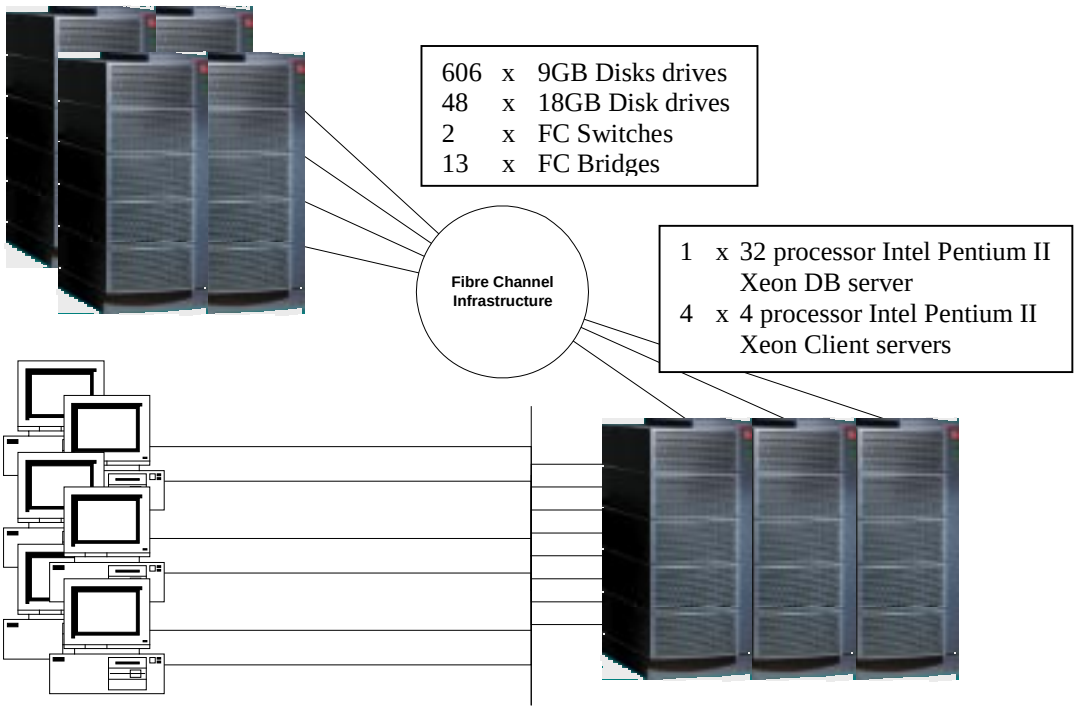
The results of the benchmark and test methodology used to produce the results were audited by Tom Sawyer of Performance Metrics, Inc. and have fully met the TPC-C rev 3.4 specifications.

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 13-Oct-1998
Total System Cost	**TPC-C Throughput**	**Price/Performance**	**Availability Date**
$6,222,683	**48,793.40**	**$127.53/tpmC**	15-Mar-1999

Processors	Database Manager	OS	Other Software	Number of Users
32 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++	40320



System Component	Server		Each Client	
Processors Cache	32	Intel Pentium II Xeon® 405MHz 2MB	2	Intel Pentium II Xeon® @ 400MHz 512 KB
Memory	1	32GB	1	1GB
Disk Controllers	16 16	Fiber controllers SCSI controllers	 1	 Adaptec On-Board
Disk Drives	606 48	9 GB (8.473 GB useable) 18GB disks (17.1 useable)	1	9GB
Total Storage		5905GB		9GB
Other	9 1	PCI 1-port Ethernet Tape Drive	1 3	CD-ROM PCI 4-port Ethernet NIC

SEQUENT / ORACLE	NUMACenter 2000	TPC-C Rev 3.4 Report Date 13-Oct-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	2	13,000	
Boot Pbay	EXP-PBAY-04	1	3,900	1	3,900	
SCSI Pbay cable	CBL-0300-RL	1	59	17	1,003	
PCI SCSI Adapter	IOC-0120-51	1	500	17	8,500	
Xeon Quad w/ IQ-LINK 405MHz 2MB	QUAD-0301	1	159,250	8	1,274,000	
IQ Plus-IQ Link interconnect cable	CBL-PLUS-03	1	455	8	3,640	
8G Addon Memory for Xeon Quad	MEM-1100-00	1	127,000	3	381,000	
Single port ether	COM-0041-50	1	650	9	5,850	
Fibre Channel PCI Host adapter	IOC-0210-52	1	715	16	11,440	
Server Hardware total					**1,708,833**	
Storage Subsystem						
48 x 9GB disk Storage unit - Includes 1 FC Bridge, 4 PBAYs, Cables, Cabinet	DSK-0948-40	1	84,500	12	1,014,000	
48 x 18GB disk Storage unit - Includes 1 FC Bridge, 4 PBAYs, Cables, Cabinet	DSK-1848-40	1	123,500	1	123,500	
9gb Add-on disk drive	DSK-0009-40	1	1,380	30	41,400	
Add-on PBAY	EXP-PBAY-00	1	3,900	6	23,400	
Fibre Channel Switch	FCS-0016-02	1	32,450	2	64,900	
Storage Subsystem total					**1,267,200**	
Server Hardware total					**2,976,033**	**595,207**
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-4100-001	1	2,500	1	2,500	
Sequent Volume Manager for Additional Quads	SVM-4100-ADD	1	2,430	7	17,010	
Server software subtotal					**19,510**	**3,902**
Oracle8 DataBase licenses	ORA8.0.4	2	1,071,372	1	1,071,372	1,071,372
Server Software total					**1,090,882**	**1,075,274**
Client Hardware						
Xeon 4x 400MHz 512K cache 1G mem	NTM-NODE-20	1	40,860	4	163,440	
4-port Ether	COM-0040-50	1	435	12	5,220	
Client Hardware Subtotal					**168,660**	**33,732**
Client Software						
MS Windows NT Server 4.0 w/5 CAL	OS-NT4-00	1	$809	4	3,236	970
Tuxedo 6.3 Core Functionality Services for NT		3	$12,000	4	48,000	36,000
Visual C++ 32 bit Edition		Microsoft	$499	1	499	150
Client Software subtotal					**51,735**	**37,120**
User Connectivity						
COMPEX MicroHub/8 8+1 A11Port 10BaseT Hub (including 10% spare)	DEH2924	Compex	$35	5,544	194,040	
User Connectivity subtotal					**194,040**	
Total cost of hardware and software					**4,481,350**	**1,741,333**
	Five-Year Cost of Ownership:				**$6,222,683**	
	TPM-C rating				**48,793.40**	
	$ / tpmC:				**$127.53**	

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			**48,793.4tpm-C**
% throughput difference, reported & reproducibility runs			<1%
Response Times (in seconds)	Average	90th	Max
- Neworder	0.96	1.800	463.29
- Payment	0.74	1.500	1038.51
- Order Status	0.78	1.500	8.22
- Delivery (interactive portion)	0.78	1.280	6.97
- Delivery (deferred portion)	0.96	1.409	735.26
- Stock-Level	1.00	1.900	8.11
- Menu	0.28	0.690	6.23
Response time delay added for emulated components			Menu 0.1 Resp 0.1
Transaction Mix, in percent of total transactions			
- New-Order			44.80%
- Payment			43.05%
- Order-Status			4.03%
- Delivery			4.04%
- Stock-Level			4.05%

Keying/Think Times (in seconds),	Min		Average		Max	
- New-Order	18.00	0.01	18.01	12.01	18.08	120.03
- Payment	3.00	0.01	3.01	12.04	3.07	120.24
- Order-Status	2.00	0.01	2.01	10.14	2.06	102.02
- Delivery	2.00	0.01	2.01	5.02	2.05	50.01
- Stock-Level	2.00	0.01	2.01	5.00	2.06	50.02

Test Duration	
- Ramp-up time	60 minutes
- Measurement interval	120 minutes
- Number of checkpoints	4
- Checkpoint interval	30 minutes
- Number of transactions (all types) completed in measurement interval	13068161

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 4 independent instances of version 4.0 of the operating system, each running on a 4 processor node. These 4 instances hosted the TPC application code. The 32 processor single instance of Dynix/ptx 4.4.4 hosted the Oracle 8.0.4 database software.

Eight remote terminal emulators (RTE) emulated 40,320 users executing the standard TPC-C workload. The RTE's were connected to the Client partitions via 10BaseT hubs.

The I/O subsystem comprised 2 Fiber Channel switches connected to 13 Bridges, supporting 100 SCSI channels, each with 6 disks. In addition, 16 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 8 RTE slave processes (one per network segment), distributed across 8 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

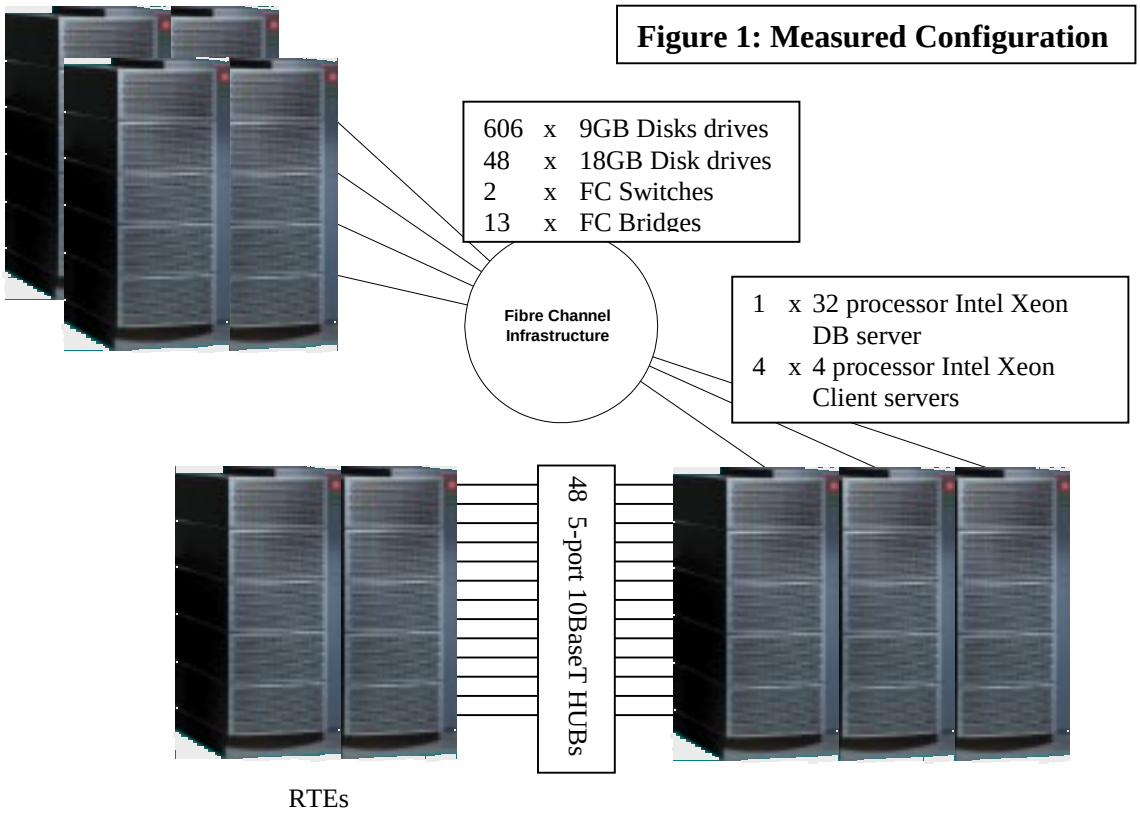
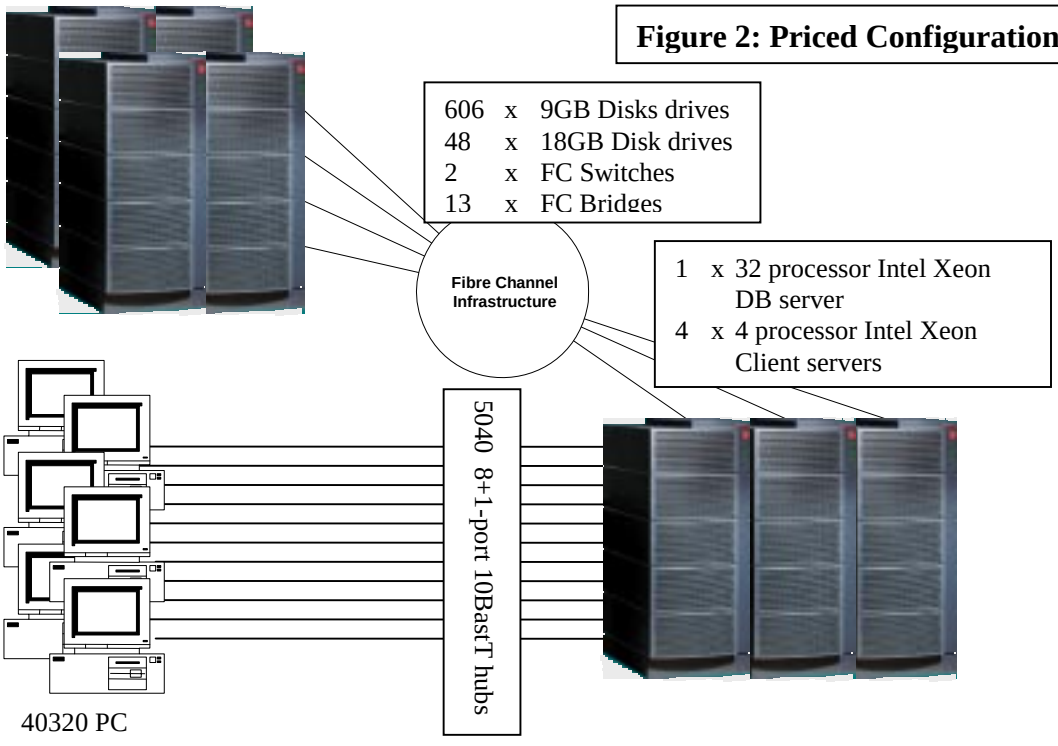


Figure 2: Priced Configuration



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 600 disk drives for data and 48 mirrored 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	84.98%
	Remote Warehouse	15.02%
	Non-Primary Key Access	60.01%
Order Status	Non-Primary Key Access	59.97%
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	44.80%
Payment	43.05%
Order Status	4.03%
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)

All ACID property tests were successful. The executions are described below.

6.1.1 Atomicity

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

Completed Transactions

A row was selected in a script from the warehouse, district and customer tables, and the balances noted. A payment transaction was started with the same warehouse, district and customer identifiers and a known amount. The payment transaction was committed and the rows were verified to contain correctly updated balances.

Aborted Transactions

A row was selected in a script from the warehouse, district and customer tables, and the balances noted. A payment transaction was started with the same warehouse, district and customer identifiers and a known amount. The payment transaction was rolled back and the rows were verified to contain the original balances.

6.1.2 Consistency

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

Consistency conditions one through four were tested using a shell script to issue queries to the database. The results of the queries verified that the database was consistent for all four tests. A run was executed under full load lasting over ten (10) minutes and included a checkpoint. The shell script was executed again. The result of the same queries verified that the database remained consistent after the run.

6.1.3 Isolation

Sufficient conditions must be enabled at either the system or application level to ensure the required isolation defined above (clause 3.4.1) is obtained.

Isolation tests one through seven were executed using shell scripts to issue queries to the database. Each script included timestamps to demonstrate the concurrency of operations. The results of the queries were captured to files. The captured files were verified by the auditor to demonstrate the required isolation had been met.

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

For Isolation test seven, case A was followed.

6.1.4 Durability

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

Durable Media Failure

For convenience, durability from media failure was demonstrated on a 10 Warehouse database having similar characteristics to the fully scaled database. The standard driving mechanism was used to generate the transaction load of 100 users for the Loss of Data. The fully scaled database under full load would also have passed the following test.

Loss of Log and Data

Loss of data was demonstrated on a 10 Warehouse database for convenience. The standard driving mechanism was used to generate the transaction load of 100 users for the test. To demonstrate recovery from a permanent failure of durable media containing TPC-C tables, the following steps were executed:

1. The database was backed up to extra disks.
2. The total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
3. The RTE was started with 100 users.
4. The test was allowed to run for more than 5 minutes.
5. One of the log disks was removed from the system. Since the log disk was mirrored using Oracle feature, the transactions continued to run without interruption.
6. The test was allowed to run for another 5 minutes and a data disk was removed from the system. Oracle aborted upon loss of the data disk.
7. The RTE's were shut down.
8. The data disk was put back into the system, and data restored from backup.
9. Oracle was restarted. Oracle automatically performed data recovery.
10. Step 2 was repeated to get the difference between the before and after counts.
11. The success file was used to determine the number of NEW_ORDERS successfully returned to the RTE's.
12. The counts in step 10 and 11 were compared, verifying that all committed transactions were successfully recovered.
13. Data from the success file was used to query the database to demonstrate that successful transactions had corresponding rows in the ORDER table and that rolled back transactions did not.

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 4608 warehouse under a full load of 46080 users. The following steps were executed:

1. The sum of D_NEXT_O_ID was taken.
2. 46080 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 4608 warehouses. No rows were deleted from the warehouse table prior to the run per Clause 4.2.2 of the TPC specification. The warehouse range was divided into 8 parts and 72 warehouses at the end of each part were skipped as per clause 4.2.2 of the TPC-C specification. The RTE was configured to distribute transactions evenly over the remaining portions of the warehouse range. This was verified by the auditor, who examined the D_NEXT_O_ID.

Table 3: Table Cardinality

Table	Cardinality as Benchmarked
Warehouse	4608
District	46,080
Customer	138,240,000
History	138,240,000
Orders	138,240,000
New Order	41,472,000
Order Line	1,382,303,435
Stock	460,800,000
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.

Table 5: Data Distribution

Controller	Drives	channel	Size	Use
Bridge 1	9 GB	Chan. 1-4, disks 1-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan. 1-4, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan. 1-4, disk 6	980 MB each	stock, cust & ord
	9 GB	Chan 5-8, disk 1	888 MB each	stock, cust, icust1, icust2 & iord1
	9 GB	Chan 5-8, disks 2-3	718 MB each	stock , cust , icust1 & icust2
	9 GB	Chan 5-8, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 5-8, disk 6	2652 MB each	ordl & iord2
Bridge 2	9 GB	Chan 1 , disk 1	1164 MB each	stock, cust, icust1, icust2 & inord
	9 GB	Chan 2 , disk 1	755 MB each	stock, cust, icust1, icust2 & ware
	9 GB	Chan 3 , disk 1	970 MB each	stock, cust, icust1, icust2 & nord
	9 GB	Chan 1-3 , disks 2-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan 1-4 , disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 1-3 , disk 6	2652 MB each	ordl & iord2
	9 GB	Chan 4 , disks 1-4	718 MB each	stock & cust, icust1,icust2
	9 GB	Chan 4 , disk 6	1500 MB each	hist
	9 GB	Chan. 5-8, disks 1-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan. 5-8, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan. 5-8, disk 6	980 MB each	stock, cust & ord
Bridge 3	9 GB	Chan 1-4, disk 1	888 MB each	stock, cust, icust1, icust2 & iord1
	9 GB	Chan 1-4, disks 2-3	718 MB each	stock , cust , icust1 & icust2
	9 GB	Chan 1-4, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 1-4, disk 6	2652 MB each	ordl & iord2
	9 GB	Chan 5 , disk 1	1164 MB each	stock, cust, icust1, icust2 & inord
	9 GB	Chan 6 , disk 1	755 MB each	stock, cust, icust1, icust2 & ware
	9 GB	Chan 7 , disk 1	970 MB each	stock, cust, icust1, icust2 & nord

	9 GB	Chan 5-7 , disks 2-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan 5-8, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 5-7, disk 6	2652 MB each	ordl & iord2
	9 GB	Chan 8 , disks 1-4	718 MB each	stock & cust, icust1,icust2
	9 GB	Chan 8 , disk 6	1500 MB each	hist
Bridge 4	9 GB	Chan. 1-4, disks 1-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan. 1-4, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan. 1-4, disk 6	980 MB each	stock, cust & ord
	9 GB	Chan 5-8, disk 1	888 MB each	stock, cust, icust1, icust2 & iord1
	9 GB	Chan 5-8, disks 2-3	718 MB each	stock , cust , icust1 & icust2
	9 GB	Chan 5-8, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 5-8, disk 6	2652 MB each	ordl & iord2
Bridge 5	9 GB	Chan 1 , disk 1	1164 MB each	stock, cust, icust1, icust2 & inord
	9 GB	Chan 2 , disk 1	755 MB each	stock, cust, icust1, icust2 & ware
	9 GB	Chan 3 , disk 1	970 MB each	stock, cust, icust1, icust2 & nord
	9 GB	Chan 1-3 , disks 2-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan 1-4 , disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 1-3 , disk 6	2652 MB each	ordl & iord2
	9 GB	Chan 4 , disks 1-4	718 MB each	stock & cust, icust1,icust2
	9 GB	Chan 4 , disk 6	1500 MB each	hist
	9 GB	Chan. 5-8, disks 0-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan. 5-8, disk 5	1256 MB each	stock, cust, item & iordl
		Chan. 5-8, disk 6	980 MB each	stock, cust & ord
Bridge 6	9 GB	Chan 1-4, disk 1	888 MB each	stock, cust, icust1, icust2 & iord1
	9 GB	Chan 1-4, disks 2-3	718 MB each	stock , cust , icust1 & icust2
	9 GB	Chan 1-4, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 1-4, disk 6	2652 MB each	ordl & iord2
	9 GB	Chan 5 , disk 1	1164 MB each	stock, cust, icust1, icust2 & inord
	9 GB	Chan 6 , disk 1	755 MB each	stock, cust, icust1, icust2 & ware
	9 GB	Chan 7 , disk 1	970 MB each	stock, cust, icust1, icust2 & nord
	9 GB	Chan 5-7 , disks 2-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan 5-8, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 5-7, disk 6	2652 MB each	ordl & iord2
	9 GB	Chan 8 , disks 1-4	718 MB each	stock & cust, icust1,icust2
	9 GB	Chan 8 , disk 6	1500 MB each	hist
Bridge 7	9 GB	Chan. 1-4, disks 1-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan. 1-4, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan. 1-4, disk 6	980 MB each	stock, cust & ord
	9 GB	Chan 5-8, disk 1	888 MB each	stock, cust, icust1, icust2 & iord1
	9 GB	Chan 5-8, disks 2-3	718 MB each	stock , cust , icust1 & icust2
	9 GB	Chan 5-8, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 5-8, disk 6	2652 MB each	ordl & iord2
Bridge 8	9 GB	Chan 1 , disk 1	1164 MB each	stock, cust, icust1, icust2 & inord
	9 GB	Chan 2 , disk 1	755 MB each	stock, cust, icust1, icust2 & ware
	9 GB	Chan 3 , disk 1	970 MB each	stock, cust, icust1, icust2 & nord
	9 GB	Chan 1-3 , disks 2-4	718 MB each	stock, cust, icust1 & icust2

	9 GB	Chan 1-4 , disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 1-3 , disk 6	2652 MB each	ordl & iord2
	9 GB	Chan 4 , disks 1-4	718 MB each	stock & cust, icust1,icust2
	9 GB	Chan 4 , disk 6	1500 MB each	hist
	9 GB	Chan. 5-8, disks 1-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan. 5-8, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan. 5-8, disk 6	980 MB each	stock, cust & ord
Bridge 9	9 GB	Chan 1-4, disk 1	888 MB each	stock, cust, icust1, icust2 & iord1
	9 GB	Chan 1-4, disks 2-3	718 MB each	stock , cust , icust1 & icust2
	9 GB	Chan 1-4, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 1-4, disk 6	2652 MB each	ordl & iord2
	9 GB	Chan 5 , disk 1	1164 MB each	stock, cust, icust1, icust2 & inord
	9 GB	Chan 6 , disk 1	755 MB each	stock, cust, icust1, icust2 & ware
	9 GB	Chan 7 , disk 1	970 MB each	stock, cust, icust1, icust2 & nord
	9 GB	Chan 5-7 , disks 2-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan 5-8, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 5-7, disk 6	2652 MB each	ordl & iord2
	9 GB	Chan 8 , disks 1-4	718 MB each	stock & cust, icust1,icust2
	9 GB	Chan 8 , disk 6	1500 MB each	hist
Bridge 10	9 GB	Chan. 1-4, disks 1-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan. 1-4, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan. 1-4, disk 6	980 MB each	stock, cust & ord
	9 GB	Chan 5-8, disk 1	888 MB each	stock, cust, icust1, icust2 & iord1
	9 GB	Chan 5-8, disks 2-3	718 MB each	stock , cust , icust1 & icust2
	9 GB	Chan 5-8, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 5-8, disk 6	2652 MB each	ordl & iord2
Bridge 11	9 GB	Chan 1 , disk 1	1164 MB each	stock, cust, icust1, icust2 & inord
	9 GB	Chan 2 , disk 1	755 MB each	stock, cust, icust1, icust2 & ware
	9 GB	Chan 3 , disk 1	970 MB each	stock, cust, icust1, icust2 & nord
	9 GB	Chan 1-3 , disks 2-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan 1-4 , disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 1-3 , disk 6	2652 MB each	ordl & iord2
	9 GB	Chan 4 , disks 1-4	718 MB each	stock & cust, icust1,icust2
	9 GB	Chan 4 , disk 6	1500 MB each	hist
	9 GB	Chan. 5-8, disks 1-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan. 5-8, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan. 5-8, disk 6	980 MB each	stock, cust & ord
Bridge 12	9 GB	Chan 1-4, disk 1	888 MB each	stock, cust, icust1, icust2 & iord1
	9 GB	Chan 1-4, disks 2-3	718 MB each	stock , cust , icust1 & icust2
	9 GB	Chan 1-4, disk 5	1256 MB each	stock, cust, item & iordl
	9 GB	Chan 1-4, disk 6	2652 MB each	ordl & iord2
	9 GB	Chan 5 , disk 1	1164 MB each	stock, cust, icust1, icust2 & inord
	9 GB	Chan 6 , disk 1	755 MB each	stock, cust, icust1, icust2 & ware
	9 GB	Chan 7 , disk 1	970 MB each	stock, cust, icust1, icust2 & nord
	9 GB	Chan 5-7 , disks 2-4	718 MB each	stock, cust, icust1 & icust2
	9 GB	Chan 5-8, disk 5	1256 MB each	stock, cust, item & iordl

	9 GB	Chan 5-7, disk 6	2652 MB each	ordl & iord2
	9 GB	Chan 8 , disks 1-4	718 MB each	stock & cust, icust1,icust2
	9 GB	Chan 8 , disk 6	1500 MB each	hist
Bridge 13	9 GB	Chan 1-4 disk 1-6	650 MB each	roll & istk
	9 GB	Chan 1 disk 1	750 MB each	system
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

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: 4608	**tpmC: 48,793.40**

Segment	**Rows**	**Data 2K pages**	**Index 2K pages**	**Extra 5%**	**Total with 5%**
Warehouse	4,608	4,632	800	272	5,704
District	46,080	46,112	8,000	2,706	56,818
Item	100,000	6,667	1,000	383	8,050
Customer	138,240,000	69,120,031	5,425,269	3,727,265	78,272,565
New_order	41,472,000	369,649	516,208	44,293	930,150
Stock	460,800,000	92,160,001	4,869,778	4,851,489	101,881,268
History (D)	138,240,000	3,921,652		0	3,921,652
Orders (D)	138,240,000	2,818,055	4,268,227	0	7,086,282
Order_line (D)	1,382,400,00 0	51,828,294	18,829,369	0	70,657,663
Roll_back	0	1,781,760	0	0	0
Totals (in 2K pages)		**222,056,853**	**33,918,651**	**8,626,407**	**262,820,151**

Dynamic space (MB)	114,390.63
Static space (MB)	398,929.98
Free space (MB)	77,449.36
Daily growth (MB)	19,380.23
Daily spread (MB)	48,379.01
180 day space (MB)	3,887,372.24
Log block size	512
Log blocks/tpmC	30.83
8hr log w/o mirror (GB)	344.36
180 day space (GB)	**3,796.26**
Logical logs w/ mirror (GB)	**688.71**
Total	**4,484.98**

	Capacity (GB)	**# Required**	**# Configured**
Database	8.49	448	600
Logs	17.09	42	48
O/S & Oracle			5
Total drives			653

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	48,793.40
Price per TpmC	$127.53

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	0.96	1.800	463.29
Payment	0.74	1.500	1038.51
Order Status	0.78	1.500	8.22
Interactive Delivery	0.78	1.280	4.91
Deferred Delivery	0.96	1.409	735.26
Stock Level	0.78	1.280	6.97
Menu	0.28	0.690	6.23

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	18.00	18.01	18.08
Payment	3.00	3.01	3.07
Order Status	2.00	2.01	2.06
Delivery	2.00	2.01	2.05
Stock Level	2.00	2.01	2.06

Table 7: Transaction Think Times

Transaction	Minimum	Average	Maximum
New Order	0.01	12.01	120.03
Payment	0.01	12.04	120.24
Order Status	2.01	10.14	102.02
Delivery	0.01	5.02	50.01
Stock Level	0.01	5.00	50.02

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

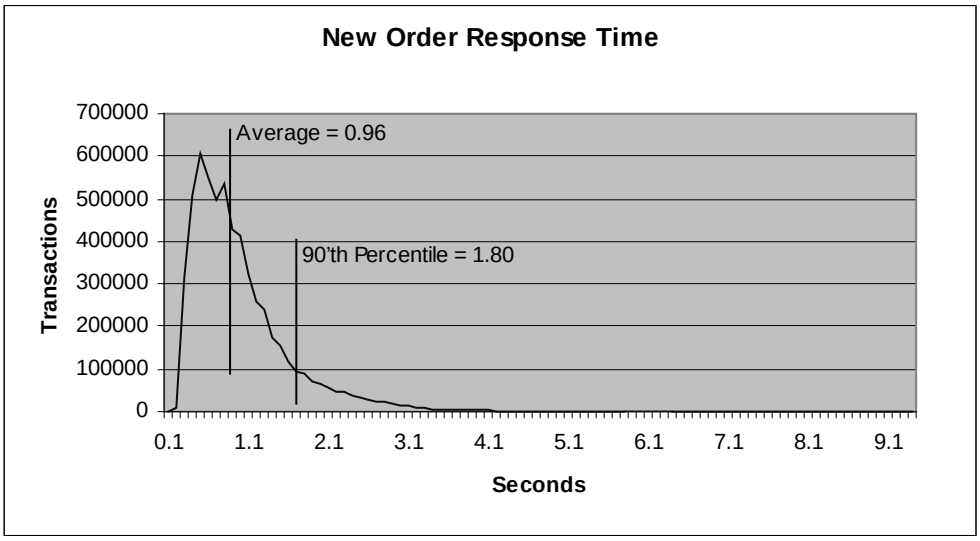


Figure 2: Payment Response Time Distribution

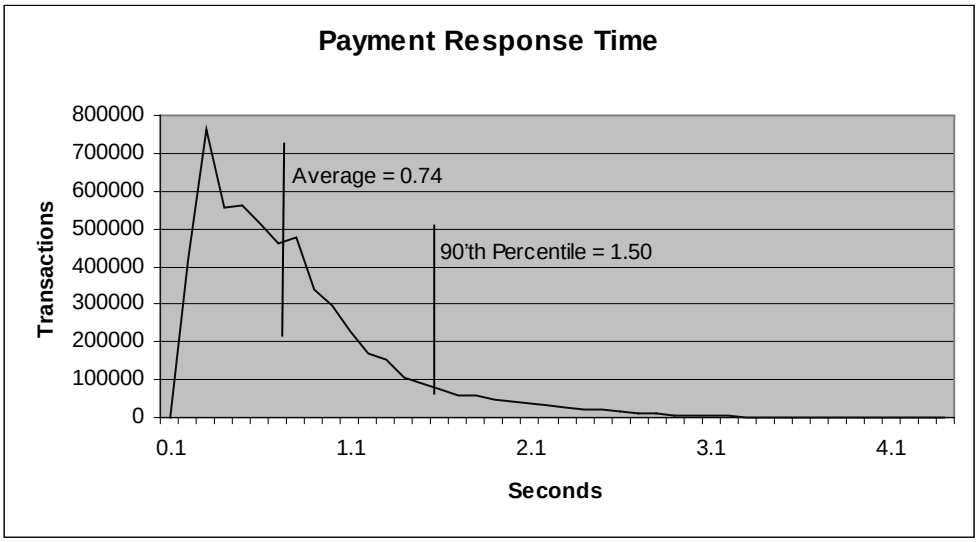


Figure 3: Order Status Response Time Distribution

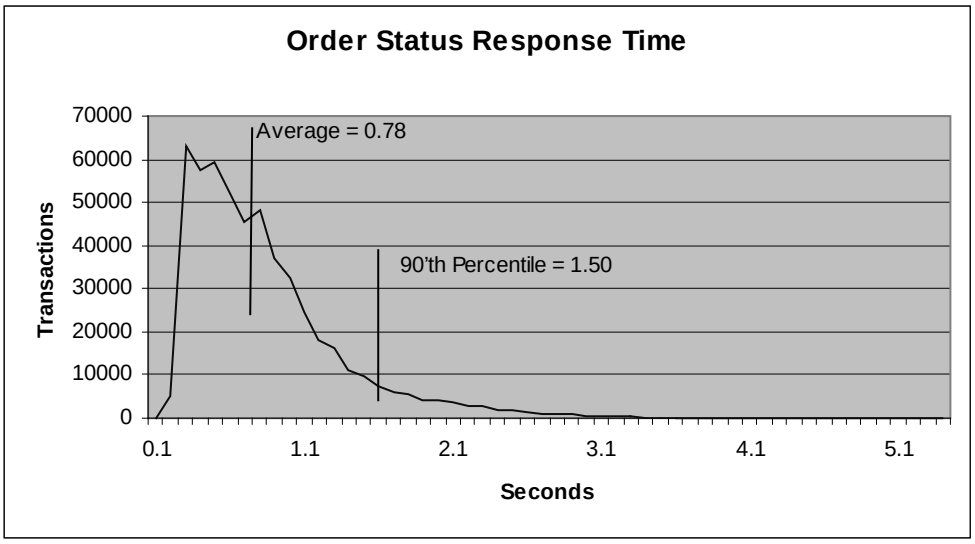


Figure 4: Delivery Response Time Distribution

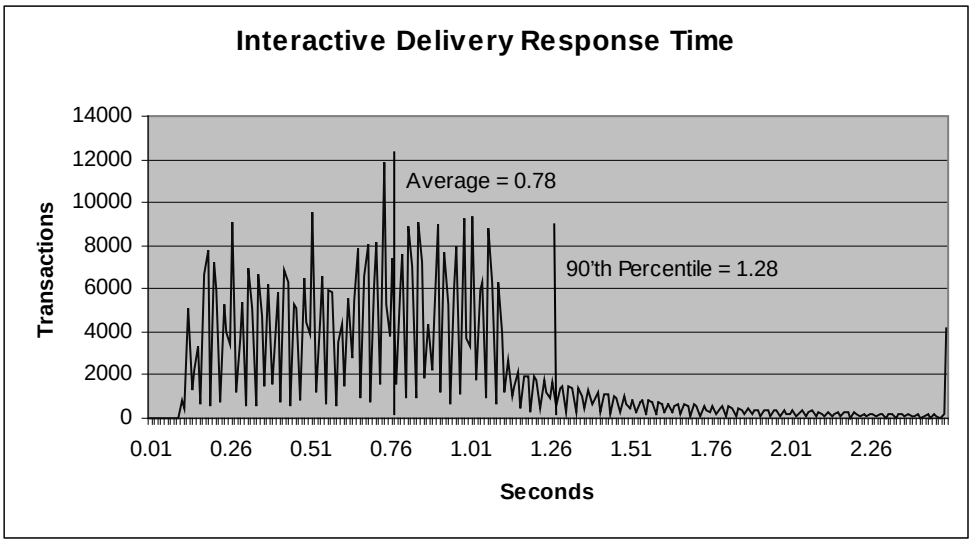
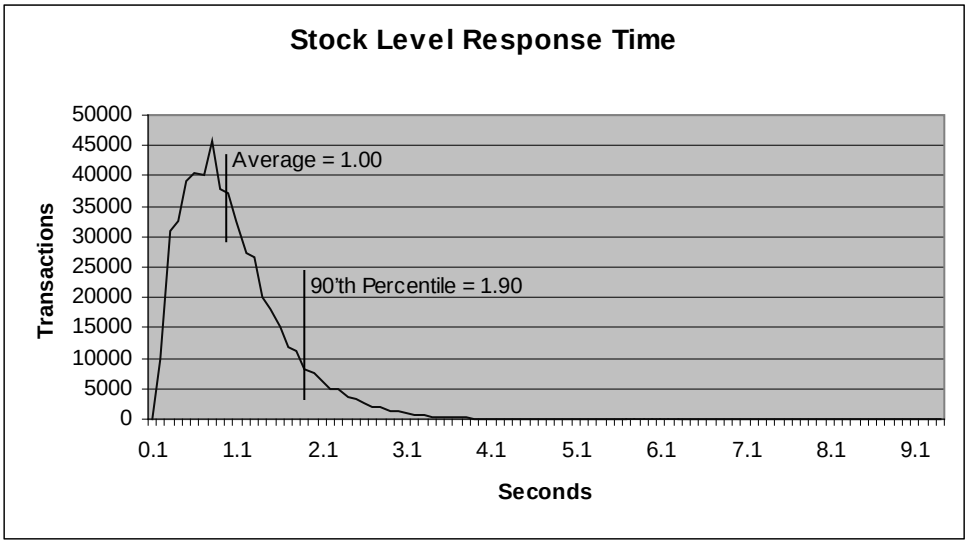


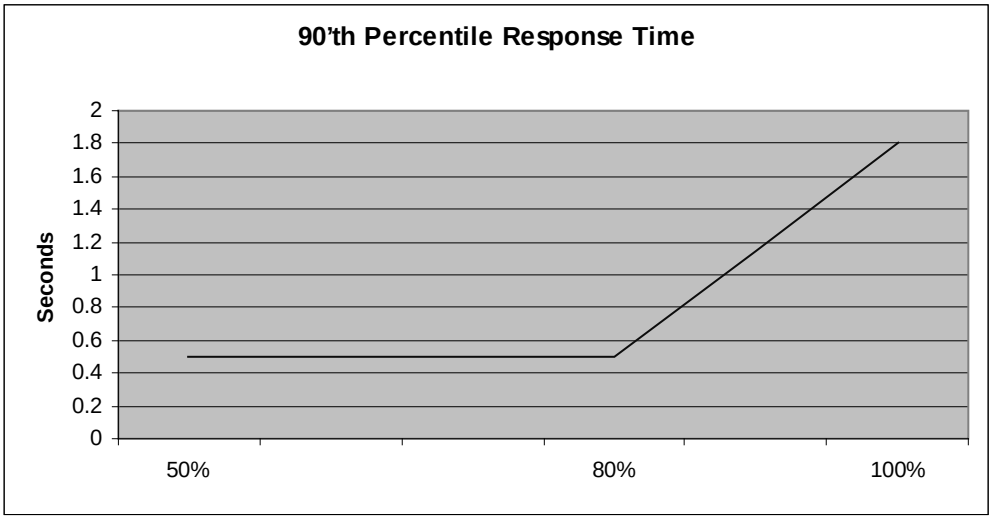
Figure 7: Stock Level Response Time Distribution



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)

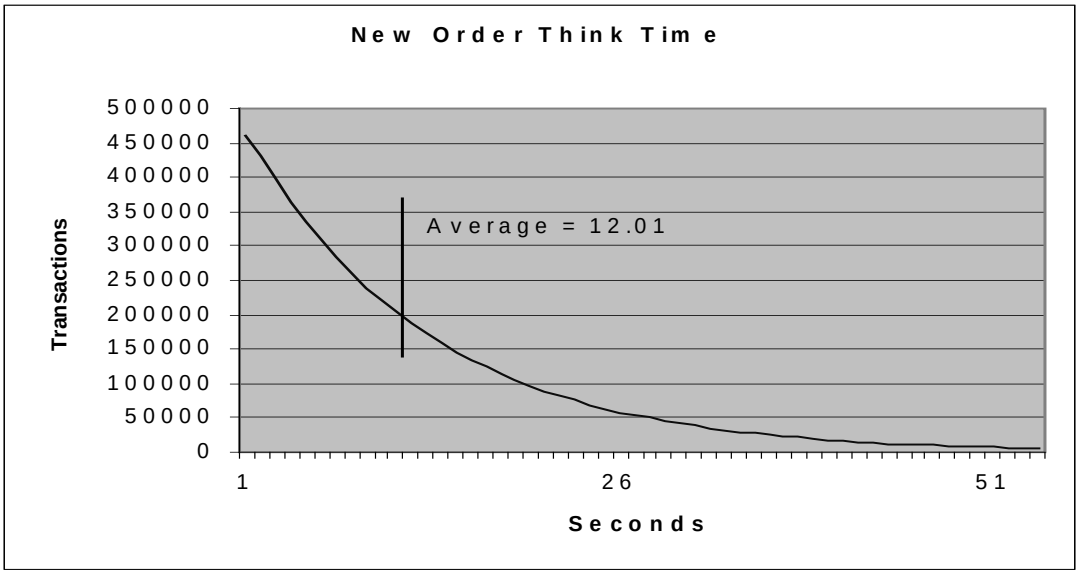
Figure 8: New Order Response Time vs. Throughput



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)

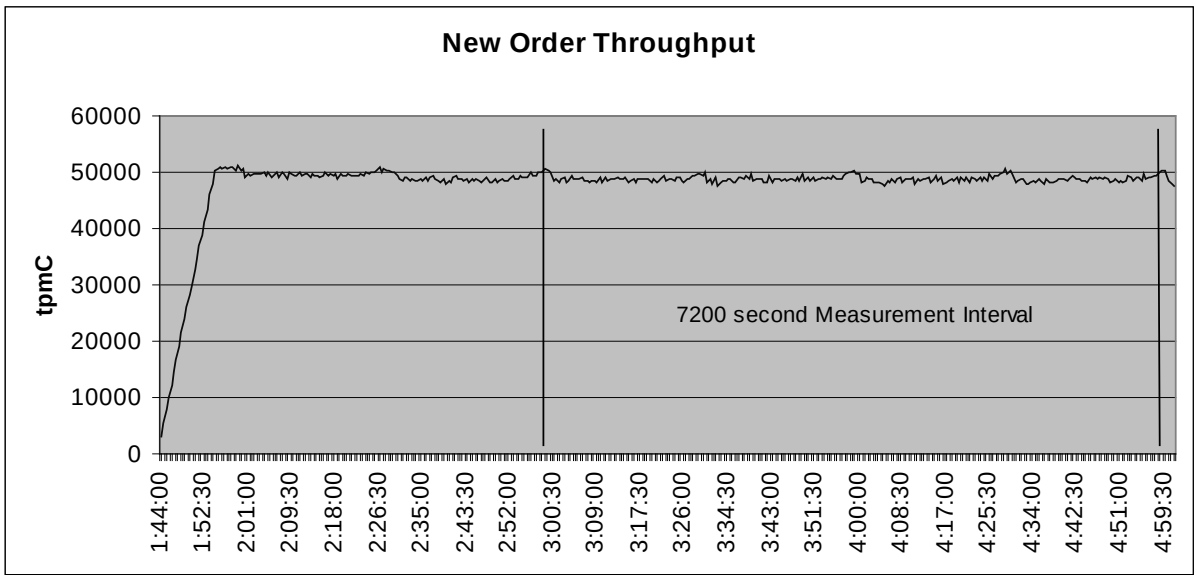
Figure 9: New Order Think Time Distribution



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 each checkpoints. The checkpoint script was started manually 5 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 therefore does 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 0.4% of the reported interval 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	44.80%
Payment	43.05%
Order Status	4.03%
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	84.98%
	Remote Warehouse	15.02%
	Non-Primary Key Access	60.01%
Order Status	Non-Primary Key Access	59.97
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 90 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)

Eight 100Mbps ethernet connections were used between the clients and the database server. Forty-eight 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:	March 15th 1998
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:	48,793.40 TpmC
Price Performance Metric:	$127.53 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:

- 4 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

October 7, 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
Transaction Manager:	BEA Tuxedo CFS Version 6.3
Operating System:	Windows NT 4.0 Enterprise Edition

Server: E300				
CPU's	Memory	Disks	90% Response	tpmC
32 Pentium II Xeon @ 405MHz	Main: 32 GB Cache: 2MB each	606 @ 9.1GB 48 @ 18GB	1.80 sec	48,793.40
4 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 4608 warehouses; 576 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 properties were met.
- The ACID tests were performed on a database scaled to 10 warehouses.
- 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.
- Two checkpoints were taken before the measured interval.
- Guard zone requirements do not apply.
- The system pricing was checked for major components and maintenance.

Auditor Notes:

The implementation used 8 instances of Oracle. The warehouse range was divided into 8 parts and 72 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.

Sincerely,

Tom Sawyer

Auditor

2229 Benita Dr. Suite 101, Rancho Cordova, CA 95670
(916) 635-2822 fax: (916) 858-0109 email: Loma@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"
                "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
%-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-----------------------------------
```

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.
```

```
 */

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_%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"
```

```
#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

Benchsetup.sh

```
#
˜
#=========================================================
========+
˜
#       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
#=========================================================
========
#

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=4608

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
 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
   $BUILD_SQL/create_rollback_segments.sh
 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)
#

#
# Get No. of CPU to use.
```

```
NCPUS=10
#
# Get no. of Warehouse / CPU which can be loaded
WPCPU=‘expr $MULT / $NCPUS ‘


print '\n\n\n\n\n Loading CUSTOMER table data for '$NCPUS' CPUs, '$MULT'
warehouses, and '$WPCPU' per load. . .\n\n\n\n\n\n'
I=1
SW=1
EW=$WPCPU
INC=$WPCPU
while [ $I -le $NCPUS ]
do
  tpccload -M $MULT -c -b $SW -e $EW > ${OUTDIR}/cust${I}.out 2>&1 &
  I=‘expr $I + 1‘
  SW=‘expr $SW + $INC‘
  EW=‘expr $EW + $INC‘
done

#
#

#
# If the MULT is not exactly divisible by NCPUS then load the remaining rows

REM=‘expr $MULT % $NCPUS ‘

if [ $REM -ne 0 ]; then
      SW=‘expr $WPCPU \* $NCPUS + 1‘
      EW=$MULT
            I=‘expr $NCPUS + 1 ‘
            tpccload -M $MULT -c -b $SW -e $EW >
${OUTDIR}/cust${I}.out 2>&1 &
fi


# wait

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

print '\n\n\n\n\n Loading STOCK table data . . .\n\n\n\n\n\n'

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

wait

#
#
#

switchlog.sh

#

if [ -f /tmp/exit_load ]; then
            echo "Exit for now. Build Indexs later..."
            exit;
fi



#
#  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\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\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\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\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 55000 $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;
!



#
#  Shutdown database
#

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

```
  shutdown;
  exit;
!
```

benchdb.sh

```
#
~
#=========================================================
========+
#       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
#=========================================================
========
#

print '\n\n\n\n\nStarting BENCHDB.SH\n\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=4608

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 tpcc controlfile reuse maxdatafiles 2000
maxinstances 8
                   datafile '?/dbs/tpcc_disks/system_1' size 750M reuse
                   logfile  '?/dbs/tpcc_disks/log_1' size 16000M reuse,
                                '?/dbs/tpcc_disks/log_2' size 16000M reuse,
                                '?/dbs/tpcc_disks/log_3' size 16000M reuse;
                exit;
!

#
# Create more rollback segments
#
  sleep 5

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

  svrmgrl <<!
    set echo on;
    connect system/manager
    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;
!

ps -fu oracle|grep [o]ra_dbw|sed 's/oracle \([^ ]*\).*a_dbw\(.\).*/\1 \2 /|
while read a b
do
surun /etc/rqadmin -assign $a $(( $b + 1 ))
done
surun /etc/rqadmin -assign $(ps -fu oracle|nawk '/[l]gw/   { print $2 }') 1

#
#
# Create the tablespaces in parallel
 print '\nInitalizing first datafiles in all tablespaces . . .\n'

                crts.sh ordl_1 /usr/oracle/dbs/tpcc_disks/ordl_1 4375M &
                crts.sh ordl_2 /usr/oracle/dbs/tpcc_disks/ordl_2 4375M &
                crts.sh ordl_3 /usr/oracle/dbs/tpcc_disks/ordl_3 4375M &
                crts.sh ordl_4 /usr/oracle/dbs/tpcc_disks/ordl_4 4375M &
                crts.sh ordl_5 /usr/oracle/dbs/tpcc_disks/ordl_5 4375M &
                crts.sh ordl_6 /usr/oracle/dbs/tpcc_disks/ordl_6 4375M &
                crts.sh ordl_7 /usr/oracle/dbs/tpcc_disks/ordl_7 4375M &
                crts.sh ordl_8 /usr/oracle/dbs/tpcc_disks/ordl_8 4375M &


                crts.sh ordl_9 /usr/oracle/dbs/tpcc_disks/ordl_9 4375M &
                crts.sh ordl_10 /usr/oracle/dbs/tpcc_disks/ordl_10 4375M &
                crts.sh ordl_11 /usr/oracle/dbs/tpcc_disks/ordl_11 4375M &
                crts.sh ordl_12 /usr/oracle/dbs/tpcc_disks/ordl_12 4375M &
                crts.sh ordl_13 /usr/oracle/dbs/tpcc_disks/ordl_13 4375M &
                crts.sh ordl_14 /usr/oracle/dbs/tpcc_disks/ordl_14 4375M &
                crts.sh ordl_15 /usr/oracle/dbs/tpcc_disks/ordl_15 4375M &
                crts.sh ordl_16 /usr/oracle/dbs/tpcc_disks/ordl_16 4375M &


                crts.sh ordl_17 /usr/oracle/dbs/tpcc_disks/ordl_17 4375M &
```

```
crts.sh ordl_18 /usr/oracle/dbs/tpcc_disks/ordl_18 4375M &
crts.sh ordl_19 /usr/oracle/dbs/tpcc_disks/ordl_19 4375M &
crts.sh ordl_20 /usr/oracle/dbs/tpcc_disks/ordl_20 4375M &
crts.sh ordl_21 /usr/oracle/dbs/tpcc_disks/ordl_21 4375M &
crts.sh ordl_22 /usr/oracle/dbs/tpcc_disks/ordl_22 4375M &
crts.sh ordl_23 /usr/oracle/dbs/tpcc_disks/ordl_23 4375M &
crts.sh ordl_24 /usr/oracle/dbs/tpcc_disks/ordl_24 4375M &

crts.sh ordl_25 /usr/oracle/dbs/tpcc_disks/ordl_25 4375M &
crts.sh ordl_26 /usr/oracle/dbs/tpcc_disks/ordl_26 4375M &
crts.sh ordl_27 /usr/oracle/dbs/tpcc_disks/ordl_27 4375M &
crts.sh ordl_28 /usr/oracle/dbs/tpcc_disks/ordl_28 4375M &
crts.sh ordl_29 /usr/oracle/dbs/tpcc_disks/ordl_29 4375M &
crts.sh ordl_30 /usr/oracle/dbs/tpcc_disks/ordl_30 4375M &
crts.sh ordl_31 /usr/oracle/dbs/tpcc_disks/ordl_31 4375M &
crts.sh ordl_32 /usr/oracle/dbs/tpcc_disks/ordl_32 4375M &

wait

crts.sh iordl_1 /usr/oracle/dbs/tpcc_disks/iordl_1 1719M &
crts.sh iordl_2 /usr/oracle/dbs/tpcc_disks/iordl_2 1719M &
crts.sh iordl_3 /usr/oracle/dbs/tpcc_disks/iordl_3 1719M &
crts.sh iordl_4 /usr/oracle/dbs/tpcc_disks/iordl_4 1719M &
crts.sh iordl_5 /usr/oracle/dbs/tpcc_disks/iordl_5 1719M &
crts.sh iordl_6 /usr/oracle/dbs/tpcc_disks/iordl_6 1719M &
crts.sh iordl_7 /usr/oracle/dbs/tpcc_disks/iordl_7 1719M &
crts.sh iordl_8 /usr/oracle/dbs/tpcc_disks/iordl_8 1719M &

crts.sh iordl_9 /usr/oracle/dbs/tpcc_disks/iordl_9 1719M &
crts.sh iordl_10 /usr/oracle/dbs/tpcc_disks/iordl_10 1719M &
crts.sh iordl_11 /usr/oracle/dbs/tpcc_disks/iordl_11 1719M &
crts.sh iordl_12 /usr/oracle/dbs/tpcc_disks/iordl_12 1719M &
crts.sh iordl_13 /usr/oracle/dbs/tpcc_disks/iordl_13 1719M &
crts.sh iordl_14 /usr/oracle/dbs/tpcc_disks/iordl_14 1719M &
crts.sh iordl_15 /usr/oracle/dbs/tpcc_disks/iordl_15 1719M &
crts.sh iordl_16 /usr/oracle/dbs/tpcc_disks/iordl_16 1719M &

crts.sh iordl_17 /usr/oracle/dbs/tpcc_disks/iordl_17 1719M &
crts.sh iordl_18 /usr/oracle/dbs/tpcc_disks/iordl_18 1719M &
crts.sh iordl_19 /usr/oracle/dbs/tpcc_disks/iordl_19 1719M &
crts.sh iordl_20 /usr/oracle/dbs/tpcc_disks/iordl_20 1719M &
crts.sh iordl_21 /usr/oracle/dbs/tpcc_disks/iordl_21 1719M &
crts.sh iordl_22 /usr/oracle/dbs/tpcc_disks/iordl_22 1719M &
crts.sh iordl_23 /usr/oracle/dbs/tpcc_disks/iordl_23 1719M &
crts.sh iordl_24 /usr/oracle/dbs/tpcc_disks/iordl_24 1719M &

crts.sh iordl_25 /usr/oracle/dbs/tpcc_disks/iordl_25 1719M &
crts.sh iordl_26 /usr/oracle/dbs/tpcc_disks/iordl_26 1719M &
crts.sh iordl_27 /usr/oracle/dbs/tpcc_disks/iordl_27 1719M &
crts.sh iordl_28 /usr/oracle/dbs/tpcc_disks/iordl_28 1719M &
crts.sh iordl_29 /usr/oracle/dbs/tpcc_disks/iordl_29 1719M &
crts.sh iordl_30 /usr/oracle/dbs/tpcc_disks/iordl_30 1719M &
crts.sh iordl_31 /usr/oracle/dbs/tpcc_disks/iordl_31 1719M &
crts.sh iordl_32 /usr/oracle/dbs/tpcc_disks/iordl_32 1719M &

wait

crts.sh ware_1 /usr/oracle/dbs/tpcc_disks/ware_1 36M &
crts.sh ware_2 /usr/oracle/dbs/tpcc_disks/ware_2 36M &
crts.sh ware_3 /usr/oracle/dbs/tpcc_disks/ware_3 36M &
crts.sh ware_4 /usr/oracle/dbs/tpcc_disks/ware_4 36M &
crts.sh ware_5 /usr/oracle/dbs/tpcc_disks/ware_5 36M &
crts.sh ware_6 /usr/oracle/dbs/tpcc_disks/ware_6 36M &
crts.sh ware_7 /usr/oracle/dbs/tpcc_disks/ware_7 36M &
crts.sh ware_8 /usr/oracle/dbs/tpcc_disks/ware_8 36M &

crts.sh hist_1 /usr/oracle/dbs/tpcc_disks/hist_1 1500M &
crts.sh hist_2 /usr/oracle/dbs/tpcc_disks/hist_2 1500M &
crts.sh hist_3 /usr/oracle/dbs/tpcc_disks/hist_3 1500M &
crts.sh hist_4 /usr/oracle/dbs/tpcc_disks/hist_4 1500M &
crts.sh hist_5 /usr/oracle/dbs/tpcc_disks/hist_5 1500M &
crts.sh hist_6 /usr/oracle/dbs/tpcc_disks/hist_6 1500M &
crts.sh hist_7 /usr/oracle/dbs/tpcc_disks/hist_7 1500M &
crts.sh hist_8 /usr/oracle/dbs/tpcc_disks/hist_8 1500M &

crts.sh nord_1 /usr/oracle/dbs/tpcc_disks/nord_1 250M &
crts.sh nord_2 /usr/oracle/dbs/tpcc_disks/nord_2 250M &
crts.sh nord_3 /usr/oracle/dbs/tpcc_disks/nord_3 250M &
crts.sh nord_4 /usr/oracle/dbs/tpcc_disks/nord_4 250M &
crts.sh nord_5 /usr/oracle/dbs/tpcc_disks/nord_5 250M &
crts.sh nord_6 /usr/oracle/dbs/tpcc_disks/nord_6 250M &
crts.sh nord_7 /usr/oracle/dbs/tpcc_disks/nord_7 250M &
crts.sh nord_8 /usr/oracle/dbs/tpcc_disks/nord_8 250M &

crts.sh inord_1 /usr/oracle/dbs/tpcc_disks/inord_1 446M &
crts.sh inord_2 /usr/oracle/dbs/tpcc_disks/inord_2 446M &
crts.sh inord_3 /usr/oracle/dbs/tpcc_disks/inord_3 446M &
crts.sh inord_4 /usr/oracle/dbs/tpcc_disks/inord_4 446M &
crts.sh inord_5 /usr/oracle/dbs/tpcc_disks/inord_5 446M &
crts.sh inord_6 /usr/oracle/dbs/tpcc_disks/inord_6 446M &
crts.sh inord_7 /usr/oracle/dbs/tpcc_disks/inord_7 446M &
crts.sh inord_8 /usr/oracle/dbs/tpcc_disks/inord_8 446M &

wait

crts.sh ord_1 /usr/oracle/dbs/tpcc_disks/ord_1 300M &
crts.sh ord_2 /usr/oracle/dbs/tpcc_disks/ord_2 300M &
crts.sh ord_3 /usr/oracle/dbs/tpcc_disks/ord_3 300M &
crts.sh ord_4 /usr/oracle/dbs/tpcc_disks/ord_4 300M &
crts.sh ord_5 /usr/oracle/dbs/tpcc_disks/ord_5 300M &
crts.sh ord_6 /usr/oracle/dbs/tpcc_disks/ord_6 300M &
crts.sh ord_7 /usr/oracle/dbs/tpcc_disks/ord_7 300M &
crts.sh ord_8 /usr/oracle/dbs/tpcc_disks/ord_8 300M &

crts.sh ord_9 /usr/oracle/dbs/tpcc_disks/ord_9 300M &
crts.sh ord_10 /usr/oracle/dbs/tpcc_disks/ord_10 300M &
crts.sh ord_11 /usr/oracle/dbs/tpcc_disks/ord_11 300M &
crts.sh ord_12 /usr/oracle/dbs/tpcc_disks/ord_12 300M &
crts.sh ord_13 /usr/oracle/dbs/tpcc_disks/ord_13 300M &
crts.sh ord_14 /usr/oracle/dbs/tpcc_disks/ord_14 300M &
crts.sh ord_15 /usr/oracle/dbs/tpcc_disks/ord_15 300M &
crts.sh ord_16 /usr/oracle/dbs/tpcc_disks/ord_16 300M &

crts.sh ord_17 /usr/oracle/dbs/tpcc_disks/ord_17 300M &
crts.sh ord_18 /usr/oracle/dbs/tpcc_disks/ord_18 300M &
crts.sh ord_19 /usr/oracle/dbs/tpcc_disks/ord_19 300M &
crts.sh ord_20 /usr/oracle/dbs/tpcc_disks/ord_20 300M &
crts.sh ord_21 /usr/oracle/dbs/tpcc_disks/ord_21 300M &
crts.sh ord_22 /usr/oracle/dbs/tpcc_disks/ord_22 300M &
crts.sh ord_23 /usr/oracle/dbs/tpcc_disks/ord_23 300M &
crts.sh ord_24 /usr/oracle/dbs/tpcc_disks/ord_24 300M &

crts.sh ord_25 /usr/oracle/dbs/tpcc_disks/ord_25 300M &
crts.sh ord_26 /usr/oracle/dbs/tpcc_disks/ord_26 300M &
crts.sh ord_27 /usr/oracle/dbs/tpcc_disks/ord_27 300M &
crts.sh ord_28 /usr/oracle/dbs/tpcc_disks/ord_28 300M &
crts.sh ord_29 /usr/oracle/dbs/tpcc_disks/ord_29 300M &
crts.sh ord_30 /usr/oracle/dbs/tpcc_disks/ord_30 300M &
crts.sh ord_31 /usr/oracle/dbs/tpcc_disks/ord_31 300M &
crts.sh ord_32 /usr/oracle/dbs/tpcc_disks/ord_32 300M &

wait

crts.sh iord2_1 /usr/oracle/dbs/tpcc_disks/iord2_1 265M &
crts.sh iord2_2 /usr/oracle/dbs/tpcc_disks/iord2_2 265M &
crts.sh iord2_3 /usr/oracle/dbs/tpcc_disks/iord2_3 265M &
crts.sh iord2_4 /usr/oracle/dbs/tpcc_disks/iord2_4 265M &
crts.sh iord2_5 /usr/oracle/dbs/tpcc_disks/iord2_5 265M &
crts.sh iord2_6 /usr/oracle/dbs/tpcc_disks/iord2_6 265M &
crts.sh iord2_7 /usr/oracle/dbs/tpcc_disks/iord2_7 265M &
crts.sh iord2_8 /usr/oracle/dbs/tpcc_disks/iord2_8 265M &

crts.sh iord2_9 /usr/oracle/dbs/tpcc_disks/iord2_9 265M &
crts.sh iord2_10 /usr/oracle/dbs/tpcc_disks/iord2_10 265M &
crts.sh iord2_11 /usr/oracle/dbs/tpcc_disks/iord2_11 265M &
crts.sh iord2_12 /usr/oracle/dbs/tpcc_disks/iord2_12 265M &
crts.sh iord2_13 /usr/oracle/dbs/tpcc_disks/iord2_13 265M &
crts.sh iord2_14 /usr/oracle/dbs/tpcc_disks/iord2_14 265M &
crts.sh iord2_15 /usr/oracle/dbs/tpcc_disks/iord2_15 265M &
crts.sh iord2_16 /usr/oracle/dbs/tpcc_disks/iord2_16 265M &

crts.sh iord2_17 /usr/oracle/dbs/tpcc_disks/iord2_17 265M &
crts.sh iord2_18 /usr/oracle/dbs/tpcc_disks/iord2_18 265M &
crts.sh iord2_19 /usr/oracle/dbs/tpcc_disks/iord2_19 265M &
crts.sh iord2_20 /usr/oracle/dbs/tpcc_disks/iord2_20 265M &
crts.sh iord2_21 /usr/oracle/dbs/tpcc_disks/iord2_21 265M &
crts.sh iord2_22 /usr/oracle/dbs/tpcc_disks/iord2_22 265M &
crts.sh iord2_23 /usr/oracle/dbs/tpcc_disks/iord2_23 265M &
crts.sh iord2_24 /usr/oracle/dbs/tpcc_disks/iord2_24 265M &

crts.sh iord2_25 /usr/oracle/dbs/tpcc_disks/iord2_25 265M &
crts.sh iord2_26 /usr/oracle/dbs/tpcc_disks/iord2_26 265M &
crts.sh iord2_27 /usr/oracle/dbs/tpcc_disks/iord2_27 265M &
crts.sh iord2_28 /usr/oracle/dbs/tpcc_disks/iord2_28 265M &
crts.sh iord2_29 /usr/oracle/dbs/tpcc_disks/iord2_29 265M &
crts.sh iord2_30 /usr/oracle/dbs/tpcc_disks/iord2_30 265M &
crts.sh iord2_31 /usr/oracle/dbs/tpcc_disks/iord2_31 265M &
crts.sh iord2_32 /usr/oracle/dbs/tpcc_disks/iord2_32 265M &
```

```
wait

        crts.sh iord1_1 /usr/oracle/dbs/tpcc_disks/iord1_1 170M &
        crts.sh iord1_2 /usr/oracle/dbs/tpcc_disks/iord1_2 170M &
        crts.sh iord1_3 /usr/oracle/dbs/tpcc_disks/iord1_3 170M &
        crts.sh iord1_4 /usr/oracle/dbs/tpcc_disks/iord1_4 170M &
        crts.sh iord1_5 /usr/oracle/dbs/tpcc_disks/iord1_5 170M &
        crts.sh iord1_6 /usr/oracle/dbs/tpcc_disks/iord1_6 170M &
        crts.sh iord1_7 /usr/oracle/dbs/tpcc_disks/iord1_7 170M &
        crts.sh iord1_8 /usr/oracle/dbs/tpcc_disks/iord1_8 170M &

        crts.sh iord1_9 /usr/oracle/dbs/tpcc_disks/iord1_9 170M &
        crts.sh iord1_10 /usr/oracle/dbs/tpcc_disks/iord1_10 170M &
        crts.sh iord1_11 /usr/oracle/dbs/tpcc_disks/iord1_11 170M &
        crts.sh iord1_12 /usr/oracle/dbs/tpcc_disks/iord1_12 170M &
        crts.sh iord1_13 /usr/oracle/dbs/tpcc_disks/iord1_13 170M &
        crts.sh iord1_14 /usr/oracle/dbs/tpcc_disks/iord1_14 170M &
        crts.sh iord1_15 /usr/oracle/dbs/tpcc_disks/iord1_15 170M &
        crts.sh iord1_16 /usr/oracle/dbs/tpcc_disks/iord1_16 170M &

        crts.sh iord1_17 /usr/oracle/dbs/tpcc_disks/iord1_17 170M &
        crts.sh iord1_18 /usr/oracle/dbs/tpcc_disks/iord1_18 170M &
        crts.sh iord1_19 /usr/oracle/dbs/tpcc_disks/iord1_19 170M &
        crts.sh iord1_20 /usr/oracle/dbs/tpcc_disks/iord1_20 170M &
        crts.sh iord1_21 /usr/oracle/dbs/tpcc_disks/iord1_21 170M &
        crts.sh iord1_22 /usr/oracle/dbs/tpcc_disks/iord1_22 170M &
        crts.sh iord1_23 /usr/oracle/dbs/tpcc_disks/iord1_23 170M &
        crts.sh iord1_24 /usr/oracle/dbs/tpcc_disks/iord1_24 170M &

        crts.sh iord1_25 /usr/oracle/dbs/tpcc_disks/iord1_25 170M &
        crts.sh iord1_26 /usr/oracle/dbs/tpcc_disks/iord1_26 170M &
        crts.sh iord1_27 /usr/oracle/dbs/tpcc_disks/iord1_27 170M &
        crts.sh iord1_28 /usr/oracle/dbs/tpcc_disks/iord1_28 170M &
        crts.sh iord1_29 /usr/oracle/dbs/tpcc_disks/iord1_29 170M &
        crts.sh iord1_30 /usr/oracle/dbs/tpcc_disks/iord1_30 170M &
        crts.sh iord1_31 /usr/oracle/dbs/tpcc_disks/iord1_31 170M &
        crts.sh iord1_32 /usr/oracle/dbs/tpcc_disks/iord1_32 170M &

wait

        crts.sh roll_1 /usr/oracle/dbs/tpcc_disks/roll_1 145M &
        crts.sh roll_2 /usr/oracle/dbs/tpcc_disks/roll_2 145M &
        crts.sh roll_3 /usr/oracle/dbs/tpcc_disks/roll_3 145M &
        crts.sh roll_4 /usr/oracle/dbs/tpcc_disks/roll_4 145M &
        crts.sh roll_5 /usr/oracle/dbs/tpcc_disks/roll_5 145M &
        crts.sh roll_6 /usr/oracle/dbs/tpcc_disks/roll_6 145M &
        crts.sh roll_7 /usr/oracle/dbs/tpcc_disks/roll_7 145M &
        crts.sh roll_8 /usr/oracle/dbs/tpcc_disks/roll_8 145M &

        crts.sh roll_9 /usr/oracle/dbs/tpcc_disks/roll_9 145M &
        crts.sh roll_10 /usr/oracle/dbs/tpcc_disks/roll_10 145M &
        crts.sh roll_11 /usr/oracle/dbs/tpcc_disks/roll_11 145M &
        crts.sh roll_12 /usr/oracle/dbs/tpcc_disks/roll_12 145M &
        crts.sh roll_13 /usr/oracle/dbs/tpcc_disks/roll_13 145M &
        crts.sh roll_14 /usr/oracle/dbs/tpcc_disks/roll_14 145M &
        crts.sh roll_15 /usr/oracle/dbs/tpcc_disks/roll_15 145M &
        crts.sh roll_16 /usr/oracle/dbs/tpcc_disks/roll_16 145M &

        crts.sh roll_17 /usr/oracle/dbs/tpcc_disks/roll_17 145M &
        crts.sh roll_18 /usr/oracle/dbs/tpcc_disks/roll_18 145M &
        crts.sh roll_19 /usr/oracle/dbs/tpcc_disks/roll_19 145M &
        crts.sh roll_20 /usr/oracle/dbs/tpcc_disks/roll_20 145M &
        crts.sh roll_21 /usr/oracle/dbs/tpcc_disks/roll_21 145M &
        crts.sh roll_22 /usr/oracle/dbs/tpcc_disks/roll_22 145M &
        crts.sh roll_23 /usr/oracle/dbs/tpcc_disks/roll_23 145M &
        crts.sh roll_24 /usr/oracle/dbs/tpcc_disks/roll_24 145M &

wait

        crts.sh items /usr/oracle/dbs/tpcc_disks/items 195M &

        crts.sh icust1 /usr/oracle/dbs/tpcc_disks/icust1 4356M &

        crts.sh icust2 /usr/oracle/dbs/tpcc_disks/icust2_1 5000M &

        crts.sh istk /usr/oracle/dbs/tpcc_disks/istk_1 499M &
        crts.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_1 390M &
        crts.sh cust /usr/oracle/dbs/tpcc_disks/cust_1 290M &
        crts.sh temp    /usr/oracle/dbs/tpcc_disks/temp_1_1          7500M
&

wait

        addfile.sh icust2 /usr/oracle/dbs/tpcc_disks/icust2_2 5000M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_2 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_3 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_4 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_5 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_6 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_7 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_8 499M &

        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_9 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_10 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_11 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_12 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_13 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_14 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_15 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_16 499M &

        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_17 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_18 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_19 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_20 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_21 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_22 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_23 499M &
        addfile.sh istk /usr/oracle/dbs/tpcc_disks/istk_24 499M &

wait

        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_2 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_3 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_4 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_5 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_6 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_7 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_8 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_9 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_10 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_11 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_12 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_13 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_14 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_15 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_16 390M &

        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_17 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_18 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_19 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_20 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_21 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_22 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_23 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_24 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_25 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_26 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_27 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_28 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_29 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_30 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_31 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_32 390M &

        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_33 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_34 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_35 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_36 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_37 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_38 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_39 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_40 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_41 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_42 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_43 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_44 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_45 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_46 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_47 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_48 390M &

        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_49 390M &
```

```
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_50 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_51 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_52 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_53 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_54 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_55 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_56 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_57 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_58 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_59 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_60 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_61 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_62 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_63 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_64 390M &
wait
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_65 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_66 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_67 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_68 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_69 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_70 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_71 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_72 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_73 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_74 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_75 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_76 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_77 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_78 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_79 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_80 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_81 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_82 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_83 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_84 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_85 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_86 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_87 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_88 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_89 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_90 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_91 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_92 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_93 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_94 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_95 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_96 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_97 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_98 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_99 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_100 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_101 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_102 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_103 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_104 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_105 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_106 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_107 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_108 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_109 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_110 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_111 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_112 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_113 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_114 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_115 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_116 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_117 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_118 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_119 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_120 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_121 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_122 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_123 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_124 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_125 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_126 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_127 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_128 390M &
wait
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_129 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_130 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_131 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_132 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_133 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_134 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_135 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_136 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_137 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_138 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_139 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_140 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_141 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_142 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_143 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_144 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_145 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_146 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_147 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_148 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_149 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_150 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_151 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_152 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_153 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_154 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_155 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_156 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_157 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_158 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_159 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_160 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_161 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_162 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_163 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_164 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_165 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_166 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_167 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_168 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_169 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_170 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_171 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_172 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_173 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_174 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_175 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_176 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_177 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_178 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_179 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_180 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_181 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_182 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_183 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_184 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_185 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_186 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_187 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_188 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_189 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_190 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_191 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_192 390M &
wait
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_193 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_194 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_195 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_196 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_197 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_198 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_199 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_200 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_201 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_202 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_203 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_204 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_205 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_206 390M &
```

```
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_207 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_208 390M &

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_209 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_210 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_211 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_212 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_213 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_214 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_215 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_216 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_217 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_218 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_219 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_220 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_221 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_222 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_223 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_224 390M &

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_225 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_226 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_227 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_228 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_229 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_230 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_231 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_232 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_233 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_234 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_235 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_236 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_237 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_238 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_239 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_240 390M &

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_241 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_242 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_243 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_244 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_245 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_246 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_247 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_248 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_249 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_250 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_251 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_252 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_253 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_254 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_255 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_256 390M &
wait

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_257 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_258 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_259 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_260 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_261 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_262 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_263 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_264 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_265 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_266 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_267 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_268 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_269 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_270 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_271 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_272 390M &

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_273 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_274 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_275 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_276 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_277 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_278 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_279 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_280 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_281 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_282 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_283 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_284 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_285 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_286 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_287 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_288 390M &

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_289 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_290 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_291 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_292 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_293 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_294 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_295 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_296 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_297 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_298 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_299 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_300 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_301 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_302 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_303 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_304 390M &

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_305 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_306 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_307 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_308 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_309 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_310 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_311 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_312 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_313 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_314 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_315 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_316 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_317 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_318 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_319 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_320 390M &
wait

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_321 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_322 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_323 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_324 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_325 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_326 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_327 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_328 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_329 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_330 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_331 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_332 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_333 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_334 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_335 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_336 390M &

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_337 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_338 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_339 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_340 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_341 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_342 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_343 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_344 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_345 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_346 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_347 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_348 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_349 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_350 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_351 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_352 390M &

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_353 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_354 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_355 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_356 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_357 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_358 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_359 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_360 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_361 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_362 390M &
```

```
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_363 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_364 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_365 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_366 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_367 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_368 390M &

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_369 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_370 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_371 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_372 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_373 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_374 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_375 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_376 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_377 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_378 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_379 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_380 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_381 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_382 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_383 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_384 390M &
wait

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_385 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_386 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_387 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_388 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_389 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_390 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_391 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_392 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_393 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_394 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_395 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_396 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_397 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_398 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_399 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_400 390M &

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_401 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_402 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_403 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_404 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_405 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_406 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_407 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_408 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_409 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_410 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_411 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_412 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_413 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_414 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_415 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_416 390M &

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_417 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_418 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_419 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_420 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_421 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_422 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_423 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_424 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_425 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_426 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_427 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_428 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_429 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_430 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_431 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_432 390M &

addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_433 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_434 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_435 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_436 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_437 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_438 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_439 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_440 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_441 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_442 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_443 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_444 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_445 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_446 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_447 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_448 390M &
wait
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_449 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_450 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_451 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_452 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_453 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_454 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_455 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_456 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_457 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_458 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_459 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_460 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_461 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_462 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_463 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_464 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_465 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_466 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_467 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_468 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_469 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_470 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_471 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_472 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_473 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_474 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_475 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_476 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_477 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_478 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_479 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_480 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_481 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_482 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_483 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_484 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_485 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_486 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_487 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_488 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_489 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_490 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_491 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_492 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_493 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_494 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_495 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_496 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_497 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_498 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_499 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_500 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_501 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_502 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_503 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_504 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_505 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_506 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_507 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_508 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_509 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_510 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_511 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_512 390M &
wait
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_513 390M &
addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_514 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_515 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_516 390M &
# addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_517 390M &
```

```
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_518 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_519 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_520 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_521 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_522 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_523 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_524 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_525 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_526 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_527 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_528 390M &

#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_529 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_530 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_531 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_532 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_533 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_534 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_535 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_536 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_537 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_538 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_539 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_540 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_541 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_542 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_543 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_544 390M &

#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_545 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_546 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_547 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_548 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_549 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_550 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_551 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_552 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_553 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_554 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_555 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_556 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_557 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_558 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_559 390M &
#       addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_560 390M &
        addfile.sh stocks /usr/oracle/dbs/tpcc_disks/stocks_561 390M &

wait

        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_2 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_3 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_4 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_5 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_6 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_7 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_8 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_9 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_10 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_11 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_12 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_13 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_14 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_15 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_16 290M &

        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_17 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_18 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_19 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_20 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_21 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_22 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_23 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_24 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_25 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_26 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_27 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_28 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_29 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_30 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_31 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_32 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_33 290M &

        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_34 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_35 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_36 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_37 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_38 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_39 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_40 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_41 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_42 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_43 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_44 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_45 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_46 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_47 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_48 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_49 290M &

        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_50 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_51 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_52 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_53 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_54 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_55 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_56 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_57 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_58 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_59 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_60 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_61 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_62 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_63 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_64 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_65 290M &

wait

        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_66 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_67 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_68 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_69 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_70 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_71 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_72 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_73 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_74 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_75 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_76 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_77 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_78 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_79 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_80 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_81 290M &

        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_82 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_83 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_84 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_85 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_86 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_87 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_88 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_89 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_90 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_91 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_92 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_93 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_94 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_95 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_96 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_97 290M &

        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_98 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_99 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_100 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_101 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_102 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_103 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_104 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_105 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_106 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_107 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_108 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_109 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_110 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_111 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_112 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_113 290M &
```

```
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_114 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_115 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_116 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_117 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_118 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_119 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_120 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_121 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_122 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_123 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_124 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_125 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_126 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_127 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_128 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_129 290M &
wait

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_130 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_131 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_132 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_133 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_134 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_135 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_136 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_137 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_138 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_139 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_140 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_141 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_142 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_143 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_144 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_145 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_146 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_147 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_148 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_149 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_150 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_151 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_152 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_153 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_154 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_155 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_156 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_157 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_158 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_159 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_160 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_161 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_162 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_163 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_164 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_165 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_166 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_167 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_168 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_169 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_170 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_171 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_172 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_173 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_174 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_175 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_176 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_177 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_178 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_179 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_180 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_181 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_182 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_183 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_184 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_185 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_186 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_187 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_188 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_189 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_190 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_191 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_192 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_193 290M &
wait

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_194 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_195 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_196 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_197 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_198 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_199 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_200 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_201 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_202 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_203 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_204 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_205 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_206 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_207 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_208 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_209 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_210 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_211 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_212 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_213 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_214 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_215 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_216 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_217 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_218 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_219 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_220 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_221 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_222 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_223 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_224 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_225 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_226 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_227 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_228 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_229 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_230 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_231 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_232 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_233 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_234 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_235 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_236 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_237 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_238 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_239 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_240 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_241 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_242 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_243 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_244 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_245 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_246 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_247 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_248 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_249 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_250 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_251 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_252 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_253 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_254 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_255 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_256 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_257 290M &
wait

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_258 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_259 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_260 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_261 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_262 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_263 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_264 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_265 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_266 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_267 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_268 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_269 290M &
```

```
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_270 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_271 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_272 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_273 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_274 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_275 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_276 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_277 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_278 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_279 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_280 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_281 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_282 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_283 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_284 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_285 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_286 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_287 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_288 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_289 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_290 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_291 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_292 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_293 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_294 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_295 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_296 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_297 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_298 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_299 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_300 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_301 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_302 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_303 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_304 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_305 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_306 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_307 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_308 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_309 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_310 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_311 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_312 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_313 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_314 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_315 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_316 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_317 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_318 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_319 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_320 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_321 290M &
wait
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_322 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_323 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_324 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_325 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_326 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_327 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_328 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_329 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_330 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_331 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_332 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_333 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_334 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_335 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_336 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_337 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_338 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_339 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_340 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_341 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_342 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_343 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_344 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_345 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_346 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_347 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_348 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_349 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_350 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_351 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_352 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_353 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_354 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_355 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_356 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_357 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_358 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_359 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_360 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_361 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_362 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_363 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_364 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_365 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_366 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_367 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_368 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_369 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_370 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_371 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_372 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_373 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_374 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_375 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_376 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_377 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_378 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_379 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_380 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_381 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_382 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_383 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_384 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_385 290M &
wait
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_386 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_387 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_388 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_389 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_390 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_391 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_392 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_393 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_394 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_395 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_396 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_397 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_398 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_399 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_400 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_401 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_402 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_403 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_404 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_405 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_406 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_407 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_408 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_409 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_410 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_411 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_412 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_413 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_414 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_415 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_416 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_417 290M &

addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_418 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_419 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_420 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_421 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_422 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_423 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_424 290M &
addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_425 290M &
```

```
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_426 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_427 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_428 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_429 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_430 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_431 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_432 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_433 290M &

        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_434 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_435 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_436 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_437 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_438 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_439 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_440 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_441 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_442 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_443 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_444 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_445 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_446 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_447 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_448 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_449 290M &
wait
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_450 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_451 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_452 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_453 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_454 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_455 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_456 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_457 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_458 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_459 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_460 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_461 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_462 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_463 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_464 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_465 290M &

        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_466 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_467 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_468 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_469 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_470 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_471 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_472 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_473 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_474 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_475 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_476 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_477 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_478 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_479 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_480 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_481 290M &

        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_482 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_483 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_484 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_485 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_486 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_487 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_488 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_489 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_490 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_491 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_492 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_493 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_494 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_495 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_496 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_497 290M &

#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_498 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_499 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_500 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_501 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_502 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_503 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_504 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_505 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_506 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_507 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_508 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_509 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_510 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_511 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_512 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_513 290M &
wait
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_514 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_515 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_516 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_517 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_518 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_519 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_520 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_521 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_522 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_523 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_524 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_525 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_526 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_527 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_528 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_529 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_530 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_531 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_532 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_533 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_534 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_535 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_536 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_537 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_538 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_539 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_540 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_541 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_542 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_543 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_544 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_545 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_546 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_547 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_548 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_549 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_550 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_551 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_552 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_553 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_554 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_555 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_556 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_557 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_558 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_559 290M &
#       addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_560 290M &
        addfile.sh cust /usr/oracle/dbs/tpcc_disks/cust_561 290M &
wait
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_2_1 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_3_1 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_4_1 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_5_1 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_6_1 7500M &
wait
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_1_2 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_2_2 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_3_2 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_4_2 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_5_2 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_6_2 7500M &
wait
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_1_3 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_2_3 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_3_3 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_4_3 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_5_3 7500M &
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_6_3 7500M &
wait
        addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_1_4 7500M &
```

```
                addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_2_4 7500M &
                addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_3_4 7500M &
                addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_4_4 7500M &
                addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_5_4 7500M &
                addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_6_4 7500M &
wait
                addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_1_5 7500M &
                addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_2_5 7500M &
                addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_3_5 7500M &
                addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_4_5 7500M &
                addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_5_5 7500M &
                addfile.sh temp /usr/oracle/dbs/tpcc_disks/temp_6_5 7500M &
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;
        connect system/manager;
        @?/rdbms/admin/catdbsyn;
        @?/sqlplus/admin/pupbld;
        exit;
!
fi
```

pload.sh

```
#!/bin/ksh
˝
#
˝
#============================================================
========+
˝
#       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_LOADER=$BENCH_HOME/tpcc/loader
TPCC_ADMIN=admin

LDIR=data
OUTDIR=outdir
MULT=4608

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

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


#
#  Load history table
#

print 'Starting HISTORY data load . . .'
date

#
# For partitioned load set NFILES = no of partitions
NFILES=8

#
# Get no. of Warehouse / FILE which can be loaded
WPFILE=`expr $MULT / $NFILES `

I=1
while [ $I -le $NFILES ]
do
  /etc/mknod ${LDIR}/hist${I}.dat p
  I=`expr $I + 1`
done

print '     TPCCLOAD phase . . .'

I=1
SW=1
EW=$WPFILE
INC=$WPFILE
while [ $I -le $NFILES ]
do
  tpccload -M $MULT -h -g -b $SW -e $EW > ${LDIR}/hist${I}.dat 2> \
    ${OUTDIR}/hist${I}.out &
  I=`expr $I + 1`
  SW=`expr $SW + $INC`
  EW=`expr $EW + $INC`
done

sleep 30

print '\n\n\n\n\n     SQLLDR phase:'
print '          no. files/partitions:          '$NFILES';'
print '          warehouses per file: '$WPFILE';'
print '          warehouses:          '$MULT';\n\n\n\n\n\n'

I=1
while [ $I -le $NFILES ]
do

  sqlldr tpcc/tpcc control=$TPCC_LOADER/histP${I}.ctl \
       log=${OUTDIR}/hist${I}.log \
       bad=${OUTDIR}/hist${I}.bad data=${LDIR}/hist${I}.dat \
       discard=${OUTDIR}/hist${I}.dsc &

  I=`expr $I + 1`
done


wait

# If number of warehouses isn't directly divisable by number of files, then

REM=`expr $MULT % $NFILES `
```

```
if [ $REM -ne 0 ]; then
        SW=‘expr $NFILES \* $WPFILE + 1‘
        EW=$MULT
        NFILES=‘expr $NFILES + 1 ‘
                I=$NFILES
        /etc/mknod ${LDIR}/hist${I}.dat p
        tpccload -M $MULT -h -g -b $SW -e $EW > ${LDIR}/hist${I}.dat 2> \
          ${OUTDIR}/hist${I}.out &

                sleep 5

        sqlldr tpcc/tpcc control=$TPCC_LOADER/hist.ctl \
          log=${OUTDIR}/hist${I}.log \
          bad=${OUTDIR}/hist${I}.bad data=${LDIR}/hist${I}.dat \
          discard=${OUTDIR}/hist${I}.dsc &

fi


wait

print ’    HISTORY import DONE!’
print ’ ’

I=1
while [ $I -le $NFILES ]
do
   rm -f ${LDIR}/hist${I}.dat
   I=‘expr $I + 1‘
done


#  Load new-order table

print 'Starting NEW-ORDER table load . . .’
date

#
# For partitioned load set NFILES = no of partitions
NFILES=8

#
# Get no. of Warehouse / FILE which can be loaded
WPFILE=‘expr $MULT / $NFILES ‘

I=1
while [ $I -le $NFILES ]
do
   /etc/mknod ${LDIR}/neword${I}.dat p
   I=‘expr $I + 1‘
done

print ’    TPCCLOAD phase . . .’
I=1
SW=1
EW=$WPFILE
INC=$WPFILE
while [ $I -le $NFILES ]
do
   tpccload -M $MULT -n -g -b $SW -e $EW > ${LDIR}/neword${I}.dat 2> \
     ${OUTDIR}/neword${I}.out &
   I=‘expr $I + 1‘
   SW=‘expr $SW + $INC‘
   EW=‘expr $EW + $INC‘
done

sleep 30

print '\n\n\n\n\n     SQLLDR phase:’
print ’        no. files:          '$NFILES';’
print ’        warehouses per file: '$WPFILE';’
print ’        warehouses:         '$MULT';\n\n\n\n\n’


I=1
while [ $I -le $NFILES ]
do
  sqlldr tpcc/tpcc control=$TPCC_LOADER/newordP${I}.ctl \
        log=${OUTDIR}/neword${I}.log \
        bad=${OUTDIR}/neword${I}.bad data=${LDIR}/neword${I}.dat \
        discard=${OUTDIR}/neword${I}.dsc &
  I=‘expr $I + 1‘
done


wait

# if number of warehouses doesn't divide evenly by $NFILES, then . . .

REM=‘expr $MULT % $NFILES ‘

if [ $REM -ne 0 ]; then
        SW=‘expr $NFILES \* $WPFILE + 1‘
        EW=$MULT
        NFILES=‘expr $NFILES + 1 ‘
                I=$NFILES
        /etc/mknod ${LDIR}/neword${I}.dat p
        tpccload -M $MULT -n -g -b $SW -e $EW > ${LDIR}/neword${I}.dat 2> \
           ${OUTDIR}/neword${I}.out &

        sleep 5

        sqlldr tpcc/tpcc control=$TPCC_LOADER/neword.ctl \
          log=${OUTDIR}/neword${I}.log \
          bad=${OUTDIR}/neword${I}.bad data=${LDIR}/neword${I}.dat \
          discard=${OUTDIR}/neword${I}.dsc &

fi

wait

I=1
while [ $I -le $NFILES ]
do
   rm -f ${LDIR}/neword${I}.dat
   I=‘expr $I + 1‘
done



print 'NEWORD phase DONE!’
print ’ ’

#
#  Load order and order-line table
#
print 'Starting ORDER and ORDER-LINE import . . .’

#
# For partitioned load set NFILES = no of partitions
NFILES=32

#
# Set no. of procs / file
NPPFILE=1
#
# Get no. of Warehouse / FILE which can be loaded
WPFILE=‘expr $MULT / $NFILES / $NPPFILE ‘

NFF=‘expr $NFILES \* $NPPFILE ‘

I=1
while [ $I -le $NFF ]
do
   /etc/mknod ${LDIR}/order${I}.dat p
   /etc/mknod ${LDIR}/ordline${I}.dat p
   I=‘expr $I + 1‘
done

print ’    TPCCLOAD phase . . .’
K=0
SW=1
EW=$WPFILE
INC=$WPFILE
while [ $K -lt $NFILES ]
do
   J=1
   while [ $J -le $NPPFILE ]
   do
      I=‘expr $K \* $NPPFILE + $J  ‘
      tpccload -M $MULT -o ${LDIR}/ordline${I}.dat -g -b $SW -e $EW > \
        ${LDIR}/order${I}.dat 2> ${OUTDIR}/order${I}.out &
      SW=‘expr $SW + $INC‘
      EW=‘expr $EW + $INC‘

      J=‘expr $J + 1 ‘
   done
      K=‘expr $K + 1‘
done


sleep 30

print '\n\n\n\n\n     SQLLDR phase:’
print ’        no. files:          '$NFILES';’
print ’        warehouses per file: '$(( $WPFILE * $NPPFILE ));’
print ’        warehouses:         '$MULT';\n\n\n\n\n’
```

```
K=0
while [ $K -lt $NFILES ]
do
  J=1
  L=‘expr $K + 1 ‘
  while [ $J -le $NPPFILE ]
  do
    I=‘expr $K \* $NPPFILE + $J  ‘
    sqlldr tpcc/tpcc control=$TPCC_LOADER/orderP${L}.ctl \
       log=${OUTDIR}/order${I}.log \
       bad=${OUTDIR}/order${I}.bad data=${LDIR}/order${I}.dat \
       discard=${OUTDIR}/order${I}.dsc &

    sqlldr tpcc/tpcc control=$TPCC_LOADER/ordlineP${L}.ctl \
       log=${OUTDIR}/ordline${I}.log \
       bad=${OUTDIR}/ordline${I}.bad data=${LDIR}/ordline${I}.dat \
       discard=${OUTDIR}/ordline${I}.dsc &
    J=‘expr $J + 1 ‘

                sleep 5
  done
    K=‘expr $K + 1‘
done

wait

# if number of warehouses not even divisible by number of files, then . . .

REM=‘expr $MULT % \( $NFILES \* $NPPFILE \) ‘

if [ $REM -ne 0 ]; then
      SW=‘expr $NFILES \* $WPFILE \* $NPPFILE + 1 ‘
      EW=$MULT
      NFF=‘expr $NFILES \* $NPPFILE + 1 ‘
                I=$NFF

      /etc/mknod ${LDIR}/ord${I}.dat p
      /etc/mknod ${LDIR}/ordline${I}.dat p

      tpccload -M $MULT -o ${LDIR}/ordline${I}.dat -g -b $SW -e $EW > \
        ${LDIR}/order${I}.dat 2> ${OUTDIR}/order${I}.out &

      sleep 5

      sqlldr tpcc/tpcc control=$TPCC_LOADER/order.ctl \
         log=${OUTDIR}/order${I}.log \
         bad=${OUTDIR}/order${I}.bad data=${LDIR}/order${I}.dat \
         discard=${OUTDIR}/order${I}.dsc &

      sqlldr tpcc/tpcc control=$TPCC_LOADER/ordline.ctl \
         log=${OUTDIR}/ordline${I}.log \
         bad=${OUTDIR}/ordline${I}.bad data=${LDIR}/ordline${I}.dat \
         discard=${OUTDIR}/ordline${I}.dsc &

fi

wait
I=1
while [ $I -le $NFF ]
do
  rm -f ${LDIR}/order${I}.dat
  rm -f ${LDIR}/ordline${I}.dat
  I=‘expr $I + 1‘
done

print '\n\n\nORDER and ORDER-LINE phase DONE!'
date
```

ploadstock.sh

```
#
˜
#=============================================================
========+
#       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#                  OPEN SYSTEMS PERFORMANCE GROUP                |
#                       All Rights Reserved                      |
#=============================================================
========+
# FILENAME
#      ploadstock.sh
# DESCRIPTION
#      Usage: ploadstock.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=4608

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

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 $LDIR ]
then
  mkdir $LDIR
fi

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



#
#  Load Stock table
#

print `date +%T`' Starting STOCK data load . . .'

NFILES=20
#
# Get no. of Items / FILE which can be loaded
IPFILE=`expr 100000 / $NFILES `

I=1
while [ $I -le $NFILES ]
do
```

```
  /etc/mknod ${LDIR}/stock${I}.dat p
  I=‘expr $I + 1‘
done

print ’    TPCCLOAD phase . . .’

I=1
SI=1
EI=$IPFILE
INC=$IPFILE
while [ $I -le $NFILES ]
do
  K=‘expr $I % 8‘
  print ’TPCCLOAD for file ‘$I’, starting at ‘$SI’, ending at ‘$EI
  tpccload -M $MULT -S -g -j $SI -k $EI > ${LDIR}/stock${I}.dat 2> \
    ${OUTDIR}/stock${I}.out &
  I=‘expr $I + 1‘
  SI=‘expr $SI + $INC‘
  EI=‘expr $EI + $INC‘
done

sleep 10

print ‘\n\n\n\n\n     SQLLDR phase:’
print ’        no. files:          ‘$NFILES’;’
print ’        Items per file: ‘$IPFILE’;’
print ’        warehouses:         ‘$MULT’;\n\n\n\n\n’

I=1
while [ $I -le $NFILES ]
do

  K=‘expr $I % 8‘
  sqlldr tpcc/tpcc control=$TPCC_LOADER/stock.ctl \
       log=${OUTDIR}/stock${I}.log \
       bad=${OUTDIR}/stock${I}.bad data=${LDIR}/stock${I}.dat \
       discard=${OUTDIR}/stock${I}.dsc &

  I=‘expr $I + 1‘
done

wait

print ‘date +%T‘’     STOCK load DONE!’
print ’’

I=1
while [ $I -le $NFILES ]
do
  rm -f ${LDIR}/stock${I}.dat
  I=‘expr $I + 1‘
done

switchlog.sh

print ’’
print ‘date +%T‘’End STOCK data load . . .’
```

tpcc_tab.sql

```
 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
                drop cluster icluster including tables;
                drop table warehouse;
                drop table district;
                drop table history;
                drop table orders;
                drop table new_order;
                drop table order_line;
                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 (577) tablespace ware_1 ,
                partition ware_P2 values less than (1153) tablespace ware_2 ,
                partition ware_P3 values less than (1729) tablespace ware_3 ,
                partition ware_P4 values less than (2305) tablespace ware_4 ,
                partition ware_P5 values less than (2881) tablespace ware_5 ,
                partition ware_P6 values less than (3457) tablespace ware_6 ,
                partition ware_P7 values less than (4033) tablespace ware_7 ,
                partition ware_P8 values less than (MAXVALUE) tablespace
ware_8
       )
       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 (577) tablespace ware_1 ,
                partition dist_P2 values less than (1153) tablespace ware_2 ,
                partition dist_P3 values less than (1729) tablespace ware_3 ,
                partition dist_P4 values less than (2305) tablespace ware_4 ,
                partition dist_P5 values less than (2881) tablespace ware_5 ,
                partition dist_P6 values less than (3457) tablespace ware_6 ,
                partition dist_P7 values less than (4033) tablespace ware_7 ,
                partition dist_P8 values less than (MAXVALUE) tablespace
ware_8
       )
       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 (577) tablespace hist_1 ,
            partition hist_P2 values less than (1153) tablespace hist_2 ,
            partition hist_P3 values less than (1729) tablespace hist_3 ,
            partition hist_P4 values less than (2305) tablespace hist_4 ,
            partition hist_P5 values less than (2881) tablespace hist_5 ,
            partition hist_P6 values less than (3457) tablespace hist_6 ,
            partition hist_P7 values less than (4033) tablespace hist_7 ,
            partition hist_P8 values less than (MAXVALUE) tablespace hist_8
    )
    initrans 3
    pctfree 1
    storage (initial 10K next 450M pctincrease 0 maxextents unlimited
        freelist groups 40 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 (577) tablespace nord_1 ,
            partition nord_P2 values less than (1153) tablespace nord_2 ,
            partition nord_P3 values less than (1729) tablespace nord_3 ,
            partition nord_P4 values less than (2305) tablespace nord_4 ,
            partition nord_P5 values less than (2881) tablespace nord_5 ,
            partition nord_P6 values less than (3457) tablespace nord_6 ,
            partition nord_P7 values less than (4033) tablespace nord_7 ,
            partition nord_P8 values less than (MAXVALUE) tablespace
nord_8
    )
    initrans 4
    pctfree 5
    storage (initial 10K next 125M pctincrease 0 maxextents unlimited
        freelist groups 40 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 (145) tablespace ord_1 ,
            partition order_P2 values less than (289) tablespace ord_2 ,
            partition order_P3 values less than (433) tablespace ord_3 ,
            partition order_P4 values less than (577) tablespace ord_4 ,
            partition order_P5 values less than (721) tablespace ord_5 ,
            partition order_P6 values less than (865) tablespace ord_6 ,
            partition order_P7 values less than (1009) tablespace ord_7 ,
            partition order_P8 values less than (1153) tablespace ord_8 ,
            partition order_P9 values less than (1297) tablespace ord_9 ,
            partition order_P10 values less than (1441) tablespace ord_10 ,
            partition order_P11 values less than (1585) tablespace ord_11 ,
            partition order_P12 values less than (1729) tablespace ord_12 ,
            partition order_P13 values less than (1873) tablespace ord_13 ,
            partition order_P14 values less than (2017) tablespace ord_14 ,
            partition order_P15 values less than (2161) tablespace ord_15 ,
            partition order_P16 values less than (2305) tablespace ord_16 ,
            partition order_P17 values less than (2449) tablespace ord_17 ,
            partition order_P18 values less than (2593) tablespace ord_18 ,
            partition order_P19 values less than (2737) tablespace ord_19 ,
            partition order_P20 values less than (2881) tablespace ord_20 ,
            partition order_P21 values less than (3025) tablespace ord_21 ,
            partition order_P22 values less than (3169) tablespace ord_22 ,
            partition order_P23 values less than (3313) tablespace ord_23 ,
            partition order_P24 values less than (3457) tablespace ord_24 ,
            partition order_P25 values less than (3601) tablespace ord_25 ,
            partition order_P26 values less than (3745) tablespace ord_26 ,
            partition order_P27 values less than (3889) tablespace ord_27 ,
            partition order_P28 values less than (4033) tablespace ord_28 ,
            partition order_P29 values less than (4177) tablespace ord_29 ,
            partition order_P30 values less than (4321) tablespace ord_30 ,
            partition order_P31 values less than (4465) tablespace ord_31 ,
            partition order_P32 values less than (MAXVALUE) tablespace
ord_32
    )
    initrans 3
    pctfree 5
    storage (initial 10K next 100M pctincrease 0 maxextents unlimited
        freelist groups 40 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 (145) tablespace ordl_1 ,
            partition ordl_P2 values less than (289) tablespace ordl_2 ,
            partition ordl_P3 values less than (433) tablespace ordl_3 ,
            partition ordl_P4 values less than (577) tablespace ordl_4 ,
            partition ordl_P5 values less than (721) tablespace ordl_5 ,
            partition ordl_P6 values less than (865) tablespace ordl_6 ,
            partition ordl_P7 values less than (1009) tablespace ordl_7 ,
            partition ordl_P8 values less than (1153) tablespace ordl_8 ,
            partition ordl_P9 values less than (1297) tablespace ordl_9 ,
            partition ordl_P10 values less than (1441) tablespace ordl_10 ,
            partition ordl_P11 values less than (1585) tablespace ordl_11 ,
            partition ordl_P12 values less than (1729) tablespace ordl_12 ,
            partition ordl_P13 values less than (1873) tablespace ordl_13 ,
            partition ordl_P14 values less than (2017) tablespace ordl_14 ,
            partition ordl_P15 values less than (2161) tablespace ordl_15 ,
            partition ordl_P16 values less than (2305) tablespace ordl_16 ,
            partition ordl_P17 values less than (2449) tablespace ordl_17 ,
            partition ordl_P18 values less than (2593) tablespace ordl_18 ,
            partition ordl_P19 values less than (2737) tablespace ordl_19 ,
            partition ordl_P20 values less than (2881) tablespace ordl_20 ,
            partition ordl_P21 values less than (3025) tablespace ordl_21 ,
            partition ordl_P22 values less than (3169) tablespace ordl_22 ,
            partition ordl_P23 values less than (3313) tablespace ordl_23 ,
            partition ordl_P24 values less than (3457) tablespace ordl_24 ,
            partition ordl_P25 values less than (3601) tablespace ordl_25 ,
            partition ordl_P26 values less than (3745) tablespace ordl_26 ,
            partition ordl_P27 values less than (3889) tablespace ordl_27 ,
            partition ordl_P28 values less than (4033) tablespace ordl_28 ,
            partition ordl_P29 values less than (4177) tablespace ordl_29 ,
            partition ordl_P30 values less than (4321) tablespace ordl_30 ,
            partition ordl_P31 values less than (4465) tablespace ordl_31 ,
            partition ordl_P32 values less than (MAXVALUE) tablespace
ordl_32
    )
    initrans 4
    pctfree 5
    storage (initial 10K next 1000M pctincrease 0 maxextents unlimited
        freelist groups 40 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 25M next 720K 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
drop cluster ccluster including tables;
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(4,0)
       )
       hashkeys        138240000
       hash is         (c_w_id * 30000 + c_d_id * 3000 + c_id - 33001)
       size            850
       initrans        3
       pctfree         0
       tablespace      cust
       storage (initial 270000K next 270000K pctincrease 0 minextents 512
                                    maxextents unlimited freelist groups 8 freelists 19 );

                   create table customer (
       c_id            number(5,0),
       c_d_id          number(2,0),
       c_w_id          number(4,0),
       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
drop cluster scluster including tables;
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(4,0)
       )
       hashkeys        460800000
       hash is         (abs(s_i_id - 1) * 576 + mod((s_w_id - 1), 576) +
                       trunc ((s_w_id - 1) / 576) * 57600000)
       size            350
       initrans        3
       pctfree         0
       tablespace      stocks
       storage (initial 360000K next 360000K pctincrease 0 minextents 512
                        maxextents unlimited  freelist groups 8 freelists 19 );

                   create table stock (
       s_i_id          number(6,0),
       s_w_id          number(4,0),
       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_ix1.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;

drop index iwarehouse;
drop index idistrict;
drop index icustomer;
drop index icustomer2;
drop index istock;
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 2000K 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 16
      storage (initial 100M next 100M 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 16
      storage (initial 200M next 200M pctincrease 0 maxextents unlimited ) pctfree
1;

create unique index istock on stock(s_i_id, s_w_id)
      tablespace istk
      initrans 3
      nologging
      parallel 24
      storage (initial 400M 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;
```

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_ix2.sql
rem  DESCRIPTION
rem       Create indexes for TPC-C database.
rem
=========================================================
========
rem
set echo on;

drop index iorders;
drop index iorders2;
drop index inew_order;
drop index iorder_line;

set timing on

create unique index iorders on orders(o_w_id, o_d_id, o_id)
      local
      (
                  partition order_P1 tablespace iord1_1 ,
                  partition order_P2 tablespace iord1_2 ,
                  partition order_P3 tablespace iord1_3 ,
                  partition order_P4 tablespace iord1_4 ,
                  partition order_P5 tablespace iord1_5 ,
                  partition order_P6 tablespace iord1_6 ,
                  partition order_P7 tablespace iord1_7 ,
                  partition order_P8 tablespace iord1_8 ,
                  partition order_P9 tablespace iord1_9 ,
                  partition order_P10 tablespace iord1_10 ,
                  partition order_P11 tablespace iord1_11 ,
                  partition order_P12 tablespace iord1_12 ,
                  partition order_P13 tablespace iord1_13 ,
                  partition order_P14 tablespace iord1_14 ,
                  partition order_P15 tablespace iord1_15 ,
                  partition order_P16 tablespace iord1_16 ,
                  partition order_P17 tablespace iord1_17 ,
                  partition order_P18 tablespace iord1_18 ,
                  partition order_P19 tablespace iord1_19 ,
                  partition order_P20 tablespace iord1_20 ,
                  partition order_P21 tablespace iord1_21 ,
                  partition order_P22 tablespace iord1_22 ,
                  partition order_P23 tablespace iord1_23 ,
                  partition order_P24 tablespace iord1_24 ,
                  partition order_P25 tablespace iord1_25 ,
                  partition order_P26 tablespace iord1_26 ,
                  partition order_P27 tablespace iord1_27 ,
                  partition order_P28 tablespace iord1_28 ,
                  partition order_P29 tablespace iord1_29 ,
                  partition order_P30 tablespace iord1_30 ,
                  partition order_P31 tablespace iord1_31 ,
                  partition order_P32 tablespace iord1_32
      )
      initrans 3
      nologging
      parallel 32
      pctfree  1
      storage (initial 20K next 75M pctincrease 0
            freelist groups 40 freelists 17 maxextents unlimited);

create unique index iorders2 on orders(o_w_id, o_d_id, o_c_id, o_id)
      local
      (
                  partition order_P1 tablespace iord2_1 ,
                  partition order_P2 tablespace iord2_2 ,
                  partition order_P3 tablespace iord2_3 ,
                  partition order_P4 tablespace iord2_4 ,
                  partition order_P5 tablespace iord2_5 ,
                  partition order_P6 tablespace iord2_6 ,
                  partition order_P7 tablespace iord2_7 ,
                  partition order_P8 tablespace iord2_8 ,
                  partition order_P9 tablespace iord2_9 ,
                  partition order_P10 tablespace iord2_10 ,
                  partition order_P11 tablespace iord2_11 ,
                  partition order_P12 tablespace iord2_12 ,
                  partition order_P13 tablespace iord2_13 ,
                  partition order_P14 tablespace iord2_14 ,
                  partition order_P15 tablespace iord2_15 ,
                  partition order_P16 tablespace iord2_16 ,
                  partition order_P17 tablespace iord2_17 ,
                  partition order_P18 tablespace iord2_18 ,
                  partition order_P19 tablespace iord2_19 ,
                  partition order_P20 tablespace iord2_20 ,
                  partition order_P21 tablespace iord2_21 ,
                  partition order_P22 tablespace iord2_22 ,
                  partition order_P23 tablespace iord2_23 ,
                  partition order_P24 tablespace iord2_24 ,
                  partition order_P25 tablespace iord2_25 ,
                  partition order_P26 tablespace iord2_26 ,
                  partition order_P27 tablespace iord2_27 ,
                  partition order_P28 tablespace iord2_28 ,
                  partition order_P29 tablespace iord2_29 ,
                  partition order_P30 tablespace iord2_30 ,
```

```
                partition order_P31 tablespace iord2_31 ,
                partition order_P32 tablespace iord2_32
        )
        initrans 3
        nologging
        parallel 32
        pctfree  25
        storage (initial 20K next 100M pctincrease 0
                freelist groups 40 freelists 17 maxextents unlimited);

create unique index inew_order on new_order(no_w_id, no_d_id, no_o_id)
        local
        (
        partition nord_P1  tablespace inord_1 ,
        partition nord_P2  tablespace inord_2 ,
        partition nord_P3  tablespace inord_3 ,
        partition nord_P4  tablespace inord_4 ,
        partition nord_P5  tablespace inord_5 ,
        partition nord_P6  tablespace inord_6 ,
        partition nord_P7  tablespace inord_7 ,
        partition nord_P8  tablespace inord_8
        )
        initrans 4
        nologging
        parallel 32
        pctfree  5
        storage (initial 20K next 100M pctincrease 0
                freelist groups 40 freelists 17 maxextents unlimited);

create unique index iorder_line on order_line(ol_w_id, ol_d_id, ol_o_id,
ol_number)
        local
        (
                partition ordl_P1 tablespace iordl_1 ,
                partition ordl_P2 tablespace iordl_2 ,
                partition ordl_P3 tablespace iordl_3 ,
                partition ordl_P4 tablespace iordl_4 ,
                partition ordl_P5 tablespace iordl_5 ,
                partition ordl_P6 tablespace iordl_6 ,
                partition ordl_P7 tablespace iordl_7 ,
                partition ordl_P8 tablespace iordl_8 ,
                partition ordl_P9 tablespace iordl_9 ,
                partition ordl_P10 tablespace iordl_10 ,
                partition ordl_P11 tablespace iordl_11 ,
                partition ordl_P12 tablespace iordl_12 ,
                partition ordl_P13 tablespace iordl_13 ,
                partition ordl_P14 tablespace iordl_14 ,
                partition ordl_P15 tablespace iordl_15 ,
                partition ordl_P16 tablespace iordl_16 ,
                partition ordl_P17 tablespace iordl_17 ,
                partition ordl_P18 tablespace iordl_18 ,
                partition ordl_P19 tablespace iordl_19 ,
                partition ordl_P20 tablespace iordl_20 ,
                partition ordl_P21 tablespace iordl_21 ,
                partition ordl_P22 tablespace iordl_22 ,
                partition ordl_P23 tablespace iordl_23 ,
                partition ordl_P24 tablespace iordl_24 ,
                partition ordl_P25 tablespace iordl_25 ,
                partition ordl_P26 tablespace iordl_26 ,
                partition ordl_P27 tablespace iordl_27 ,
                partition ordl_P28 tablespace iordl_28 ,
                partition ordl_P29 tablespace iordl_29 ,
                partition ordl_P30 tablespace iordl_30 ,
                partition ordl_P31 tablespace iordl_31 ,
                partition ordl_P32 tablespace iordl_32
        )
        initrans 4
        nologging
        parallel 32
        pctfree  1
        storage (initial 20K next 800M pctincrease 0
                freelist groups 40 freelists 17 maxextents unlimited);

alter index iorders     deallocate unused;
alter index iorders2    deallocate unused;
alter index inew_order  deallocate unused;
alter index iorder_line 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 -Z4 n=1000
˜
for q in 1 2 3 4 5 6 7 8
˜
do
˜
i=1
˜
(( m =  $q * 1000 ))
˜
while [ $i -le 210 ]
˜
do
˜
(( j = ( $i - 1 ) % 3 + 1 + (( $q - 1 ) * 3 )  ))
(( n = $i + $m ))
echo "\
  CREATE ROLLBACK SEGMENT t$n
     TABLESPACE roll_$j
     STORAGE (initial 100K next 100K minextents 2);
"
let i=i+1
done
done
) | $ORACLE_HOME/bin/svrmgrl
```

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                   = 1500
_lm_direct_sends                        = all
buffer_pool_recycle                     = (buffers:10000, lru_latches:2)
compatible                              = 8.0.4
control_files
                =(?/dbs/tpcc_disks/ctrl1,?/dbs/tpcc_disks/ctrl2)
cpu_count                               = 4
cursor_space_for_time                   = true
db_block_buffers            = 1400000
db_block_checkpoint_batch   = 1000
db_block_lru_latches                    = 9
db_block_max_dirty_target   = 0
db_block_size                           = 2048
db_files                                = 2000
db_name                                                 = tpcc
discrete_transactions_enabled           = false
distributed_transactions    = 0
dml_locks                               = 100
enqueue_resources                       = 2000
gc_defer_time                           = 10
gc_files_to_locks                       = "\
229-236,238-584,588,599,610,612-752,755,757-759,\
763,766,769,773,785-786,788,790,794=983each:\
1040-1277,228,753-754,756,760-762,764-765,767-768,\
770-772,774-784,787,789,791-793,795-1038=34each:\
2-188,190,194-201,585-587,589-598,600-609,611,1278-1301=1each:\
237,1302-1309=1each:\
1039=1:\
1=100"
gc_latches                              = 8
gc_releasable_locks                     = 20000
gc_rollback_locks                       = "0,31-1710=20each"
hash_join_enabled                       = false
lm_locks                                = 1100000
lm_procs                                = 220
lm_ress                                                 = 600000
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                      = 1000000
max_rollback_segments                   = 401
open_cursors                            = 200
parallel_max_servers                    = 0
parallel_min_message_pool   = 0
parallel_min_servers                    = 0
parallel_server                         = true
processes                               = 200
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
########################################
instance_number=1
thread=1
ifile=/usr/oracle/dbs/mult/init.ora
rollback_segments = (
t1001, t1002, t1003, t1004, t1005, t1006, t1007, t1008, t1009, t1010, \
t1011, t1012, t1013, t1014, t1015, t1016, t1017, t1018, t1019, t1020, \
t1021, t1022, t1023, t1024, t1025, t1026, t1027, t1028, t1029, t1030, \
t1031, t1032, t1033, t1034, t1035, t1036, t1037, t1038, t1039, t1040, \
t1041, t1042, t1043, t1044, t1045, t1046, t1047, t1048, t1049, t1050, \
t1051, t1052, t1053, t1054, t1055, t1056, t1057, t1058, t1059, t1060, \
t1061, t1062, t1063, t1064, t1065, t1066, t1067, t1068, t1069, t1070, \
t1071, t1072, t1073, t1074, t1075, t1076, t1077, t1078, t1079, t1080, \
t1081, t1082, t1083, t1084, t1085, t1086, t1087, t1088, t1089, t1090, \
t1091, t1092, t1093, t1094, t1095, t1096, t1097, t1098, t1099, t1100, \
t1101, t1102, t1103, t1104, t1105, t1106, t1107, t1108, t1109, t1110, \
t1111, t1112, t1113, t1114, t1115, t1116, t1117, t1118, t1119, t1120, \
t1121, t1122, t1123, t1124, t1125, t1126, t1127, t1128, t1129, t1130, \
t1131, t1132, t1133, t1134, t1135, t1136, t1137, t1138, t1139, t1140, \
t1141, t1142, t1143, t1144, t1145, t1146, t1147, t1148, t1149, t1150, \
t1151, t1152, t1153, t1154, t1155, t1156, t1157, t1158, t1159, t1160, \
t1161, t1162, t1163, t1164, t1165, t1166, t1167, t1168, t1169, t1170, \
t1171, t1172, t1173, t1174, t1175, t1176, t1177, t1178, t1179, t1180, \
t1181, t1182, t1183, t1184, t1185, t1186, t1187, t1188, t1189, t1190, \
t1191, t1192, t1193, t1194, t1195, t1196, t1197, t1198, t1199, t1200 )


########################################
##      init2.ora
########################################
instance_number=2
thread=2
ifile=/usr/oracle/dbs/mult/init.ora
rollback_segments = (
t2001, t2002, t2003, t2004, t2005, t2006, t2007, t2008, t2009, t2010, \
t2011, t2012, t2013, t2014, t2015, t2016, t2017, t2018, t2019, t2020, \
t2021, t2022, t2023, t2024, t2025, t2026, t2027, t2028, t2029, t2030, \
t2031, t2032, t2033, t2034, t2035, t2036, t2037, t2038, t2039, t2040, \
t2041, t2042, t2043, t2044, t2045, t2046, t2047, t2048, t2049, t2050, \
t2051, t2052, t2053, t2054, t2055, t2056, t2057, t2058, t2059, t2060, \
t2061, t2062, t2063, t2064, t2065, t2066, t2067, t2068, t2069, t2070, \
t2071, t2072, t2073, t2074, t2075, t2076, t2077, t2078, t2079, t2080, \
t2081, t2082, t2083, t2084, t2085, t2086, t2087, t2088, t2089, t2090, \
t2091, t2092, t2093, t2094, t2095, t2096, t2097, t2098, t2099, t2100, \
t2101, t2102, t2103, t2104, t2105, t2106, t2107, t2108, t2109, t2110, \
t2111, t2112, t2113, t2114, t2115, t2116, t2117, t2118, t2119, t2120, \
t2121, t2122, t2123, t2124, t2125, t2126, t2127, t2128, t2129, t2130, \
t2131, t2132, t2133, t2134, t2135, t2136, t2137, t2138, t2139, t2140, \
t2141, t2142, t2143, t2144, t2145, t2146, t2147, t2148, t2149, t2150, \
t2151, t2152, t2153, t2154, t2155, t2156, t2157, t2158, t2159, t2160, \
t2161, t2162, t2163, t2164, t2165, t2166, t2167, t2168, t2169, t2170, \
t2171, t2172, t2173, t2174, t2175, t2176, t2177, t2178, t2179, t2180, \
t2181, t2182, t2183, t2184, t2185, t2186, t2187, t2188, t2189, t2190, \
t2191, t2192, t2193, t2194, t2195, t2196, t2197, t2198, t2199, t2200 )


########################################
##      init3.ora
########################################
instance_number=3
thread=3
ifile=/usr/oracle/dbs/mult/init.ora
rollback_segments = (
t3001, t3002, t3003, t3004, t3005, t3006, t3007, t3008, t3009, t3010, \
t3011, t3012, t3013, t3014, t3015, t3016, t3017, t3018, t3019, t3020, \
t3021, t3022, t3023, t3024, t3025, t3026, t3027, t3028, t3029, t3030, \
t3031, t3032, t3033, t3034, t3035, t3036, t3037, t3038, t3039, t3040, \
t3041, t3042, t3043, t3044, t3045, t3046, t3047, t3048, t3049, t3050, \
t3051, t3052, t3053, t3054, t3055, t3056, t3057, t3058, t3059, t3060, \
t3061, t3062, t3063, t3064, t3065, t3066, t3067, t3068, t3069, t3070, \
t3071, t3072, t3073, t3074, t3075, t3076, t3077, t3078, t3079, t3080, \
t3081, t3082, t3083, t3084, t3085, t3086, t3087, t3088, t3089, t3090, \
t3091, t3092, t3093, t3094, t3095, t3096, t3097, t3098, t3099, t3100, \
t3101, t3102, t3103, t3104, t3105, t3106, t3107, t3108, t3109, t3110, \
t3111, t3112, t3113, t3114, t3115, t3116, t3117, t3118, t3119, t3120, \
t3121, t3122, t3123, t3124, t3125, t3126, t3127, t3128, t3129, t3130, \
t3131, t3132, t3133, t3134, t3135, t3136, t3137, t3138, t3139, t3140, \
t3141, t3142, t3143, t3144, t3145, t3146, t3147, t3148, t3149, t3150, \
t3151, t3152, t3153, t3154, t3155, t3156, t3157, t3158, t3159, t3160, \
t3161, t3162, t3163, t3164, t3165, t3166, t3167, t3168, t3169, t3170, \
t3171, t3172, t3173, t3174, t3175, t3176, t3177, t3178, t3179, t3180, \
t3181, t3182, t3183, t3184, t3185, t3186, t3187, t3188, t3189, t3190, \
t3191, t3192, t3193, t3194, t3195, t3196, t3197, t3198, t3199, t3200)


########################################
```

```
##      init4.ora
########################################
instance_number=4
thread=4
ifile=/usr/oracle/dbs/mult/init.ora
rollback_segments = (
t4001, t4002, t4003, t4004, t4005, t4006, t4007, t4008, t4009, t4010, \
t4011, t4012, t4013, t4014, t4015, t4016, t4017, t4018, t4019, t4020, \
t4021, t4022, t4023, t4024, t4025, t4026, t4027, t4028, t4029, t4030, \
t4031, t4032, t4033, t4034, t4035, t4036, t4037, t4038, t4039, t4040, \
t4041, t4042, t4043, t4044, t4045, t4046, t4047, t4048, t4049, t4050, \
t4051, t4052, t4053, t4054, t4055, t4056, t4057, t4058, t4059, t4060, \
t4061, t4062, t4063, t4064, t4065, t4066, t4067, t4068, t4069, t4070, \
t4071, t4072, t4073, t4074, t4075, t4076, t4077, t4078, t4079, t4080, \
t4081, t4082, t4083, t4084, t4085, t4086, t4087, t4088, t4089, t4090, \
t4091, t4092, t4093, t4094, t4095, t4096, t4097, t4098, t4099, t4100, \
t4101, t4102, t4103, t4104, t4105, t4106, t4107, t4108, t4109, t4110, \
t4111, t4112, t4113, t4114, t4115, t4116, t4117, t4118, t4119, t4120, \
t4121, t4122, t4123, t4124, t4125, t4126, t4127, t4128, t4129, t4130, \
t4131, t4132, t4133, t4134, t4135, t4136, t4137, t4138, t4139, t4140, \
t4141, t4142, t4143, t4144, t4145, t4146, t4147, t4148, t4149, t4150, \
t4151, t4152, t4153, t4154, t4155, t4156, t4157, t4158, t4159, t4160, \
t4161, t4162, t4163, t4164, t4165, t4166, t4167, t4168, t4169, t4170, \
t4171, t4172, t4173, t4174, t4175, t4176, t4177, t4178, t4179, t4180, \
t4181, t4182, t4183, t4184, t4185, t4186, t4187, t4188, t4189, t4190, \
t4191, t4192, t4193, t4194, t4195, t4196, t4197, t4198, t4199, t4200 )
```

```
########################################
##      init5.ora
########################################
instance_number=5
thread=5
ifile=/usr/oracle/dbs/mult/init.ora
rollback_segments = (
t5001, t5002, t5003, t5004, t5005, t5006, t5007, t5008, t5009, t5010, \
t5011, t5012, t5013, t5014, t5015, t5016, t5017, t5018, t5019, t5020, \
t5021, t5022, t5023, t5024, t5025, t5026, t5027, t5028, t5029, t5030, \
t5031, t5032, t5033, t5034, t5035, t5036, t5037, t5038, t5039, t5040, \
t5041, t5042, t5043, t5044, t5045, t5046, t5047, t5048, t5049, t5050, \
t5051, t5052, t5053, t5054, t5055, t5056, t5057, t5058, t5059, t5060, \
t5061, t5062, t5063, t5064, t5065, t5066, t5067, t5068, t5069, t5070, \
t5071, t5072, t5073, t5074, t5075, t5076, t5077, t5078, t5079, t5080, \
t5081, t5082, t5083, t5084, t5085, t5086, t5087, t5088, t5089, t5090, \
t5091, t5092, t5093, t5094, t5095, t5096, t5097, t5098, t5099, t5100, \
t5101, t5102, t5103, t5104, t5105, t5106, t5107, t5108, t5109, t5110, \
t5111, t5112, t5113, t5114, t5115, t5116, t5117, t5118, t5119, t5120, \
t5121, t5122, t5123, t5124, t5125, t5126, t5127, t5128, t5129, t5130, \
t5131, t5132, t5133, t5134, t5135, t5136, t5137, t5138, t5139, t5140, \
t5141, t5142, t5143, t5144, t5145, t5146, t5147, t5148, t5149, t5150, \
t5151, t5152, t5153, t5154, t5155, t5156, t5157, t5158, t5159, t5160, \
t5161, t5162, t5163, t5164, t5165, t5166, t5167, t5168, t5169, t5170, \
t5171, t5172, t5173, t5174, t5175, t5176, t5177, t5178, t5179, t5180, \
t5181, t5182, t5183, t5184, t5185, t5186, t5187, t5188, t5189, t5190, \
t5191, t5192, t5193, t5194, t5195, t5196, t5197, t5198, t5199, t5200 )
```

```
########################################
##      init6.ora
########################################
instance_number=6
thread=6
ifile=/usr/oracle/dbs/mult/init.ora
rollback_segments = (
t6001, t6002, t6003, t6004, t6005, t6006, t6007, t6008, t6009, t6010, \
t6011, t6012, t6013, t6014, t6015, t6016, t6017, t6018, t6019, t6020, \
t6021, t6022, t6023, t6024, t6025, t6026, t6027, t6028, t6029, t6030, \
t6031, t6032, t6033, t6034, t6035, t6036, t6037, t6038, t6039, t6040, \
t6041, t6042, t6043, t6044, t6045, t6046, t6047, t6048, t6049, t6050, \
t6051, t6052, t6053, t6054, t6055, t6056, t6057, t6058, t6059, t6060, \
t6061, t6062, t6063, t6064, t6065, t6066, t6067, t6068, t6069, t6070, \
t6071, t6072, t6073, t6074, t6075, t6076, t6077, t6078, t6079, t6080, \
t6081, t6082, t6083, t6084, t6085, t6086, t6087, t6088, t6089, t6090, \
t6091, t6092, t6093, t6094, t6095, t6096, t6097, t6098, t6099, t6100, \
t6101, t6102, t6103, t6104, t6105, t6106, t6107, t6108, t6109, t6110, \
t6111, t6112, t6113, t6114, t6115, t6116, t6117, t6118, t6119, t6120, \
t6121, t6122, t6123, t6124, t6125, t6126, t6127, t6128, t6129, t6130, \
t6131, t6132, t6133, t6134, t6135, t6136, t6137, t6138, t6139, t6140, \
t6141, t6142, t6143, t6144, t6145, t6146, t6147, t6148, t6149, t6150, \
t6151, t6152, t6153, t6154, t6155, t6156, t6157, t6158, t6159, t6160, \
t6161, t6162, t6163, t6164, t6165, t6166, t6167, t6168, t6169, t6170, \
t6171, t6172, t6173, t6174, t6175, t6176, t6177, t6178, t6179, t6180, \
t6181, t6182, t6183, t6184, t6185, t6186, t6187, t6188, t6189, t6190, \
t6191, t6192, t6193, t6194, t6195, t6196, t6197, t6198, t6199, t6200 )
```

```
########################################
##      init7.ora
########################################
instance_number=7
thread=7
ifile=/usr/oracle/dbs/mult/init.ora
rollback_segments = (
t7001, t7002, t7003, t7004, t7005, t7006, t7007, t7008, t7009, t7010, \
t7011, t7012, t7013, t7014, t7015, t7016, t7017, t7018, t7019, t7020, \
t7021, t7022, t7023, t7024, t7025, t7026, t7027, t7028, t7029, t7030, \
t7031, t7032, t7033, t7034, t7035, t7036, t7037, t7038, t7039, t7040, \
t7041, t7042, t7043, t7044, t7045, t7046, t7047, t7048, t7049, t7050, \
t7051, t7052, t7053, t7054, t7055, t7056, t7057, t7058, t7059, t7060, \
t7061, t7062, t7063, t7064, t7065, t7066, t7067, t7068, t7069, t7070, \
t7071, t7072, t7073, t7074, t7075, t7076, t7077, t7078, t7079, t7080, \
t7081, t7082, t7083, t7084, t7085, t7086, t7087, t7088, t7089, t7090, \
t7091, t7092, t7093, t7094, t7095, t7096, t7097, t7098, t7099, t7100, \
t7101, t7102, t7103, t7104, t7105, t7106, t7107, t7108, t7109, t7110, \
t7111, t7112, t7113, t7114, t7115, t7116, t7117, t7118, t7119, t7120, \
t7121, t7122, t7123, t7124, t7125, t7126, t7127, t7128, t7129, t7130, \
t7131, t7132, t7133, t7134, t7135, t7136, t7137, t7138, t7139, t7140, \
t7141, t7142, t7143, t7144, t7145, t7146, t7147, t7148, t7149, t7150, \
t7151, t7152, t7153, t7154, t7155, t7156, t7157, t7158, t7159, t7160, \
t7161, t7162, t7163, t7164, t7165, t7166, t7167, t7168, t7169, t7170, \
t7171, t7172, t7173, t7174, t7175, t7176, t7177, t7178, t7179, t7180, \
t7181, t7182, t7183, t7184, t7185, t7186, t7187, t7188, t7189, t7190, \
t7191, t7192, t7193, t7194, t7195, t7196, t7197, t7198, t7199, t7200 )
```

```
########################################
##      init8.ora
########################################
instance_number=8
thread=8
ifile=/usr/oracle/dbs/mult/init.ora
rollback_segments = (
t8001, t8002, t8003, t8004, t8005, t8006, t8007, t8008, t8009, t8010, \
t8011, t8012, t8013, t8014, t8015, t8016, t8017, t8018, t8019, t8020, \
t8021, t8022, t8023, t8024, t8025, t8026, t8027, t8028, t8029, t8030, \
t8031, t8032, t8033, t8034, t8035, t8036, t8037, t8038, t8039, t8040, \
t8041, t8042, t8043, t8044, t8045, t8046, t8047, t8048, t8049, t8050, \
t8051, t8052, t8053, t8054, t8055, t8056, t8057, t8058, t8059, t8060, \
t8061, t8062, t8063, t8064, t8065, t8066, t8067, t8068, t8069, t8070, \
t8071, t8072, t8073, t8074, t8075, t8076, t8077, t8078, t8079, t8080, \
t8081, t8082, t8083, t8084, t8085, t8086, t8087, t8088, t8089, t8090, \
t8091, t8092, t8093, t8094, t8095, t8096, t8097, t8098, t8099, t8100, \
t8101, t8102, t8103, t8104, t8105, t8106, t8107, t8108, t8109, t8110, \
t8111, t8112, t8113, t8114, t8115, t8116, t8117, t8118, t8119, t8120, \
t8121, t8122, t8123, t8124, t8125, t8126, t8127, t8128, t8129, t8130, \
t8131, t8132, t8133, t8134, t8135, t8136, t8137, t8138, t8139, t8140, \
t8141, t8142, t8143, t8144, t8145, t8146, t8147, t8148, t8149, t8150, \
t8151, t8152, t8153, t8154, t8155, t8156, t8157, t8158, t8159, t8160, \
t8161, t8162, t8163, t8164, t8165, t8166, t8167, t8168, t8169, t8170, \
t8171, t8172, t8173, t8174, t8175, t8176, t8177, t8178, t8179, t8180, \
t8181, t8182, t8183, t8184, t8185, t8186, t8187, t8188, t8189, t8190, \
t8191, t8192, t8193, t8194, t8195, t8196, t8197, t8198, t8199, t8200 )
```

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)
*
   6200         buffers in buffer cache (NBUF)
*
   2304         inodes  (NINODE)
 896000         entries in file table (NFILE)
    128         entries in mount table (NMOUNT)
    868         entries in proc table (NPROC)
    868         processes per user id (MAXUP)
      0         hash slots for buffer cache (NHBUF)
      0         buffers for physical I/O (NPBUF)
```

```
    38016       number of streams queues (NQUEUE)
     5520       number of streams head structures (NSTREAM)
       10       number of private stream buffers (NBLKPRIV)
     3090       number of 4096 bytes stream buffers (NBLK4096)
     6982       number of 2048 bytes stream buffers (NBLK2048)
     1122       number of 1024 bytes stream buffers (NBLK1024)
     1122       number of 512 bytes stream buffers (NBLK512)
    23000       number of 256 bytes stream buffers (NBLK256)
    33560       number of 128 bytes stream buffers (NBLK128)
     9653       number of 64 bytes stream buffers (NBLK64)
    16802       number of 16 bytes stream buffers (NBLK16)
    73318       number of 4 bytes stream buffers (NBLK4)
    60002       maximum user and group id allowed (C_MAXUID)
       20       number of symbolic link traversals (MAXSYMLINKS)
      345       entries in mapped file table (NMFILE)
     3072       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)
     8192       max #of pending async I/O request per process (MAXAIO)
     8192       #of async I/O buffers, 0-disables async I/O (NABUF)
      868       #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
*
    30080       entries in disk quota table (NDQUOT)
    10026       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)
       32       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
*
     1152       filename cache entries (NCSIZE)
      288       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
*
     5520       number of multiplexor links (NMUXLINK)
        6       maximum number of pushes allowed (NSTRPUSH)
    38016       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)
     4140       maximum #of permanent streams messages (NPERM)
*
* IPC Messages
*
      100       entries in msg map (MSGMAP)
     8192       max message size (MSGMAX)
    16384       max bytes on queue (MSGMNB)
       50       message queue identifiers (MSGMNI)
        8       message segment size (MSGSSZ)
       40       system message headers (MSGTQL)
     1024       message segments (MSGSEG)
*
* IPC Semaphores
*
      750       entries in semaphore map (SEMMAP)
     1100       semaphore identifiers (SEMMNI)
     2500       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
*
    55952       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 quad8, shared 4028.0Mb, private 4.0Mb
 MEM        1   on quad9, shared 4092.0Mb, private 4.0Mb
 MEM        2   on quad10, shared 4092.0Mb, private 4.0Mb
 MEM        3   on quad11, shared 4092.0Mb, private 4.0Mb
 MEM        4   on quad12, shared 4092.0Mb, private 4.0Mb
 MEM        5   on quad13, shared 4092.0Mb, private 4.0Mb
 MEM        6   on quad14, shared 4092.0Mb, private 4.0Mb
 MEM        7   on quad15, shared 4092.0Mb, private 4.0Mb
 QLC       10   on quad8, sw ver 4.40
 QLC       11   on quad8, sw ver 4.40
 QLC       12   on quad8, sw ver 4.40
 QLC       13   on quad9, sw ver 4.40
 QLC       14   on quad9, sw ver 4.40
 QLC       15   on quad10, sw ver 4.40
 QLC       16   on quad10, sw ver 4.40
 QLC       17   on quad11, sw ver 4.40
 QLC       18   on quad11, sw ver 4.40
 QLC       19   on quad12, sw ver 4.40
 QLC       20   on quad12, sw ver 4.40
 QLC       21   on quad13, sw ver 4.40
 QLC       22   on quad13, sw ver 4.40
 QLC       23   on quad14, sw ver 4.40
 QLC       24   on quad14, sw ver 4.40
 QLC       25   on quad15, sw ver 4.40
 QLC       26   on quad15, sw ver 4.40
 PE         0   on quad8, hw ver 2.2
 PE         1   on quad8, hw ver 2.1
 PE        11   on quad8, hw ver 2.2
 PE        12   on quad8, hw ver 2.2
 PE        13   on quad8, hw ver 2.2
 PE         2   on quad9, hw ver 2.1
 PE         3   on quad10, hw ver 2.1
 PE         4   on quad11, hw ver 2.1
 PE         5   on quad12, hw ver 2.1
 PE         6   on quad13, hw ver 2.1
 PE         7   on quad14, hw ver 2.1
 PE         8   on quad15, hw ver 2.1
 FF        32   on quad8, hw ver 4, sw ver FF2.12 0
 FF        33   on quad8, hw ver 4, sw ver FF2.12 0
 FF        34   on quad9, hw ver 4, sw ver FF2.12A2
 FF        35   on quad9, hw ver 4, sw ver FF2.12A2
 FF        36   on quad10, hw ver 4, sw ver FF2.12A2
 FF        37   on quad10, hw ver 4, sw ver FF2.12A2
 FF        38   on quad11, hw ver 4, sw ver FF2.12A2
 FF        39   on quad11, hw ver 4, sw ver FF2.12A2
 FF        40   on quad12, hw ver 4, sw ver FF2.12A2
 FF        41   on quad12, hw ver 4, sw ver FF2.12A2
 FF        42   on quad13, hw ver 4, sw ver FF2.12A2
 FF        43   on quad13, hw ver 4, sw ver FF2.12A2
 FF        44   on quad14, hw ver 4, sw ver FF2.12A2
 FF        45   on quad14, hw ver 4, sw ver FF2.12A2
 FF        46   on quad15, hw ver 4, sw ver FF2.12 0
 FF        47   on quad15, hw ver 4, sw ver FF2.12 0
 PROC/P6    0   on quad8, 405MHz, Apic: 1
 PROC/P6    1   on quad8, 405MHz, Apic: 8
 PROC/P6    2   on quad8, 405MHz, Apic: 2
 PROC/P6    3   on quad8, 405MHz, Apic: 4
 PROC/P6    4   on quad9, 405MHz, Apic: 17
 PROC/P6    5   on quad9, 405MHz, Apic: 24
 PROC/P6    6   on quad9, 405MHz, Apic: 18
 PROC/P6    7   on quad9, 405MHz, Apic: 20
 PROC/P6    8   on quad10, 405MHz, Apic: 33
 PROC/P6    9   on quad10, 405MHz, Apic: 40
 PROC/P6   10   on quad10, 405MHz, Apic: 34
 PROC/P6   11   on quad10, 405MHz, Apic: 36
 PROC/P6   12   on quad11, 405MHz, Apic: 49
 PROC/P6   13   on quad11, 405MHz, Apic: 56
 PROC/P6   14   on quad11, 405MHz, Apic: 50
 PROC/P6   15   on quad11, 405MHz, Apic: 52
 PROC/P6   16   on quad12, 405MHz, Apic: 65
 PROC/P6   17   on quad12, 405MHz, Apic: 72
 PROC/P6   18   on quad12, 405MHz, Apic: 66
 PROC/P6   19   on quad12, 405MHz, Apic: 68
 PROC/P6   20   on quad13, 405MHz, Apic: 81
 PROC/P6   21   on quad13, 405MHz, Apic: 88
 PROC/P6   22   on quad13, 405MHz, Apic: 82
 PROC/P6   23   on quad13, 405MHz, Apic: 84
 PROC/P6   24   on quad14, 405MHz, Apic: 97
 PROC/P6   25   on quad14, 405MHz, Apic: 104
 PROC/P6   26   on quad14, 405MHz, Apic: 98
 PROC/P6   27   on quad14, 405MHz, Apic: 100
 PROC/P6   28   on quad15, 405MHz, Apic: 113
 PROC/P6   29   on quad15, 405MHz, Apic: 120
 PROC/P6   30   on quad15, 405MHz, Apic: 114
 PROC/P6   31   on quad15, 405MHz, Apic: 116
nodeid: 0xfacade
masterid: none
```

Disk Configuration Dump

```
NAME       CFGTYPE  DEVNUM  UNIT       FLAGS  OnBUS    OnDEVICE
quad0      quad        4  0x00000000  L      sci      sci1
asy2       asy         2  0x00000000  L      eisa     quad0
asy3       asy         3  0x00000001  L      eisa     quad0
mdc1       mdc         1  0x00000000  L      eisa     quad0
ff0        ff         25  0x00000000  SP     pci      quad0
fabric0    fabric     10  0x00fe7800  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
```

Appendix C – Tunable Parameters

```
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
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      554 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
```

Appendix C – Tunable Parameters

```
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
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
```

Appendix C – Tunable Parameters

```
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
fcbr6     fcbr      6  0x00fe3000 SP   fcp    fabric0
scsibus48 scsibus  48  0x00000070 SM   mscsi  fcbr6
pbay48    pbay     48  0x00000100 S    scsi   scsibus48
sd3288    sd      289  0x00000000 S    scsi   scsibus48
sd3289    sd      290  0x00000010 S    scsi   scsibus48
sd3290    sd      291  0x00000020 S    scsi   scsibus48
sd3291    sd      292  0x00000030 S    scsi   scsibus48
sd3292    sd      293  0x00000040 S    scsi   scsibus48
sd3293    sd      294  0x00000050 S    scsi   scsibus48
scsibus49 scsibus  49  0x01000070 SM   mscsi  fcbr6
pbay49    pbay     49  0x00000100 S    scsi   scsibus49
sd3294    sd      295  0x00000000 S    scsi   scsibus49
sd3295    sd      296  0x00000010 S    scsi   scsibus49
sd3296    sd      297  0x00000020 S    scsi   scsibus49
sd3297    sd      298  0x00000030 S    scsi   scsibus49
sd3298    sd      299  0x00000040 S    scsi   scsibus49
sd3299    sd      300  0x00000050 S    scsi   scsibus49
scsibus50 scsibus  50  0x02000060 SM   mscsi  fcbr6
pbay50    pbay     50  0x00000100 S    scsi   scsibus50
sd3300    sd      301  0x00000000 S    scsi   scsibus50
sd3301    sd      302  0x00000010 S    scsi   scsibus50
sd3302    sd      303  0x00000020 S    scsi   scsibus50
sd3303    sd      304  0x00000030 S    scsi   scsibus50
sd3304    sd      305  0x00000040 S    scsi   scsibus50
sd3305    sd      306  0x00000050 S    scsi   scsibus50
scsibus51 scsibus  51  0x03000060 SM   mscsi  fcbr6
pbay51    pbay     51  0x00000100 S    scsi   scsibus51
sd3306    sd      307  0x00000000 S    scsi   scsibus51
sd3307    sd      308  0x00000010 S    scsi   scsibus51
sd3308    sd      309  0x00000020 S    scsi   scsibus51
sd3309    sd      310  0x00000030 S    scsi   scsibus51
sd3310    sd      311  0x00000040 S    scsi   scsibus51
sd3311    sd      312  0x00000050 S    scsi   scsibus51
scsibus52 scsibus  52  0x04000070 SM   mscsi  fcbr6
pbay52    pbay     52  0x00000100 S    scsi   scsibus52
sd3312    sd      313  0x00000000 S    scsi   scsibus52
sd3313    sd      314  0x00000010 S    scsi   scsibus52
sd3314    sd      315  0x00000020 S    scsi   scsibus52
sd3315    sd      316  0x00000030 S    scsi   scsibus52
sd3316    sd      317  0x00000040 S    scsi   scsibus52
sd3317    sd      318  0x00000050 S    scsi   scsibus52
scsibus53 scsibus  53  0x05000070 SM   mscsi  fcbr6
pbay53    pbay     53  0x00000100 S    scsi   scsibus53
sd3318    sd      319  0x00000000 S    scsi   scsibus53
sd3319    sd      320  0x00000010 S    scsi   scsibus53
sd3320    sd      321  0x00000020 S    scsi   scsibus53
sd3321    sd      322  0x00000030 S    scsi   scsibus53
sd3322    sd      323  0x00000040 S    scsi   scsibus53
sd3323    sd      324  0x00000050 S    scsi   scsibus53
scsibus54 scsibus  54  0x06000060 SM   mscsi  fcbr6
pbay54    pbay     54  0x00000100 S    scsi   scsibus54
sd3324    sd      325  0x00000000 S    scsi   scsibus54
sd3325    sd      326  0x00000010 S    scsi   scsibus54
sd3326    sd      327  0x00000020 S    scsi   scsibus54
sd3327    sd      328  0x00000030 S    scsi   scsibus54
sd3328    sd      329  0x00000040 S    scsi   scsibus54
sd3329    sd      330  0x00000050 S    scsi   scsibus54
scsibus55 scsibus  55  0x07000060 SM   mscsi  fcbr6
pbay55    pbay     55  0x00000100 S    scsi   scsibus55
sd3330    sd      331  0x00000000 S    scsi   scsibus55
sd3331    sd      332  0x00000010 S    scsi   scsibus55
sd3332    sd      333  0x00000020 S    scsi   scsibus55
sd3333    sd      334  0x00000030 S    scsi   scsibus55
sd3334    sd      335  0x00000040 S    scsi   scsibus55
sd3335    sd      336  0x00000050 S    scsi   scsibus55
ff1       ff        9  0x00000006 SP   pci    quad0
fabric1   fabric    9  0x00fe4000 SM   fc     ff1
fcbr7     fcbr      7  0x00fe0000 SP   fcp    fabric1
scsibus56 scsibus  56  0x00000070 SM   mscsi  fcbr7
pbay56    pbay     56  0x00000100 S    scsi   scsibus56
sd3336    sd      337  0x00000000 S    scsi   scsibus56
sd3337    sd      338  0x00000010 S    scsi   scsibus56
sd3338    sd      339  0x00000020 S    scsi   scsibus56
sd3339    sd      340  0x00000030 S    scsi   scsibus56
sd3340    sd      341  0x00000040 S    scsi   scsibus56
sd3341    sd      342  0x00000050 S    scsi   scsibus56
scsibus57 scsibus  57  0x01000070 SM   mscsi  fcbr7
pbay57    pbay     57  0x00000100 S    scsi   scsibus57
sd3342    sd      343  0x00000000 S    scsi   scsibus57
sd3343    sd      344  0x00000010 S    scsi   scsibus57
sd3344    sd      345  0x00000020 S    scsi   scsibus57
sd3345    sd      346  0x00000030 S    scsi   scsibus57
sd3346    sd      347  0x00000040 S    scsi   scsibus57
sd3347    sd      348  0x00000050 S    scsi   scsibus57
scsibus58 scsibus  58  0x02000060 SM   mscsi  fcbr7
pbay58    pbay     58  0x00000100 S    scsi   scsibus58
sd3348    sd      349  0x00000000 S    scsi   scsibus58
sd3349    sd      350  0x00000010 S    scsi   scsibus58
sd3350    sd      351  0x00000020 S    scsi   scsibus58
sd3351    sd      352  0x00000030 S    scsi   scsibus58
sd3352    sd      353  0x00000040 S    scsi   scsibus58
sd3353    sd      354  0x00000050 S    scsi   scsibus58
scsibus59 scsibus  59  0x03000060 SM   mscsi  fcbr7
pbay59    pbay     59  0x00000100 S    scsi   scsibus59
sd3354    sd      355  0x00000000 S    scsi   scsibus59
sd3355    sd      356  0x00000010 S    scsi   scsibus59
sd3356    sd      357  0x00000020 S    scsi   scsibus59
sd3357    sd      358  0x00000030 S    scsi   scsibus59
sd3358    sd      359  0x00000040 S    scsi   scsibus59
sd3359    sd      360  0x00000050 S    scsi   scsibus59
scsibus60 scsibus  60  0x04000070 SM   mscsi  fcbr7
pbay60    pbay     60  0x00000100 S    scsi   scsibus60
sd3360    sd      361  0x00000000 S    scsi   scsibus60
sd3361    sd      362  0x00000010 S    scsi   scsibus60
sd3362    sd      363  0x00000020 S    scsi   scsibus60
sd3363    sd      364  0x00000030 S    scsi   scsibus60
sd3364    sd      365  0x00000040 S    scsi   scsibus60
sd3365    sd      366  0x00000050 S    scsi   scsibus60
scsibus61 scsibus  61  0x05000070 SM   mscsi  fcbr7
pbay61    pbay     61  0x00000100 S    scsi   scsibus61
sd3366    sd      367  0x00000000 S    scsi   scsibus61
sd3367    sd      368  0x00000010 S    scsi   scsibus61
sd3368    sd      369  0x00000020 S    scsi   scsibus61
sd3369    sd      370  0x00000030 S    scsi   scsibus61
sd3370    sd      371  0x00000040 S    scsi   scsibus61
sd3371    sd      372  0x00000050 S    scsi   scsibus61
scsibus62 scsibus  62  0x06000060 SM   mscsi  fcbr7
pbay62    pbay     62  0x00000100 S    scsi   scsibus62
sd3372    sd      373  0x00000000 S    scsi   scsibus62
sd3373    sd      374  0x00000010 S    scsi   scsibus62
sd3374    sd      375  0x00000020 S    scsi   scsibus62
sd3375    sd      376  0x00000030 S    scsi   scsibus62
sd3376    sd      377  0x00000040 S    scsi   scsibus62
sd3377    sd      378  0x00000050 S    scsi   scsibus62
scsibus63 scsibus  63  0x07000060 SM   mscsi  fcbr7
pbay63    pbay     63  0x00000100 S    scsi   scsibus63
sd3378    sd      379  0x00000000 S    scsi   scsibus63
sd3379    sd      380  0x00000010 S    scsi   scsibus63
sd3380    sd      381  0x00000020 S    scsi   scsibus63
sd3381    sd      382  0x00000030 S    scsi   scsibus63
sd3382    sd      383  0x00000040 S    scsi   scsibus63
sd3383    sd      384  0x00000050 S    scsi   scsibus63
fcbr8     fcbr     14  0x00fe0800 SP   fcp    fabric1
scsibus130 scsibus 130  0x00000070 SM   mscsi  fcbr8
pbay122   pbay    122  0x00000100 S    scsi   scsibus130
sd3402    sd      403  0x00000000 S    scsi   scsibus130
sd3403    sd      404  0x00000010 S    scsi   scsibus130
sd3404    sd      405  0x00000020 S    scsi   scsibus130
sd3405    sd      406  0x00000030 S    scsi   scsibus130
sd3406    sd      407  0x00000040 S    scsi   scsibus130
sd3407    sd      408  0x00000050 S    scsi   scsibus130
scsibus131 scsibus 131  0x01000070 SM   mscsi  fcbr8
pbay123   pbay    123  0x00000100 S    scsi   scsibus131
sd3420    sd      421  0x00000000 S    scsi   scsibus131
sd3421    sd      422  0x00000010 S    scsi   scsibus131
sd3422    sd      423  0x00000020 S    scsi   scsibus131
sd3423    sd      424  0x00000030 S    scsi   scsibus131
sd3424    sd      425  0x00000040 S    scsi   scsibus131
sd3425    sd      426  0x00000050 S    scsi   scsibus131
scsibus132 scsibus 132  0x02000060 SM   mscsi  fcbr8
pbay124   pbay    124  0x00000100 S    scsi   scsibus132
sd3390    sd      391  0x00000000 S    scsi   scsibus132
sd3391    sd      392  0x00000010 S    scsi   scsibus132
sd3392    sd      393  0x00000020 S    scsi   scsibus132
sd3393    sd      394  0x00000030 S    scsi   scsibus132
sd3394    sd      395  0x00000040 S    scsi   scsibus132
sd3395    sd      396  0x00000050 S    scsi   scsibus132
scsibus133 scsibus 133  0x03000060 SM   mscsi  fcbr8
pbay125   pbay    125  0x00000100 S    scsi   scsibus133
sd3408    sd      409  0x00000000 S    scsi   scsibus133
sd3409    sd      410  0x00000010 S    scsi   scsibus133
sd3410    sd      411  0x00000020 S    scsi   scsibus133
sd3411    sd      412  0x00000030 S    scsi   scsibus133
sd3412    sd      413  0x00000040 S    scsi   scsibus133
sd3413    sd      414  0x00000050 S    scsi   scsibus133
scsibus134 scsibus 134  0x04000070 SM   mscsi  fcbr8
pbay126   pbay    126  0x00000100 S    scsi   scsibus134
sd3426    sd      427  0x00000000 S    scsi   scsibus134
sd3427    sd      428  0x00000010 S    scsi   scsibus134
sd3428    sd      429  0x00000020 S    scsi   scsibus134
sd3429    sd      430  0x00000030 S    scsi   scsibus134
sd3430    sd      684  0x00000040 S    scsi   scsibus134
sd3431    sd      432  0x00000050 S    scsi   scsibus134
scsibus135 scsibus 135  0x05000070 SM   mscsi  fcbr8
```

Appendix C – Tunable Parameters

```
pbay127    pbay     127 0x00000100 S    scsi    scsibus135
sd3396     sd       397 0x00000000 S    scsi    scsibus135
sd3397     sd       398 0x00000010 S    scsi    scsibus135
sd3398     sd       399 0x00000020 S    scsi    scsibus135
sd3399     sd       400 0x00000030 S    scsi    scsibus135
sd3400     sd       401 0x00000040 S    scsi    scsibus135
sd3401     sd       402 0x00000050 S    scsi    scsibus135
scsibus136 scsibus  136 0x06000060 SM   mscsi   fcbr8
pbay128    pbay     128 0x00000100 S    scsi    scsibus136
sd3414     sd       415 0x00000000 S    scsi    scsibus136
sd3415     sd       416 0x00000010 S    scsi    scsibus136
sd3416     sd       417 0x00000020 S    scsi    scsibus136
sd3417     sd       418 0x00000030 S    scsi    scsibus136
sd3418     sd       419 0x00000040 S    scsi    scsibus136
sd3419     sd       420 0x00000050 S    scsi    scsibus136
scsibus137 scsibus  137 0x07000060 SM   mscsi   fcbr8
pbay129    pbay     129 0x00000100 S    scsi    scsibus137
sd3384     sd       385 0x00000000 S    scsi    scsibus137
sd3385     sd       386 0x00000010 S    scsi    scsibus137
sd3386     sd       387 0x00000020 S    scsi    scsibus137
sd3387     sd       388 0x00000030 S    scsi    scsibus137
sd3388     sd       389 0x00000040 S    scsi    scsibus137
sd3389     sd       390 0x00000050 S    scsi    scsibus137
fcbr9      fcbr       9 0x00fe1000 SP   fcp     fabric1
scsibus72  scsibus   72 0x00000070 SM   mscsi   fcbr9
pbay72     pbay      72 0x00000100 S    scsi    scsibus72
sd3432     sd       433 0x00000000 S    scsi    scsibus72
sd3433     sd       434 0x00000010 S    scsi    scsibus72
sd3434     sd       435 0x00000020 S    scsi    scsibus72
sd3435     sd       436 0x00000030 S    scsi    scsibus72
sd3436     sd       437 0x00000040 S    scsi    scsibus72
sd3437     sd       438 0x00000050 S    scsi    scsibus72
scsibus73  scsibus   73 0x01000070 SM   mscsi   fcbr9
pbay73     pbay      73 0x00000100 S    scsi    scsibus73
sd3438     sd       439 0x00000000 S    scsi    scsibus73
sd3439     sd       440 0x00000010 S    scsi    scsibus73
sd3440     sd       441 0x00000020 S    scsi    scsibus73
sd3441     sd       442 0x00000030 S    scsi    scsibus73
sd3442     sd       443 0x00000040 S    scsi    scsibus73
sd3443     sd       444 0x00000050 S    scsi    scsibus73
scsibus74  scsibus   74 0x02000060 SM   mscsi   fcbr9
pbay74     pbay      74 0x00000100 S    scsi    scsibus74
sd3444     sd       445 0x00000000 S    scsi    scsibus74
sd3445     sd       446 0x00000010 S    scsi    scsibus74
sd3446     sd       447 0x00000020 S    scsi    scsibus74
sd3447     sd       448 0x00000030 S    scsi    scsibus74
sd3448     sd       449 0x00000040 S    scsi    scsibus74
sd3449     sd       450 0x00000050 S    scsi    scsibus74
scsibus75  scsibus   75 0x03000060 SM   mscsi   fcbr9
pbay75     pbay      75 0x00000100 S    scsi    scsibus75
sd3450     sd       451 0x00000000 S    scsi    scsibus75
sd3451     sd       452 0x00000010 S    scsi    scsibus75
sd3452     sd       453 0x00000020 S    scsi    scsibus75
sd3453     sd       454 0x00000030 S    scsi    scsibus75
sd3454     sd       455 0x00000040 S    scsi    scsibus75
sd3455     sd       456 0x00000050 S    scsi    scsibus75
scsibus76  scsibus   76 0x04000070 SM   mscsi   fcbr9
pbay76     pbay      76 0x00000100 S    scsi    scsibus76
sd3456     sd       457 0x00000000 S    scsi    scsibus76
sd3457     sd       458 0x00000010 S    scsi    scsibus76
sd3458     sd       459 0x00000020 S    scsi    scsibus76
sd3459     sd       460 0x00000030 S    scsi    scsibus76
sd3460     sd       461 0x00000040 S    scsi    scsibus76
sd3461     sd       462 0x00000050 S    scsi    scsibus76
scsibus77  scsibus   77 0x05000070 SM   mscsi   fcbr9
pbay77     pbay      77 0x00000100 S    scsi    scsibus77
sd3462     sd       463 0x00000000 S    scsi    scsibus77
sd3463     sd       464 0x00000010 S    scsi    scsibus77
sd3464     sd       465 0x00000020 S    scsi    scsibus77
sd3465     sd       466 0x00000030 S    scsi    scsibus77
sd3466     sd       467 0x00000040 S    scsi    scsibus77
sd3467     sd       468 0x00000050 S    scsi    scsibus77
scsibus78  scsibus   78 0x06000060 SM   mscsi   fcbr9
pbay78     pbay      78 0x00000100 S    scsi    scsibus78
sd3468     sd       469 0x00000000 S    scsi    scsibus78
sd3469     sd       470 0x00000010 S    scsi    scsibus78
sd3470     sd       471 0x00000020 S    scsi    scsibus78
sd3471     sd       472 0x00000030 S    scsi    scsibus78
sd3472     sd       473 0x00000040 S    scsi    scsibus78
sd3473     sd       474 0x00000050 S    scsi    scsibus78
scsibus79  scsibus   79 0x07000060 SM   mscsi   fcbr9
pbay79     pbay      79 0x00000100 S    scsi    scsibus79
sd3474     sd       475 0x00000000 S    scsi    scsibus79
sd3475     sd       476 0x00000010 S    scsi    scsibus79
sd3476     sd       477 0x00000020 S    scsi    scsibus79
sd3477     sd       478 0x00000030 S    scsi    scsibus79
sd3478     sd       479 0x00000040 S    scsi    scsibus79
sd3479     sd       480 0x00000050 S    scsi    scsibus79
fcbr10     fcbr      10 0x00fe1800 SP   fcp     fabric1
scsibus80  scsibus   80 0x00000070 SM   mscsi   fcbr10
pbay80     pbay      80 0x00000100 S    scsi    scsibus80
sd3480     sd       481 0x00000000 S    scsi    scsibus80
sd3481     sd       482 0x00000010 S    scsi    scsibus80
sd3482     sd       483 0x00000020 S    scsi    scsibus80
sd3483     sd       484 0x00000030 S    scsi    scsibus80
sd3484     sd       485 0x00000040 S    scsi    scsibus80
sd3485     sd       486 0x00000050 S    scsi    scsibus80
scsibus81  scsibus   81 0x01000070 SM   mscsi   fcbr10
pbay81     pbay      81 0x00000100 S    scsi    scsibus81
sd3486     sd       487 0x00000000 S    scsi    scsibus81
sd3487     sd       488 0x00000010 S    scsi    scsibus81
sd3488     sd       489 0x00000020 S    scsi    scsibus81
sd3489     sd       490 0x00000030 S    scsi    scsibus81
sd3490     sd       491 0x00000040 S    scsi    scsibus81
sd3491     sd       492 0x00000050 S    scsi    scsibus81
scsibus82  scsibus   82 0x02000060 SM   mscsi   fcbr10
pbay82     pbay      82 0x00000100 S    scsi    scsibus82
sd3492     sd       493 0x00000000 S    scsi    scsibus82
sd3493     sd       494 0x00000010 S    scsi    scsibus82
sd3494     sd       495 0x00000020 S    scsi    scsibus82
sd3495     sd       496 0x00000030 S    scsi    scsibus82
sd3496     sd       497 0x00000040 S    scsi    scsibus82
sd3497     sd       498 0x00000050 S    scsi    scsibus82
scsibus83  scsibus   83 0x03000060 SM   mscsi   fcbr10
pbay83     pbay      83 0x00000100 S    scsi    scsibus83
sd3498     sd       499 0x00000000 S    scsi    scsibus83
sd3499     sd       500 0x00000010 S    scsi    scsibus83
sd3500     sd       501 0x00000020 S    scsi    scsibus83
sd3501     sd       502 0x00000030 S    scsi    scsibus83
sd3502     sd       503 0x00000040 S    scsi    scsibus83
sd3503     sd       504 0x00000050 S    scsi    scsibus83
scsibus84  scsibus   84 0x04000070 SM   mscsi   fcbr10
pbay84     pbay      84 0x00000100 S    scsi    scsibus84
sd3504     sd       505 0x00000000 S    scsi    scsibus84
sd3505     sd       506 0x00000010 S    scsi    scsibus84
sd3506     sd       507 0x00000020 S    scsi    scsibus84
sd3507     sd       508 0x00000030 S    scsi    scsibus84
sd3508     sd       509 0x00000040 S    scsi    scsibus84
sd3509     sd       510 0x00000050 S    scsi    scsibus84
scsibus85  scsibus   85 0x05000070 SM   mscsi   fcbr10
pbay85     pbay      85 0x00000100 S    scsi    scsibus85
sd3510     sd       511 0x00000000 S    scsi    scsibus85
sd3511     sd       512 0x00000010 S    scsi    scsibus85
sd3512     sd       513 0x00000020 S    scsi    scsibus85
sd3513     sd       514 0x00000030 S    scsi    scsibus85
sd3514     sd       515 0x00000040 S    scsi    scsibus85
sd3515     sd       516 0x00000050 S    scsi    scsibus85
scsibus86  scsibus   86 0x06000060 SM   mscsi   fcbr10
pbay86     pbay      86 0x00000100 S    scsi    scsibus86
sd3516     sd       517 0x00000000 S    scsi    scsibus86
sd3517     sd       518 0x00000010 S    scsi    scsibus86
sd3518     sd       519 0x00000020 S    scsi    scsibus86
sd3519     sd       520 0x00000030 S    scsi    scsibus86
sd3520     sd       521 0x00000040 S    scsi    scsibus86
sd3521     sd       522 0x00000050 S    scsi    scsibus86
scsibus87  scsibus   87 0x07000060 SM   mscsi   fcbr10
pbay87     pbay      87 0x00000100 S    scsi    scsibus87
sd3522     sd       523 0x00000000 S    scsi    scsibus87
sd3523     sd       524 0x00000010 S    scsi    scsibus87
sd3524     sd       525 0x00000020 S    scsi    scsibus87
sd3525     sd       526 0x00000030 S    scsi    scsibus87
sd3526     sd       527 0x00000040 S    scsi    scsibus87
sd3527     sd       528 0x00000050 S    scsi    scsibus87
fcbr11     fcbr      11 0x00fe2000 SP   fcp     fabric1
scsibus88  scsibus   88 0x00000070 SM   mscsi   fcbr11
pbay88     pbay      88 0x00000100 S    scsi    scsibus88
sd3528     sd       529 0x00000000 S    scsi    scsibus88
sd3529     sd       530 0x00000010 S    scsi    scsibus88
sd3530     sd       531 0x00000020 S    scsi    scsibus88
sd3531     sd       532 0x00000030 S    scsi    scsibus88
sd3532     sd       533 0x00000040 S    scsi    scsibus88
sd3533     sd       534 0x00000050 S    scsi    scsibus88
scsibus89  scsibus   89 0x01000070 SM   mscsi   fcbr11
pbay89     pbay      89 0x00000100 S    scsi    scsibus89
sd3534     sd       535 0x00000000 S    scsi    scsibus89
sd3535     sd       536 0x00000010 S    scsi    scsibus89
sd3536     sd       537 0x00000020 S    scsi    scsibus89
sd3537     sd       538 0x00000030 S    scsi    scsibus89
sd3538     sd       539 0x00000040 S    scsi    scsibus89
sd3539     sd       540 0x00000050 S    scsi    scsibus89
scsibus90  scsibus   90 0x02000060 SM   mscsi   fcbr11
pbay90     pbay      90 0x00000100 S    scsi    scsibus90
sd3540     sd       541 0x00000000 S    scsi    scsibus90
sd3541     sd       542 0x00000010 S    scsi    scsibus90
sd3542     sd       543 0x00000020 S    scsi    scsibus90
sd3543     sd       544 0x00000030 S    scsi    scsibus90
sd3544     sd       545 0x00000040 S    scsi    scsibus90
sd3545     sd       546 0x00000050 S    scsi    scsibus90
```

Appendix C – Tunable Parameters

```
scsibus91 scsibus 91 0x03000060 SM mscsi fcbr11
pbay91 pbay 91 0x00000100 S scsi scsibus91
sd3546 sd 547 0x00000000 S scsi scsibus91
sd3547 sd 548 0x00000010 S scsi scsibus91
sd3548 sd 549 0x00000020 S scsi scsibus91
sd3549 sd 550 0x00000030 S scsi scsibus91
sd3550 sd 551 0x00000040 S scsi scsibus91
sd3551 sd 552 0x00000050 S scsi scsibus91
scsibus92 scsibus 92 0x04000070 SM mscsi fcbr11
pbay92 pbay 92 0x00000100 S scsi scsibus92
sd3552 sd 553 0x00000000 S scsi scsibus92
sd3553 sd 630 0x00000010 S scsi scsibus92
sd3554 sd 555 0x00000020 S scsi scsibus92
sd3555 sd 556 0x00000030 S scsi scsibus92
sd3556 sd 557 0x00000040 S scsi scsibus92
sd3557 sd 558 0x00000050 S scsi scsibus92
scsibus93 scsibus 93 0x05000070 SM mscsi fcbr11
pbay93 pbay 93 0x00000100 S scsi scsibus93
sd3558 sd 559 0x00000000 S scsi scsibus93
sd3559 sd 560 0x00000010 S scsi scsibus93
sd3560 sd 561 0x00000020 S scsi scsibus93
sd3561 sd 562 0x00000030 S scsi scsibus93
sd3562 sd 563 0x00000040 S scsi scsibus93
sd3563 sd 564 0x00000050 S scsi scsibus93
scsibus94 scsibus 94 0x06000060 SM mscsi fcbr11
pbay94 pbay 94 0x00000100 S scsi scsibus94
sd3564 sd 565 0x00000000 S scsi scsibus94
sd3565 sd 566 0x00000010 S scsi scsibus94
sd3566 sd 567 0x00000020 S scsi scsibus94
sd3567 sd 568 0x00000030 S scsi scsibus94
sd3568 sd 569 0x00000040 S scsi scsibus94
sd3569 sd 570 0x00000050 S scsi scsibus94
scsibus95 scsibus 95 0x07000060 SM mscsi fcbr11
pbay95 pbay 95 0x00000100 S scsi scsibus95
sd3570 sd 571 0x00000000 S scsi scsibus95
sd3571 sd 572 0x00000010 S scsi scsibus95
sd3572 sd 573 0x00000020 S scsi scsibus95
sd3573 sd 574 0x00000030 S scsi scsibus95
sd3574 sd 575 0x00000040 S scsi scsibus95
sd3575 sd 576 0x00000050 S scsi scsibus95
fcbr12 fcbr 12 0x00fe2800 SP fcp fabric1
scsibus96 scsibus 96 0x00000070 SM mscsi fcbr12
pbay96 pbay 96 0x00000100 S scsi scsibus96
sd3576 sd 577 0x00000000 S scsi scsibus96
sd3577 sd 578 0x00000010 S scsi scsibus96
sd3578 sd 579 0x00000020 S scsi scsibus96
sd3579 sd 580 0x00000030 S scsi scsibus96
sd3580 sd 581 0x00000040 S scsi scsibus96
sd3581 sd 582 0x00000050 S scsi scsibus96
scsibus97 scsibus 97 0x01000070 SM mscsi fcbr12
pbay97 pbay 97 0x00000100 S scsi scsibus97
sd3582 sd 583 0x00000000 S scsi scsibus97
sd3583 sd 584 0x00000010 S scsi scsibus97
sd3584 sd 585 0x00000020 S scsi scsibus97
sd3585 sd 586 0x00000030 S scsi scsibus97
sd3586 sd 587 0x00000040 S scsi scsibus97
sd3587 sd 588 0x00000050 S scsi scsibus97
scsibus98 scsibus 98 0x02000060 SM mscsi fcbr12
scsibus99 scsibus 99 0x03000060 SM mscsi fcbr12
scsibus100 scsibus 100 0x04000070 SM mscsi fcbr12
pbay98 pbay 98 0x00000100 S scsi scsibus100
sd3588 sd 589 0x00000000 S scsi scsibus100
sd3589 sd 590 0x00000010 S scsi scsibus100
sd3590 sd 591 0x00000020 S scsi scsibus100
sd3591 sd 592 0x00000030 S scsi scsibus100
sd3592 sd 593 0x00000040 S scsi scsibus100
sd3593 sd 594 0x00000050 S scsi scsibus100
scsibus101 scsibus 101 0x05000070 SM mscsi fcbr12
pbay99 pbay 99 0x00000100 S scsi scsibus101
sd3594 sd 595 0x00000000 S scsi scsibus101
sd3595 sd 596 0x00000010 S scsi scsibus101
sd3596 sd 597 0x00000020 S scsi scsibus101
sd3597 sd 598 0x00000030 S scsi scsibus101
sd3598 sd 599 0x00000040 S scsi scsibus101
sd3599 sd 600 0x00000050 S scsi scsibus101
scsibus102 scsibus 102 0x06000060 SM mscsi fcbr12
scsibus103 scsibus 103 0x07000060 SM mscsi fcbr12
qlc1 qlc 1 0x00000002 L pci quad0
scsibus113 scsibus 113 0x00000070 L mscsi qlc1
pbay105 pbay 105 0x00000100 L scsi scsibus113
sd2 sd 631 0x00000000 L scsi scsibus113
sd444 sd 632 0x00000010 L scsi scsibus113
sd1 sd 633 0x00000020 L scsi scsibus113
sd654 sd 124 0x00000030 L scsi scsibus113
sd8 sd 634 0x00000040 L scsi scsibus113
sd392 sd 635 0x00000050 L scsi scsibus113
qlc2 qlc 2 0x00000003 L pci quad0
scsibus114 scsibus 114 0x00000070 L mscsi qlc2
pbay106 pbay 106 0x00000100 L scsi scsibus114
sd2000 sd 636 0x00000030 L scsi scsibus114
sd2001 sd 637 0x00000040 L scsi scsibus114
sd2002 sd 638 0x00000050 L scsi scsibus114
qlc3 qlc 3 0x00000004 L pci quad0
scsibus115 scsibus 115 0x00000070 L mscsi qlc3
pbay107 pbay 107 0x00000100 L scsi scsibus115
sd2003 sd 639 0x00000030 L scsi scsibus115
sd2004 sd 640 0x00000040 L scsi scsibus115
sd2005 sd 641 0x00000050 L scsi scsibus115
pe1 pe 4 0x00000005 L pci quad0
pci1 pci 1 0x00000001 L pci quad0
pe0 pe 5 0x00000001 L pci pci1
pe11 pe 6 0x00000001 L pci pci1
pe12 pe 7 0x00000001 L pci pci1
pe13 pe 8 0x00000001 L pci pci1
quad1 quad 5 0x00000001 L sci sci1
ff2 ff 10 0x00000000 SP pci quad1
fabric0 fabric 10 0x00fe7000 SM fc ff2
ff3 ff 11 0x00000006 SP pci quad1
fabric1 fabric 9 0x00fe7000 SM fc ff3
qlc4 qlc 4 0x00000002 L pci quad1
scsibus116 scsibus 116 0x00000070 L mscsi qlc4
pbay108 pbay 108 0x00000100 L scsi scsibus116
sd2006 sd 642 0x00000030 L scsi scsibus116
sd2007 sd 643 0x00000040 L scsi scsibus116
sd2008 sd 644 0x00000050 L scsi scsibus116
qlc5 qlc 5 0x00000004 L pci quad1
scsibus117 scsibus 117 0x00000070 L mscsi qlc5
pbay109 pbay 109 0x00000100 L scsi scsibus117
sd2009 sd 645 0x00000030 L scsi scsibus117
sd2010 sd 646 0x00000040 L scsi scsibus117
sd2011 sd 647 0x00000050 L scsi scsibus117
pe2 pe 9 0x00000003 L pci quad1
quad2 quad 6 0x00000002 L sci sci1
ff4 ff 12 0x00000000 SP pci quad2
fabric0 fabric 10 0x00fe6800 SM fc ff4
ff5 ff 13 0x00000006 SP pci quad2
fabric1 fabric 9 0x00fe6800 SM fc ff5
qlc6 qlc 6 0x00000002 L pci quad2
scsibus118 scsibus 118 0x00000070 L mscsi qlc6
pbay110 pbay 110 0x00000100 L scsi scsibus118
sd2012 sd 648 0x00000030 L scsi scsibus118
sd2013 sd 649 0x00000040 L scsi scsibus118
sd2014 sd 650 0x00000050 L scsi scsibus118
qlc7 qlc 7 0x00000004 L pci quad2
scsibus119 scsibus 119 0x00000070 L mscsi qlc7
pbay111 pbay 111 0x00000100 L scsi scsibus119
sd2015 sd 651 0x00000030 L scsi scsibus119
sd2016 sd 652 0x00000040 L scsi scsibus119
sd2017 sd 653 0x00000050 L scsi scsibus119
pe3 pe 10 0x00000003 L pci quad2
quad3 quad 7 0x00000003 L sci sci1
ff6 ff 14 0x00000000 SP pci quad3
fabric0 fabric 10 0x00fe6000 SM fc ff6
ff7 ff 15 0x00000006 SP pci quad3
fabric1 fabric 9 0x00fe6000 SM fc ff7
qlc8 qlc 8 0x00000002 L pci quad3
scsibus120 scsibus 120 0x00000070 L mscsi qlc8
pbay112 pbay 112 0x00000100 L scsi scsibus120
sd2018 sd 654 0x00000030 L scsi scsibus120
sd2019 sd 655 0x00000040 L scsi scsibus120
sd2020 sd 656 0x00000050 L scsi scsibus120
qlc9 qlc 9 0x00000004 L pci quad3
scsibus121 scsibus 121 0x00000070 L mscsi qlc9
pbay113 pbay 113 0x00000100 L scsi scsibus121
sd2021 sd 657 0x00000030 L scsi scsibus121
sd2022 sd 658 0x00000040 L scsi scsibus121
sd2023 sd 659 0x00000050 L scsi scsibus121
pe4 pe 11 0x00000003 L pci quad3
quad4 quad 8 0x00000004 L sci sci1
ff8 ff 24 0x00000000 SP pci quad4
fabric0 fabric 10 0x00fe5800 SM fc ff8
ff9 ff 17 0x00000006 SP pci quad4
fabric1 fabric 9 0x00fe5800 SM fc ff9
qlc10 qlc 10 0x00000002 L pci quad4
scsibus122 scsibus 122 0x00000070 L mscsi qlc10
pbay114 pbay 114 0x00000100 L scsi scsibus122
sd2024 sd 660 0x00000030 L scsi scsibus122
sd2025 sd 661 0x00000040 L scsi scsibus122
sd2026 sd 662 0x00000050 L scsi scsibus122
qlc11 qlc 11 0x00000004 L pci quad4
scsibus123 scsibus 123 0x00000070 L mscsi qlc11
pbay115 pbay 115 0x00000100 L scsi scsibus123
sd2027 sd 663 0x00000030 L scsi scsibus123
sd2028 sd 664 0x00000040 L scsi scsibus123
sd2029 sd 665 0x00000050 L scsi scsibus123
pe5 pe 12 0x00000003 L pci quad4
quad5 quad 9 0x00000005 L sci sci1
ff10 ff 18 0x00000000 SP pci quad5
```

```
fabric0     fabric      10  0x00fe5000  SM     fc       ff10
ff20        ff          20  0x00000006  SP     pci      quad5
fabric1     fabric       9  0x00fe5000  SM     fc       ff20
qlc12       qlc         12  0x00000002  L      pci      quad5
scsibus124  scsibus     124  0x00000070  L      mscsi    qlc12
pbay116     pbay        116  0x00000100  L      scsi     scsibus124
sd2030      sd          666  0x00000030  L      scsi     scsibus124
sd2031      sd          667  0x00000040  L      scsi     scsibus124
sd2032      sd          668  0x00000050  L      scsi     scsibus124
qlc13       qlc         13  0x00000004  L      pci      quad5
scsibus125  scsibus     125  0x00000070  L      mscsi    qlc13
pbay117     pbay        117  0x00000100  L      scsi     scsibus125
sd2033      sd          669  0x00000030  L      scsi     scsibus125
sd2034      sd          670  0x00000040  L      scsi     scsibus125
sd2035      sd          671  0x00000050  L      scsi     scsibus125
pe6         pe          13  0x00000003  L      pci      quad5
quad6       quad        10  0x00000006  L      sci      sci1
ff12        ff          16  0x00000000  SP     pci      quad6
fabric0     fabric      10  0x00fe4800  SM     fc       ff12
ff13        ff          21  0x00000006  SP     pci      quad6
fabric1     fabric       9  0x00fe4800  SM     fc       ff13
qlc14       qlc         14  0x00000002  L      pci      quad6
scsibus126  scsibus     126  0x00000070  L      mscsi    qlc14
pbay118     pbay        118  0x00000100  L      scsi     scsibus126
sd2036      sd          672  0x00000030  L      scsi     scsibus126
sd2037      sd          673  0x00000040  L      scsi     scsibus126
sd2038      sd          674  0x00000050  L      scsi     scsibus126
qlc15       qlc         15  0x00000004  L      pci      quad6
scsibus127  scsibus     127  0x00000070  L      mscsi    qlc15
pbay119     pbay        119  0x00000100  L      scsi     scsibus127
sd2039      sd          675  0x00000030  L      scsi     scsibus127
sd2040      sd          676  0x00000040  L      scsi     scsibus127
sd2041      sd          677  0x00000050  L      scsi     scsibus127
pe7         pe          14  0x00000003  L      pci      quad6
quad7       quad        11  0x00000007  L      sci      sci1
ff14        ff          26  0x00000000  SP     pci      quad7
fabric0     fabric      10  0x00fe4000  SM     fc       ff14
ff15        ff          23  0x00000006  SP     pci      quad7
fabric1     fabric       9  0x00fe3800  SM     fc       ff15
qlc16       qlc         16  0x00000002  L      pci      quad7
scsibus128  scsibus     128  0x00000070  L      mscsi    qlc16
pbay120     pbay        120  0x00000100  L      scsi     scsibus128
sd2042      sd          678  0x00000030  L      scsi     scsibus128
sd2043      sd          679  0x00000040  L      scsi     scsibus128
sd2044      sd          680  0x00000050  L      scsi     scsibus128
qlc17       qlc         17  0x00000004  L      pci      quad7
scsibus129  scsibus     129  0x00000070  L      mscsi    qlc17
pbay121     pbay        121  0x00000100  L      scsi     scsibus129
sd2045      sd          681  0x00000030  L      scsi     scsibus129
sd2046      sd          682  0x00000040  L      scsi     scsibus129
sd2047      sd          683  0x00000050  L      scsi     scsibus129
pe8         pe          15  0x00000003  L      pci      quad7
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       -
slm         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

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
   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)
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)

Appendix C – Tunable Parameters

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)
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

Appendix C – Tunable Parameters

```
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
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

Appendix C – Tunable Parameters

```
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: 0x50

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

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: 0x1964

Appendix C – Tunable Parameters

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

Rte4032w.cfg

```
Sequent
Oracle 804
88 68 14400 2 1 1 1
trans_params_tpcc
/scripts/tpcc/tpcc.dll 87 4608
slave1 client1_1 1 84
slave2 client1_2 85 168
slave3 client1_3 169 252
slave4 client1_4 253 336
slave5 client1_5 337 420
slave6 client1_6 421 504
slave7 client2_1 577 660
slave8 client2_2 661 744
slave9 client2_3 745 828
slave10 client2_4 829 912
slave11 client2_5 913 996
slave12 client2_6 997 1080
slave13 client3_1 1153 1236
slave14 client3_2 1237 1320
slave15 client3_3 1321 1404
slave16 client3_4 1405 1488
slave17 client3_5 1489 1572
slave18 client3_6 1573 1656
slave19 client4_1 1729 1812
slave20 client4_2 1813 1896
slave21 client4_3 1897 1980
slave49 client4_4 1981 2064
slave23 client4_5 2065 2148
slave24 client4_6 2149 2232
slave25 client1_7 2305 2388
slave26 client1_8 2389 2472
slave27 client1_9 2473 2556
slave28 client1_10 2557 2640
slave29 client1_11 2641 2724
slave30 client1_12 2725 2808
slave31 client2_7 2881 2964
slave32 client2_8 2965 3048
slave33 client2_9 3049 3132
slave34 client2_10 3133 3216
slave35 client2_11 3217 3300
slave36 client2_12 3301 3384
slave37 client3_7 3457 3540
slave38 client3_8 3541 3624
slave39 client3_9 3625 3708
slave40 client3_10 3709 3792
slave41 client3_11 3793 3876
slave42 client3_12 3877 3960
slave43 client4_7 4033 4116
slave44 client4_8 4117 4200
slave45 client4_9 4201 4284
slave46 client4_10 4285 4368
slave47 client4_11 4369 4452
slave48 client4_12 4453 4536
```

trans_params_tpcc

```
NewOrder     44.80  12.01  18.00   0.10   5.00   0.10
Payment      43.05  12.03   3.00   0.10   5.00   0.10
OrderStatus   4.05  10.10   2.00   0.10   5.00   0.10
Delivery      4.05   5.02   2.00   0.10   5.00   0.10
StockLevel    4.05   5.02   2.00   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 94089

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 4829 P.02