UNISYS

ORACLE

TPC Benchmark™ C Full Disclosure Report

Unisys e-@ction ES2085R Enterprise Server

using

Oracle 8i, Enterprise Edition

on

UnixWare 7 Data Center Edition

First Edition
December 13th 1999

Unisys Part Number 4500 5121-000

First Edition – December 1999

Unisys Corporation and Oracle Corporation believe that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. Unisys Corporation and Oracle Corporation assume no responsibility for any errors that may appear in this document.

The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, Unisys Corporation and Oracle Corporation provide no warranty on the pricing information in this document.

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 Benchmark C should not be used as a substitute for a specific customer application benchmark when critical capacity planning and/or product evaluation decisions are contemplated.

All performance data contained in this report was obtained in a rigorously controlled environment, and therefore results obtained in other operating environments may vary significantly. Unisys Corporation and Oracle Corporation do not warrant or represent that a user can or will achieve similar performance expressed in transactions per minute (tpmC) or normalized price/performance ($/tpmC). No warranty of system performance or price/performance is expressed or implied in this report.

Printed in USA, December 1999.

Unisys Corporation Part Number: 4500 5121-000

Page Status

Page	Issue
i through xii	-000
0-1 through 0-3	-000
0-4	Blank
1-1 through 1-1	-000
1-2	Blank
2-1 through 2-2	-000
3-1 through 3-4	-000
4-1 through 4-5	-000
4-6	Blank
5-1 through 5-8	-000
6-1 through 6-2	-000
7-1 through 7-2	-000
8-1 through 8-1	-000
8-2	Blank
9-1 through 9-3	-000
9-4	Blank
A-1 through A-79	-000
A-80	Blank
B-1 through B-92	-000
C-1 through C-50	-000
D-1 through D-4	-000
E-1 through E-2	-000
F-1 through F-6	-000

Unisys uses an 11-digit document numbering system. The suffix of the document number (1234 5678-xyz) indicates the document level. The first digit of the suffix (x) designates a revision level; the second digit (y) designates an update level. For example, the first release of a document has a suffix of -000. A suffix of -130 designates the third update to revision 1. The third digit (z) is used to indicate an errata for a particular level and is not reflected in the page status summary.

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C (TPC-C) conducted on the Unisys e-@ction ES2085R Enterprise Server. The operating system on the server was Santa Cruz Operation UnixWare 7 Data Center Edition, version 7.1.1. The DBMS used was Oracle 8i, Enterprise Edition. The operating system on the Tuxedo client was SCO UnixWare 7 Business Edition, version 7.1.1 and on the web clients was Microsoft Windows NT Server 4.0. The clients ran Tuxcedo 6.5 CFS for UnixWare, Tuxedo 6.4 CFS for NT and Microsoft's Internet Information Server 3.0.

TPC Benchmark Metrics

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

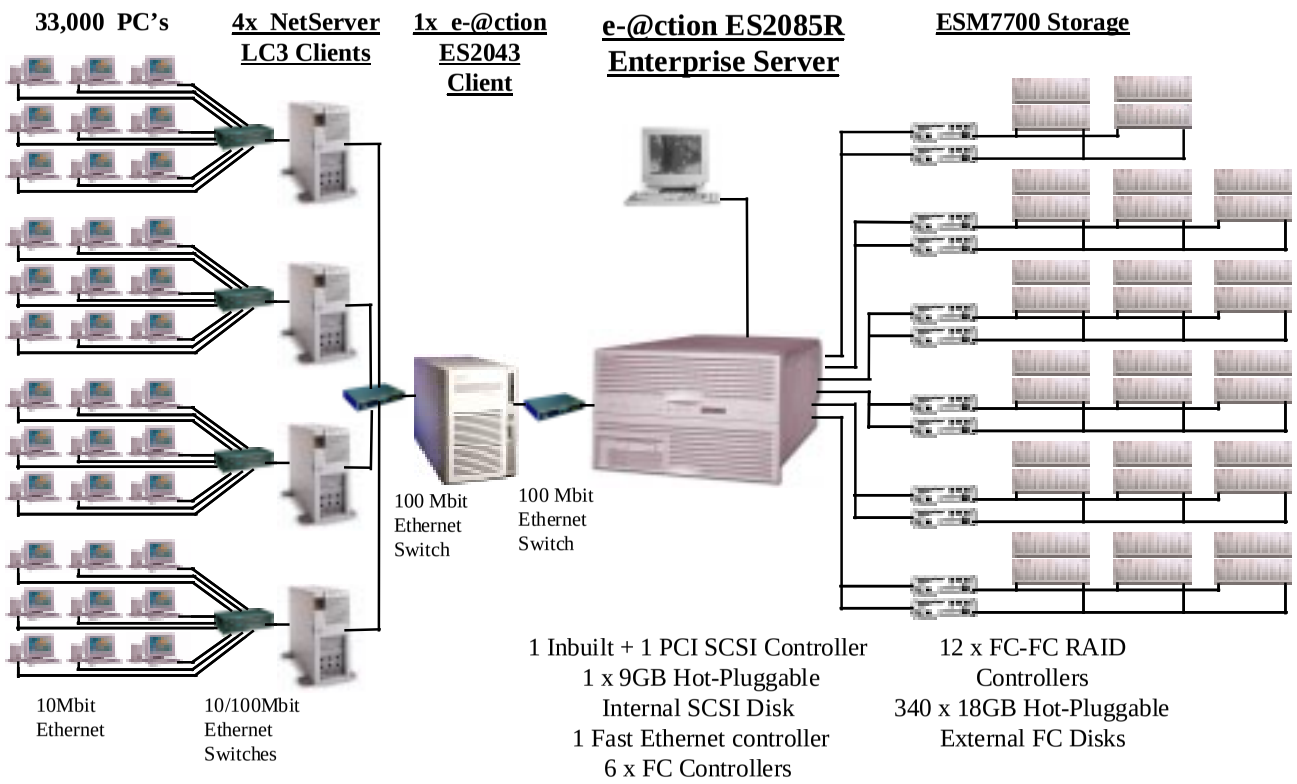
Executive Summary

The following pages contain the executive summary results of the benchmark.

Auditor

The benchmark configuration, environment, and methodology used to produce and validate the test results, along with the pricing model used to calculate the cost per tpmC, were audited by Tom Sawyer of Performance Metrics, Inc. to verify compliance with the relevant TPC specification.

UNISYS ORACLE	e-@ction ES2085R Enterprise Server (8P 550MHz/2MB)			TPC-C Rev. 3.5 Report Date: 13-Dec-1999
Total System Cost	TPC-C Throughput	Price/Performance		Availability Date
$1,527,980	**41,085.43 tpmC**	**$37.19 per tpmC**		**1-Jun-2000 ***
Processors	Database Manager	Operating System	Other Software	Number of Users
8 Pentium® III Xeon 550 MHz 2MB L2 cache	Oracle 8i, Version 8.1.6	UnixWare 7 Data Center Edition, Version 7.1.1	UnixWare 7 Business Edtn. Windows NT Server 4.0 Tuxedo 6.4 & 6.5 CFS	**33,000**



System Components	Enterprise Server		ES2043 Client		LC3 Clients (each of 4)	
	Quantity	Type	Quantity	Type	Quantity	Type
Processors	8	550 MHz Pentium® III Xeon with 2MB Level 2 Cache	4	500MHz Pentium® III Xeon w/ 1MB Level 2 Cache	2	500MHz Pentium® III w/ 512KB Level 2 Cache
Memory	1	16,384 MB	1	1024 MB	1	1024 MB
Disk Controllers	1 + 1 6	Inbuilt + PCI SCSI Fiber Channel Controllers	1	PCI SCSI	1	Inbuilt SCSI
Disk Drives	1 340	8.54 GB 16.54 GB	1	8.54 GB	1	8.54 GB
Total Storage		5565.73 GB		8.54 GB		8.54 GB
CD-ROM / Tape	1	SCSI CD-ROM Drive	1	CD-ROM Drive	1	CD-ROM Drive

* All hardware components are available now.

UNISYS ORACLE

e-@ction ES2085R Enterprise Server
(8P 550MHz/2MB)

TPC-C Rev 3.5
13-Dec-1999

Description	Style	Third Party Brand	Pricing	Unit Price	Qty.	Extended Price	5 Years Maint.
Server Hardware							
SYS: e-@ction ES2085R, w/ CDRom, 0 Proc, 0MB	ESR208151-GZU		1	$13,997	1	$13,997	$5,040
PROC: 550MHz P-III Xeon /2MB Cache & VRMs	XEO3550-2MB		1	$5,893	8	$47,144	$13,248
BRD: Processor Mezzanine Board, 0 Proc.	ESR81-MEZ		1	$1,179	1	$1,179	$312
MEM: 512 MB Memory, SDRAM, Buf 6ns	DIM6168-512		1	$1,989	32	$63,648	$13,056
BRD: Memory Carrier Board, 0 Mem.	ESR81-MCB		1	$737	1	$737	$192
MEM: Cache Coherency Filter, 4x SRAM	ESR81-CC4		1	$921	2	$1,842	$480
CTRL: Fiber Channel HBA, 32/64-bit PCI	PCI1100-FC		1	$1,175	6	$7,050	$2,304
DISK: 9GB, 10K SCSI LVD, SCA	HDL91102-CX1		1	$516	1	$516	$240
ACC: HBA, SCSI w/ VHD Connector	PCI300-SUW		1	$249	1	$249	$96
ETHERNET: 10/100Mbit/sec, PCI 32-bit	ETH1010052-PCI		1	$100	1	$100	
MONITOR: 15-inch Color	EVG2100-P		1	$221	1	$221	
					Subtotal	**$136,683**	**$34,968**
Storage Hardware							
DISK: 18GB Drive, 10Krpm, Fiber Channel	ESM18201-F96		1	$1,250	168	$210,000	Spared
DISK: 18GB Drive, 10Krpm, FC 10% spares	ESM18201-F96		1	$1,250	17		$21,250
DPE: 2 RAID Cntrls, 2x256MB, 10 18GB Disk, BBU	ESM7700-A10		1	$36,611	6	$219,666	$81,216
CAB: DAE Expn Enclsr, w/ 4 18GB Disk	ESM700-A04		1	$8,406	28	$235,368	$91,728
MEM: 256MB DPE Cache Expansion SIMM	ESM7700-44M		1	$1,479	6	$8,874	$3,024
CBL: FC, HSSDC->DB9 Conn's, 10m	CBL136-10		1	$165	6	$990	
CAB: Rackmount Kit for DPE	ESM7700-PRM		1	$84	6	$504	
CAB: Rackmount Kit for DAE	ESM700-DRM		1	$84	28	$2,352	
PWR: Distribution Kit, 220V	SFR220-PWR		1	$42	24	$1,008	
CAB: 36U x 19" x 34" Cabinet, Open	RM361934-OFT		1	$884	6	$5,304	
DOOR: 36U x 19", Rear	RM3619-RDR		1	$277	6	$1,662	
PNL: 36U x 34" Side Skins, L&R	RM3634-SDS		1	$221	6	$1,326	
					Subtotal	**$687,054**	**$197,218**
Server Software							
O/S: UnixWare 7 Data Center Edition, Ver 7.1.1, 150 users	UXW71-DCL		1	$7,366	1	$7,366	$3,792
O/S: UnixWare 7 Media Kit, Version 7.1.1	UNX971-PSK		1	$277	1	$277	$144
DB: Oracle 8i, Version 8.1.6, unlimited user license				$166,810	1	$166,810	$166,810
					Subtotal	**$174,453**	**$170,746**
Client Hardware							
SYS: NetServer LC3, w/1 500MHz Proc., 0MB Mem	D7127-AV		1	$2,018	4	$8,072	$6,444
PROC:1x500MHz Pentium III/512KB Cache UPG	D7130-AV		1	$1,300	4	$5,200	
MEM: 256 MB SDRAM Memory Upgrade	D6099-AV		1	$655	16	$10,480	
DISK: 9GB SCSI 3.5 Internal	D4911-AV		1	$338	4	$1,352	$1,632
ETHERNET: 10/100TX Mbit/sec, PCI 32-bit	D5013-AV		1	$68	8	$544	
MONITOR:15-inch Color	EVG2100-P		1	$221	4	$884	
SYS: e-@ction ES2043, w/CD-ROM, 0 Proc, 0MB	ES204131-GCU		1	$3,536	1	$3,536	$3,192
PROC: 500MZ/1MB PIII Xeon/1MB Cache	XEO3500-1MB		1	$3,536	4	$14,144	$16,512
ACC: Voltage Reg Mod	XEO24001-VRM		1	$44	4	$176	
ETHERNET: 10/100Mbit/sec, PCI 32-bit	ETH1010052-PCI		1	$100	2	$200	
MEM: 128MB 50ns DIMMs	DIM5072-128		1	$398	8	$3,184	$12,288
DISK: 9GB, 10K SCSI LVD, SCA	HDL91102-CX1		1	$516	1	$516	$240
ACC: HBA, SCSI w/ VHD Connector	PCI300-SUW		1	$249	1	$249	$96
MONITOR: 15-inch Color	EVG2100-P		1	$221	1	$221	
					Subtotal	**$48,758**	**$40,404**
Client Software							
O/S: UnixWare 7 Business Edition, Version 7.1.1, 5 users	UXW71-BZL		1	$1,030	1	$1,030	$528
O/S: Processor Upgrade, UnixWare 7	UNX71-PU		1	$553	3	$1,659	
O/S: UnixWare 7 Media Kit, Version 7.1.1	UNX971-PSK		1	$277	1	$277	$144
TM: TUXEDO Core Functional Services 6.5 for UnixWare		BEA	2	$12,000	1	$12,000	$9,600
O/S: Microsoft Windows NT Server 4.0, 5 CALs		Microsoft	3	$809	4	$3,236	$10,475
ACC: Microsoft Visual C++ Professional 6.0		Microsoft	3	$549	1	$549	Inc. above
TM: TUXEDO Core Functional Services 6.4 for NT		BEA	2	$3,000	4	$12,000	$9,600
					Subtotal	**$15,785**	**$20,075**
User Connectivity							
SWITCH: Ethernet, 8-Port 100TX TrueFast + 10% spares	NX-SW8	Netlux	4	$179	12	$2,148	Spared
HUB: Ethernet, 8-Port 10Base-T (8+1 ports) + 10% spares	HB-1009DX/P	NetCruiser	5	$24	4568	$109,632	Spared
					Subtotal	**$111,780**	**$0**
					Total	**$1,174,513**	**$463,411**
Unisys Service Pre-Pay Discount							**($19,509)**
Western Micro Discount			1			**($90,435)**	

Notes:
1. HW Maintenance - Unisys 36 month warranty is upgraded to service level: Standard Performance-Gold. Last 24 months are also at service level: Standard Performance-Gold.
2. 10% or minimum 2 spares are added in place of onsite service (products have a five year return-to-vendor warranty)
3. Pricing: 1 = Western Micro, 2 = BEA, 3 = Microsoft, 4 = Netlux, 5 = NetCruiser

Five Year Cost of Ownership	**$1,527,980**
TPC-C Throughput	**41,085.43**
$/tpmC	**$37.19**

The benchmark results and test methodology were audited by Tom Sawyer of Performance Metrics, Inc.

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 assumption 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 benchmarks 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.

NUMERICAL QUANTITIES SUMMARY
for
Unisys e-@ction ES2085R Enterprise Server

MQTh, Computed Maximum Qualified Throughput: **41,085.43**
% throughput difference, reported & reproducibility runs: 0.18%

Transaction Mix

New Order	44.84%
Payment	43.06%
Delivery	4.02%
Stock-Level	4.03%
Order-Status	4.03%

Response Times

Transaction	Average	Maximum	90th %ile
New-Order	0.33	6.97	0.46
Payment	0.25	5.32	0.37
Delivery	0.12	1.82	0.12
Stock-Level	0.31	5.48	0.59
Order Status	0.31	1.84	0.43
Menu	0.12	4.51	0.12
Delivery (Deferred)	0.30	1.97	0.44

Response time delay added for emulated components (seconds)

RT Response time	0.1
Menu Response time	0.1

Keying/Think Time Times (seconds)

Transaction	Minimum	Average	Maximum
New-Order	18.18.00/0	18.18/12.14	19.6/121.51
Payment	3.03.00/0	3.03/12.16	4.45/121.51
Delivery	2.02.00/0	2.02/5.1	3.26/51.11
Stock-Level	2.02.00/0	2.02/5.11	2.74/51.11
Order-Status	2.02.00/0	2.02/10.15	3.01/101.73

Test Duration

Ramp up time	45 minutes
Measurement interval (M)	60 minutes
Transactions (all types) completed during measurement interval	5,497,023
Ramp-down time	11 minutes

Checkpointing:

Number of checkpoints	4
Checkpoint interval	15 minutes

Table of Contents

Figures

Tables

Document Structure

The TPC Benchmark C Standard Specification requires test sponsors to publish, submit to the TPC, and make available to the public, a full disclosure report for any result to be considered compliant with the specification. The required contents of the full disclosure report are specified in Clause 8.

This report is submitted to satisfy the specification's requirement for full disclosure. It documents the compliance of the benchmark implementation and execution reported for the Unisys e-@ction ES2085R Enterprise Server using Oracle 8i, Enterprise Edition, on Santa Cruz Operation UnixWare 7 Data Center Edition.

TPC Benchmark C Overview

The TPC Benchmark™ C Standard Specification Revision 3.5 was developed by the Transaction Processing Performance Council (TPC). It is the intent of the TPC to develop a suite of benchmarks to measure the performance of computer systems executing a wide range of applications. Unisys and Oracle Corporations are active participants in the TPC to define and develop such a suite of benchmarks.

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.

Despite the fact that this benchmark offers a rich environment that emulates many OLTP environments, 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.

0. General Items

0.1. Order and Titles

The order and titles of sections in the Test Sponsor's Full Disclosure report must correspond with the order and titles of sections from the TPC-C standard specification (i.e., this document). The intent is to make it as easy as possible for readers to compare and contrast material in different Full Disclosure reports.

The order and titles of the sections in this report correspond with those from the TPC-C standard specification.

0.2. Executive Summary Statement

The TPC Executive Summary Statement must be included near the beginning of the Full Disclosure report.

The TPC Executive Summary Statement is included near the beginning of this report.

0.3. Numerical Quantities Summary

The numerical quantities listed below must be summarized near the beginning of the Full Disclosure Report :

- *measurement interval in minutes,*
- *number of checkpoints in the measurement interval,*
- *checkpoint interval in minutes,*
- *number of transactions (all types) completed within the measurement interval,*
- *computed Maximum Qualified Throughput in tpmC,*
- *percentage difference between reported throughput and throughput obtained in reproducibility run,*
- *ninetieth percentile, average and maximum response times for the New-Order, Payment, Order-Status, Stock-Level, Delivery (deferred and interactive) and Menu transactions,*
- *time in seconds added to response time to compensate for delays associated with emulated components,*
- *percentage of transaction mix for each transaction type.*

These numerical quantities are summarized near the beginning of this report.

0.4. Application Code Disclosure

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

Appendix A contains the client application code used in this TPC-C benchmark.

0.5. Benchmark Sponsor

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

This TPC benchmark C was sponsored by Unisys Corporation and Oracle Corporation. The benchmark test was developed by Oracle and Unisys. The benchmark was conducted at Unisys, Mission Viejo, California.

0.6. Parameter Settings

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

- *Data Base tuning options*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and application configuration parameters*

Appendix C contains the configuration and system parameters used in running these TPC-C tests. It also contains all the client and server OS and Oracle tunable parameters.

0.7. Configuration Diagrams

Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- *Number and type of processors.*
- *Size of allocated memory, and any specific mapping/partitioning of memory unique to the test.*
- *Number and type of disk units (and controllers, if applicable).*
- *Number of channels or bus connections to disk units, including their protocol type.*
- *Number of LAN (e.g., Ethernet) connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure (see Clause 8.1.8).*
- *Type and the run-time execution location of software components (e.g., DBMS, client processes, transaction monitors, software drivers, etc.).*

The Remote Terminal Emulator (RTE) software used for these TPC-C tests is proprietary to Unisys. The benchmarked configuration of the RTE and e-@ction ES2085R Enterprise Server is illustrated in Figure 0.1. Tables 4.3, 4.4 and 4.5 contain a detailed explanation of the disk configuration.

The priced configuration for the e-@ction ES2085R Enterprise Server is shown in Figure 0.2.

Unisys e-@ction ES2085R Enterprise Server - Benchmarked Configuration

Server
e-@ction ES2085R
8x Pentium® III Xeon 550MHz CPUs,
2MB L2 cache per CPU,
16 GB memory,
1 Internal SCSI controller,
1 PCI SCSI wide controller,
2 x 4.24 GB internal SCSI disk,
1 PCI Fast Ethernet adapter,
6 Fiber Channel controllers

Storage
ESM7700 Fiber Channel Disks
300 x 16.53 GB data,
30 x 16.53 GB log,
100x 16.53 GB backup,
12 x FC-FC RAID Controllers

Clients

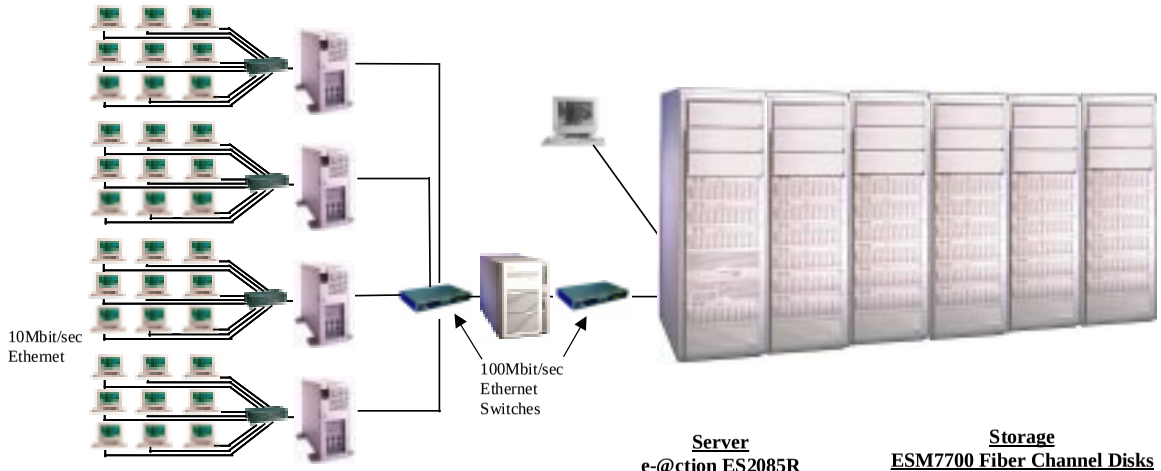
9 RTEs
emulating 33,000 users
24 LANs w/ 920 users each
12 LANs w/ 910 users each

4x NetServer LC3
2x 500MHz Pentium® III CPU,
512KB L2 cache,
1 GB memory,
1 x 4 GB disk,
2 PCI Fast Ethernet adapters

1x e-@ction ES2043
4x 500MHz Pentium® III Xeon CPUs,
1MB L2 cache,
1 GB memory,
1 x 4GB disk,
2 PCI Fast Ethernet adapters

Figure 0.1: Benchmarked Configuration

Unisys e-@ction ES2085R Enterprise Server - Priced Configuration



33,000 PC's
4 Clients w/ 8250 users

Server
e-@ction ES2085R
8x Pentium® III Xeon 550MHz CPUs,
2MB L2 cache per CPU,
16 GB memory,
1 Internal SCSI controller,
1 PCI SCSI wide controller,
1 x 8.54 GB internal SCSI disk,
1 PCI Fast Ethernet adapter,
6 Fiber Channel controllers

Storage
ESM7700 Fiber Channel Disks
300 x 16.53 GB data,
40 x 16.53 GB log,
12 x FC-FC RAID Controllers

Clients

4x NetServer LC3
2x 500MHz Pentium® III CPU,
512KB L2 cache,
1 GB memory,
1 x 8.54 GB disk,
2 PCI Fast Ethernet adapters

1x e-@ction ES2043
4x 500MHz Pentium® III Xeon CPUs,
1MB L2 cache,
1 GB memory,
1 x 8.54 GB disk,
2 PCI Fast Ethernet adapters

Figure 0.2: Priced Configuration

1. Clause 1: Logical Database Design

1.1. Table Definitions

Listings must be provided for all table definition statements and all other statements used to setup the data base.

Appendix B contains the SQL definitions of all the required database files, tables, indexes and stored procedures, plus a listing of the program used to load the database and establish the required initial populations of each table.

1.2. Physical Organization of the Database

The physical organization of tables and indices, within the data base, must be disclosed.

The disk space was allocated to Oracle according to the data in Table 4.3. The SQL definitions are contained in Appendix B.

1.3. Insert and/or 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 data base 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.

There were no restrictions on insert and/or delete operations to any of the tables.

1.4. Partitioning

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

Partitioning was not used for any table in this implementation.

1.5. Replication, Duplication or Additions

Replication of tables, if used, must be disclosed.

Additional and/or duplicate attributes in any table must be disclosed along with a statement on the impact on performance.

No replications, duplications or additional attributes were used in this implementation.

2. Clause 2: Transaction & Terminal Profiles

2.1. Random Number Generation

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

The drivers used the Unisys RTE program, which was independently audited. The initial population of the database was performed by the loader program from the Oracle TPC-C toolkit, which was also independently audited. Furthermore, the auditor sampled various initial and runtime distributions produced by this implementation to verify correctness.

2.2. Input/Output Screen Layout

The actual layout of the terminal input/output screens must be disclosed.

The screen layouts are based on those in Clauses 2.4.3, 2.5.3, 2.6.3, 2.7.3. and 2.8.3 of the TPC Benchmark C Standard Specification. There are some minor differences in appearance due to the use of a web client implementation.

2.3. Priced Terminal Feature Verification

The method used to verify that the emulated terminals provide all the features described in Clause 2.2.2.4 must be explained. Although not specifically priced, the type and model of the terminals used for the demonstration in 8.1.3.3 must be disclosed and commercially available (including supporting software and maintenance).

This was verified by the auditor.

2.4. Presentation Managers or Intelligent Terminal

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

Application code running on the client implemented the TPC-C user interface. A listing of this code is included in Appendix A. No presentation manager was used on the client, as screen manipulation and data input/output was handled for each user by the Microsoft Internet Explorer web browser running on each user PC.

2.5. Transaction Statistics

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

The number of items per order entered by New-Order transactions must be disclosed.

The percentage of home and remote Payment transactions must be disclosed.

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

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

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

Table 5.4 in Section 5 contains all these statistics.

2.6. Queuing Mechanism of Delivery

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

Deferred deliveries are queued by making an entry in an array within the appliation process (tpcc.dll) running on the web client systems. Background threads within the application process will asynchronously dequeue a request, call a Tuxedo server to process it, and log the result to a file when the response is returned. The source code for this delivery process is included in Appendix A.

3. Clause 3: Transaction & System Properties

3.1. Transaction System Properties (ACID)

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.

The TPC Benchmark C Standard Specification defines a set of transaction processing system properties that a system under test (SUT) must support during the execution of the benchmark. Those properties are Atomicity, Consistency, Isolation, and Durability (ACID).

This section defines each of these properties, describes the steps taken to ensure that they were present during the test and describes a series of tests done to demonstrate compliance with the specification. All ACID property tests were executed successfully.

3.2. Atomicity

The system under test must guarantee that data base 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.

3.2.1. Completed Transaction

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number) and verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have been changed appropriately.

The balances from a randomly selected warehouse, district, and customer row were retrieved by customer number from a script. A Payment transaction was submitted with the same warehouse, district and customer identifiers for a known amount. After completion of the Payment transaction, the balances of the selected warehouse, district, and customer were again retrieved to verify that the changes had been made correctly.

3.2.2. Aborted Transactions

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number) and substitute a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have NOT been changed.

The balances from a randomly selected warehouse, district, and customer row were retrieved by customer number from a script. A Payment transaction was submitted with the same warehouse, district and customer identifiers that issued a ROLLBACK command rather than a COMMIT. After the transaction completed, the balances of the selected warehouse, district, and customer were again retrieved to verify that no changes had been made to the database.

3.3. 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.

The benchmark specification requires explicit demonstration of the following four consistency conditions:

1. The sum of the district balances in a warehouse is equal to the warehouse balance;
2. For each district, the next order id minus one is equal to maximum order id in the ORDER table and equal to the maximum new order id in the NEW ORDER table;
3. For each district, the maximum order id minus minimum order id in the ORDER table plus one equals the number of rows in the NEW-ORDER table for that district;
4. For each district, the sum of the order line counts in the ORDER table equals the number of rows in the ORDER-LINE table for that district;

In order to demonstrate this consistency, the following steps were taken:

1. Prior to the start of a benchmark run, the consistency of the database was verified by successfully testing conditions 1-4 described above with a script.
2. After completion of the measurement run, consistency of the database was verified successfully using the same consistency script as in step 1.
3. After completion of the reproducibility run, consistency of the database was again verified successfully using the same consistency script as in step 1.

3.4. 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.

The benchmark specification defines seven required tests to be performed to demonstrate that required levels of transaction isolation are met. These tests, described in Clauses 3.4.2.1 - 3.4.2.7, were all performed from a script and verified by the auditor. In Isolation Test 7, Case D was observed. In addition, the phantom tests and stock level tests were executed and verified to be successful.

3.5. Durability

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

Three durability tests were executed to satisfy the requirements of the specification. The test for loss of memory and instantaneous interruption was combined and performed with a fully scaled database with all emulated users. The loss of log and loss of data test was performed with a fully scaled database with all emulated users.

3.5.1. Loss of Log Disk and Loss of Data Disk

The following steps were taken to demonstrate durability in the case of loss of a log disk and of a data disk.

1. The database was backed up.
2. The D_NEXT_O_ID fields for all rows in the district table were summed up to determine the initial count of orders present in the database (count1).

3. The RTE was started with 33,000 users. On the driver systems, committed and rolled back New-Order transactions were recorded in a "success" file.
4. After five minutes of running at steady state, a log disk was removed from the disk cabinet, with no effect on UnixWare or Oracle.
5. After 5 additional minutes of operation, a data disk was removed from the disk cabinet.
6. UnixWare and Oracle encountered IO errors due to the missing disk and recorded these errors in the respective log files. Oracle shutdown automatically. The RTEs also recorded errors.
7. The RTEs and clients were stopped.
8. Shutdown Abort command was issued to make sure that Oracle is shut down.
9. A data disk was inserted in the disk cabinet to replace the one removed. (The missing log drive was not replaced.)
10. Started Oracle and received an error: Datafile "filename" Not Available. Determined the logical drive that the "filename" was on.
11. Restored all the files on the logical drive set from backup after shutting down Oracle.
12. Oracle was then started with Mount (database not opened).
13. Oracle "Recover" command issued. Recovery completed successfully.
14. Consistency condition 3 of Clause 3.3.2.3 was executed to verify database consistency.
15. Step 2 was repeated to determine the total number of orders (count2). Count2 minus count1 was not less than the number of committed New-Order records in the "success" file.
16. The contents of the "success" files on the drivers were sampled to verify that the records in the "success" file for committed New-Order transactions had corresponding records in the Order table.

3.5.2. Instantaneous Interruption and Loss of Memory

Instantaneous interruption and loss of memory tests were combined because the loss of power erased the contents of memory. This failure was induced by removing the primary power to the System Under Test while the benchmark was executing.

1. The D_NEXT_O_ID fields for all rows in the district table were summed up to determine the initial count of orders present in the database (count1).
2. On the driver systems, committed and rolled back New-Order transaction were recorded in a "success" file.
3. The benchmark was executed at full load with all emulated users for a minimum of 10 minutes.
4. The system's primary power was then turned off.
5. After transaction failures were noted by the RTEs, the RTEs and clients were shutdown.
6. Power was restored to the SUT, the system was rebooted, Oracle was restarted, and automatic database recovery was performed. The database recovery used the transaction log to reapply all committed transactions and rollback any (in progress) uncommitted transactions.
7. After recovery finished, Consistency Condition 3 of Clause 3.3.2.3 (no gaps in NO_O_ID) was executed to verify that the database was consistent..

8. Next, samples of the contents of the “success” file from the drivers were compared against corresponding rows of the ORDER table to verify that records in the “success” file for committed New-Order transactions had corresponding records in the ORDER table.
9. Finally, step 1 was repeated to determine the total number of orders (count2). Count2 minus count1 was not less than the number of committed New-Order records in the “success” file.

4. Clause 4: Scaling & Database Population

4.1. Initial Cardinality of Tables

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

The TPC-C database for this test was configured with 3500 warehouses. The cardinality of each table in the database is listed in Table 4.1

Table 4.1: Initial Cardinality of Database Table

Table	Occurrences
Warehouse	3,500
District	35,000
Item	100,000
Customer	105,000,000
Stock	350,000,000
New-Order	31,500,000
History	105,000,000
Orders	105,000,000
Order-Line	1,050,000,000

200 upper warehouses were inactive and not used while executing the measurement runs.

4.2. Constant Values

The following values were used as the constant C input values to the NURand function during Build and Run time for this implementation.

Table 4.2: Constant C for NURand

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

4.3. Database Layout

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

Table 4.3 lists the distribution of the database over 300 18GB disks and the transaction log over 4 mirrored volumes of 10 18GB disks each for the benchmark configuration. In addition, there was one 4GB internal disk containing UnixWare 7 Data Center Edition and one containing Oracle 8i code and the TPCC database plus the paging file. Since these two disks had minimal IO activity during the benchmark, the priced configuration replaced them by one 9GB drive. Backup used 100 18GB disks that were not active during the benchmark and were excluded from the priced configuration. Except for those items, the tested and priced disk configurations were identical.

4.4. DBMS: Data Model and DBMS Interface/Access Language

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 8i, Enterprise Edition is a relational DBMS.

The client software used Oracle Call Interface (OCI) to interface to Oracle using SQL*Net (Net8) over TCP/IP.

4.5. DBMS Partitions/Replications

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

No table partitioning or replication was done.

4.6. DBMS Space Requirements

Details of the 180 day space computation along with proof that the database is configured to sustain 8 hours of growth for dynamic tables (Order, Order-line, and History) must be disclosed (see Clause 4.2.3).

Appendix E lists the space requirements for the 180-day space as well as the log space for eight hours.

Table 4.3: External Disk Configuration

External Disk Configuration						
Physical ID	Storage Processor	Logical Unit Number	RAID Level	No. Drives	Disk Size (MB)	Files
SCSI3:0:1:6	A	LUN0	5	10 striped	152,426	Backup
SCSI3:0:1:6	A	LUN2	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI3:0:1:6	A	LUN3	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI3:0:1:6	A	LUN4	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/20 hist; 1/30 dist, item, ware,
SCSI3:0:1:6	A	LUN5	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/20 hist; control1
SCSI3:0:1:6	A	LUN6	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem;
SCSI3:0:1:6	A	LUN7	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI3:0:3:6	B	LUN1	5	10 striped	152,426	Backup
SCSI3:0:3:6	B	LUN8	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI3:0:3:6	B	LUN9	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI3:0:3:6	B	LUN10	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/20 hist; 1/30 dist, item, ware,
SCSI3:0:3:6	B	LUN11	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/20 hist
SCSI3:0:3:6	B	LUN12	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI3:0:3:6	B	LUN13	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI4:0:1:6	A	LUN0	5	10 striped	152,426	Backup
SCSI4:0:1:6	A	LUN2	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI4:0:1:6	A	LUN3	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI4:0:1:6	A	LUN4	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/20 hist; 1/30 dist, item, ware,
SCSI4:0:1:6	A	LUN5	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/20 hist
SCSI4:0:1:6	A	LUN6	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI4:0:1:6	A	LUN7	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI4:0:2:0	B	LUN1	5	10 striped	152,426	Backup
SCSI4:0:2:0	B	LUN8	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI4:0:2:0	B	LUN9	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp

External Disk Configuration						
Physical ID	Storage Processor	Logical Unit Number	RAID Level	No. Drives	Disk Size (MB)	Files
SCSI4:0:2:0	B	LUN10	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/20 hist; 1/30 dist, item, ware,
SCSI4:0:2:0	B	LUN11	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/20 hist
SCSI4:0:2:0	B	LUN12	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI4:0:2:0	B	LUN13	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI5:0:0:0	A	LUN0	5	10 striped	152,426	Backup
SCSI5:0:0:0	A	LUN2	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI5:0:0:0	A	LUN3	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI5:0:0:0	A	LUN4	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/20 hist; 1/30 dist, item, ware,
SCSI5:0:0:0	A	LUN5	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/20 hist
SCSI5:0:0:0	A	LUN6	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI5:0:0:0	A	LUN7	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI5:0:2:0	B	LUN1	5	10 striped	152,426	Backup
SCSI5:0:2:0	B	LUN8	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI5:0:2:0	B	LUN9	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI5:0:2:0	B	LUN10	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/20 hist; 1/30 dist, item, ware,
SCSI5:0:2:0	B	LUN11	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/20 hist
SCSI5:0:2:0	B	LUN12	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI5:0:2:0	B	LUN13	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI6:0:0:0	A	LUN0	5	10 striped	152,426	Backup
SCSI6:0:0:0	A	LUN2	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI6:0:0:0	A	LUN3	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI6:0:0:0	A	LUN4	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/20 hist; 1/30 dist, item, ware,
SCSI6:0:0:0	A	LUN5	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/20 hist
SCSI6:0:0:0	A	LUN6	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI6:0:0:0	A	LUN7	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem

External Disk Configuration

Physical ID	Storage Processor	Logical Unit Number	RAID Level	No. Drives	Disk Size (MB)	Files
SCSI6:0:2:0	B	LUN1	5	10 striped	152,426	Backup
SCSI6:0:2:0	B	LUN8	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI6:0:2:0	B	LUN9	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/16 temp
SCSI6:0:2:0	B	LUN10	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/20 hist; 1/30 dist, item, ware,
SCSI6:0:2:0	B	LUN11	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/20 hist
SCSI6:0:2:0	B	LUN12	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI6:0:2:0	B	LUN13	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI7:0:0:0	A	LUN0	5	10 striped	152,426	Backup
SCSI7:0:0:0	A	LUN2	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; roll
SCSI7:0:0:0	A	LUN3	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr
SCSI7:0:0:0	A	LUN4	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/20 hist; 1/30 dist, item, ware,
SCSI7:0:0:0	A	LUN5	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/20 hist
SCSI7:0:0:0	A	LUN6	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI7:0:0:0	A	LUN7	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI7:0:2:0	B	LUN1	5	10 striped	152,426	Backup
SCSI7:0:2:0	B	LUN8	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr
SCSI7:0:2:0	B	LUN9	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr
SCSI7:0:2:0	B	LUN10	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 ordr; 1/20 hist; 1/30 dist, item, ware,
SCSI7:0:2:0	B	LUN11	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/20 hist
SCSI7:0:2:0	B	LUN12	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI7:0:2:0	B	LUN13	0	5 striped	84,681	1/60 stok, cust, ordl, icust2, iord1, iord2, idist, istok, iware; 1/30 nord; 1/30 dist, item, ware, iitem
SCSI8:0:1:0	A	LUN0	1/0	10 mirrored	152,426	Log1
SCSI8:0:1:0	A	LUN1	1/0	10 mirrored	152,426	
SCSI8:0:2:0	B	LUN2	1/0	10 mirrored	152,426	Log2
SCSI8:0:2:0	B	LUN3*	1/0	10 mirrored	152,426	

* = not configured for measurement run - see auditor's letter.

5. Clause 5: Performance Metrics & Response Time

5.1. Measured Throughput (tpmC)

Measured tpmC must be reported.

The measured tpmC was 41,085.43.

5.2. Response Times

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

Table 5.1: Response Time Data

Transaction	Average	Maximum	90th %ile
New-Order	0.33	6.97	0.46
Payment	0.25	5.32	0.37
Delivery	0.12	1.82	0.12
Stock-Level	0.31	5.48	0.59
Order Status	0.31	1.84	0.43
Menu	0.12	4.51	0.12
Delivery (Deferred)	0.30	1.97	0.44

5.3. Keying and Think Times

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

Table 5.2: Keying Times

Transaction	Minimum	Average	Maximum
New-Order	18.18	18.18	19.60
Payment	3.03	3.03	4.45
Delivery	2.02	2.02	3.26
Stock-Level	2.02	2.02	2.74
Order Status	2.02	2.02	3.01

Table 5.3: Think Times

Transaction	Minimum	Average	Maximum
New-Order	0.00	12.14	121.51
Payment	0.00	12.16	121.51
Delivery	0.00	5.10	51.11
Stock-Level	0.00	5.11	51.11
Order Status	0.00	10.15	101.73

5.4. Response Time Frequency Distribution Curves

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

Figure 5.1: New Order Response Time Distribution

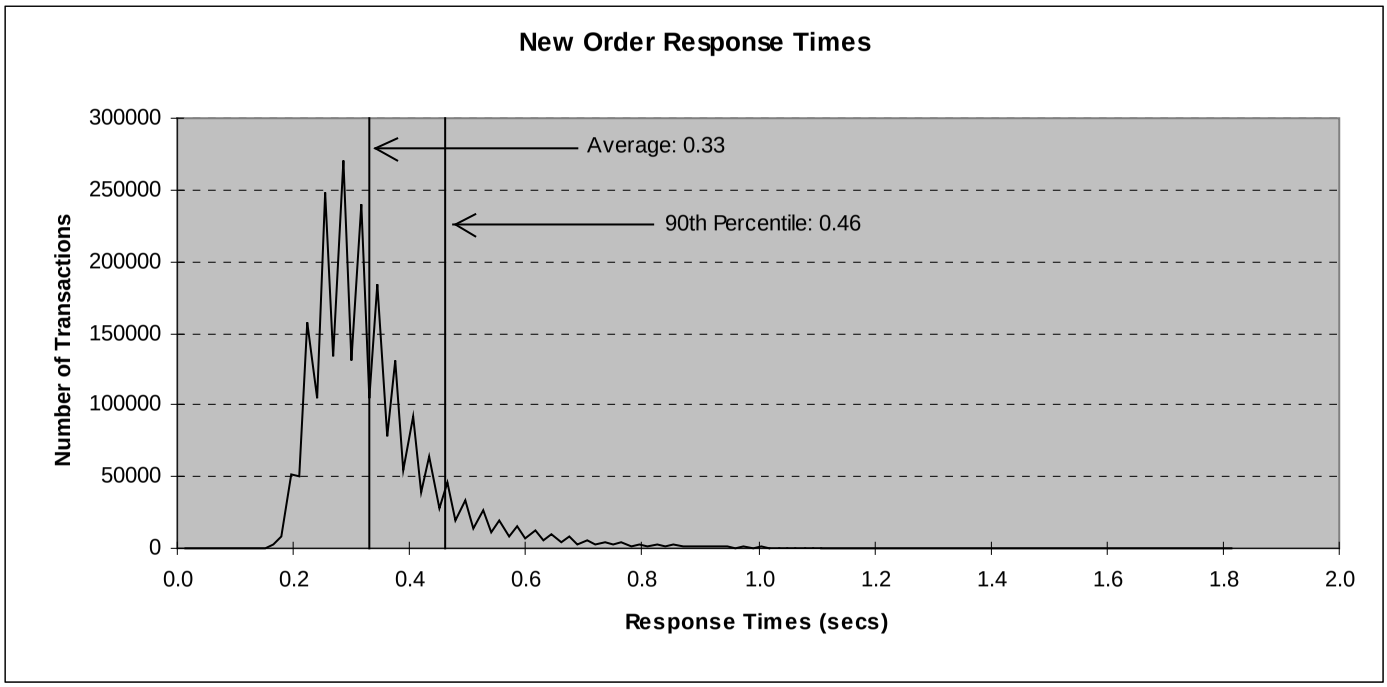


Figure 5.2: Payment Response Time Distribution

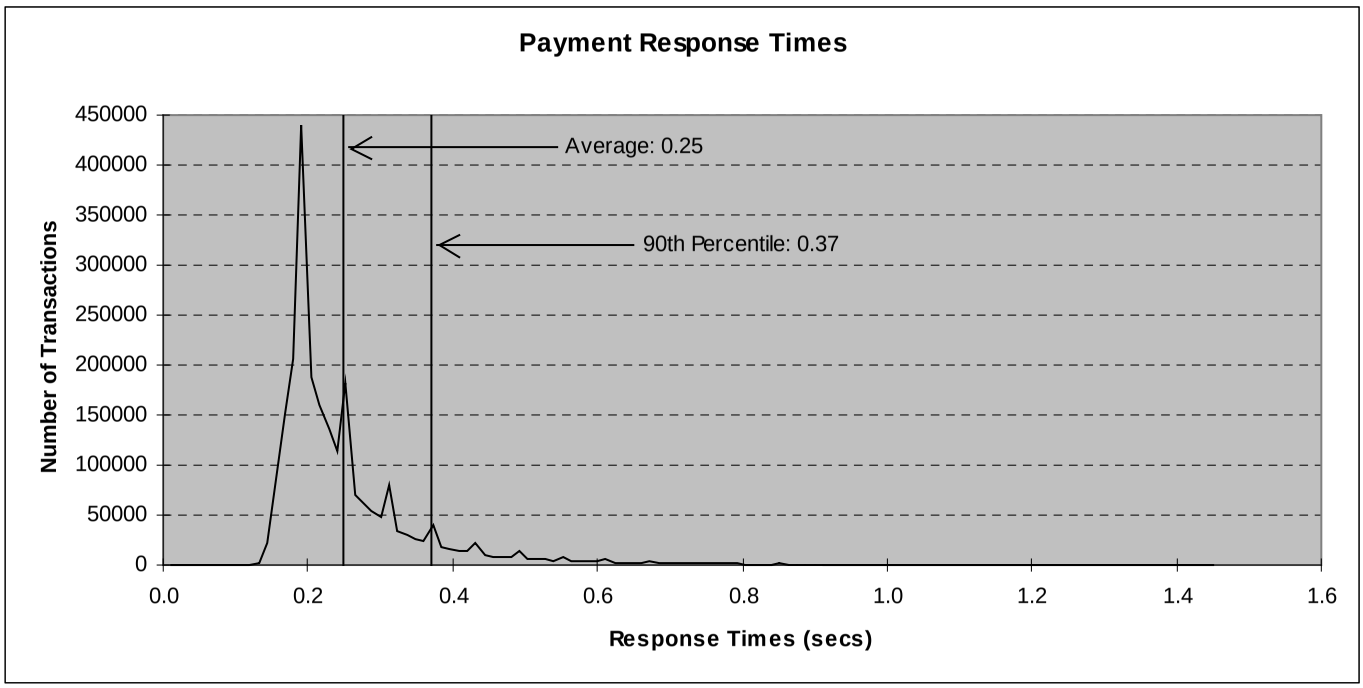


Figure 5.3: Order Status Response Time Distribution

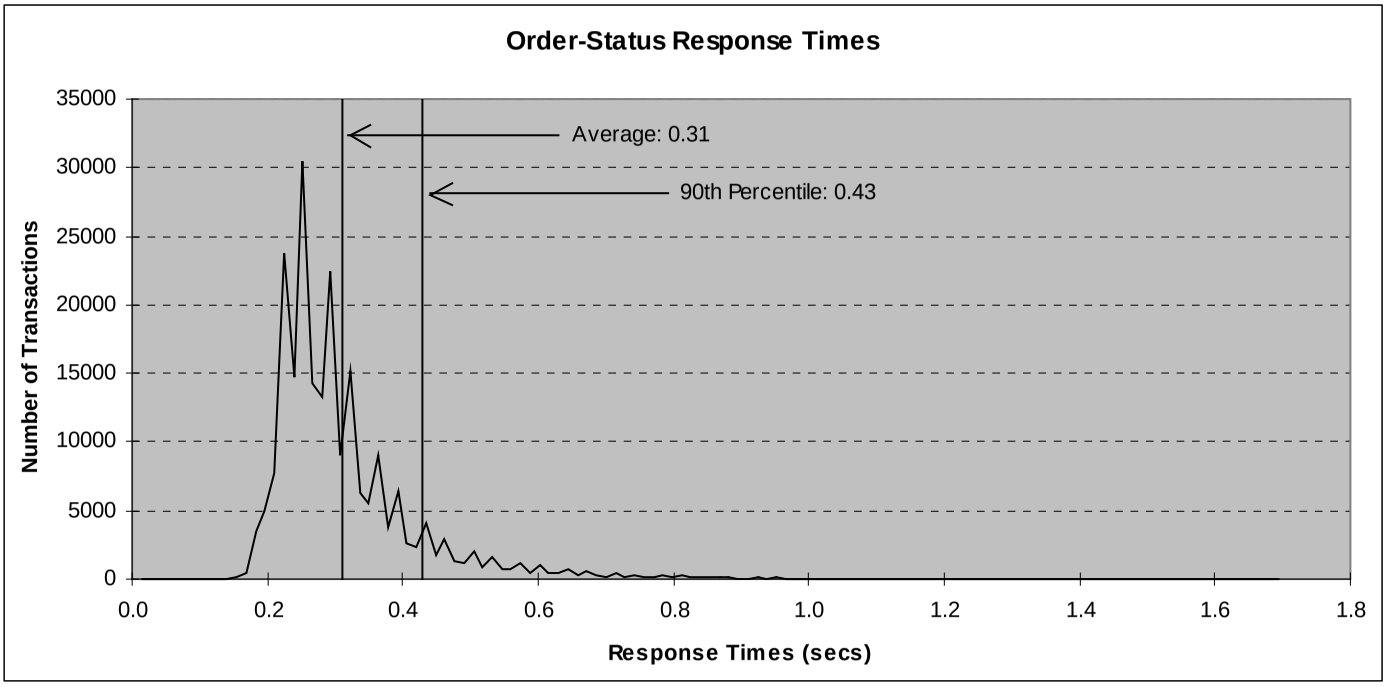


Figure 5.4: Delivery Response Time Distribution

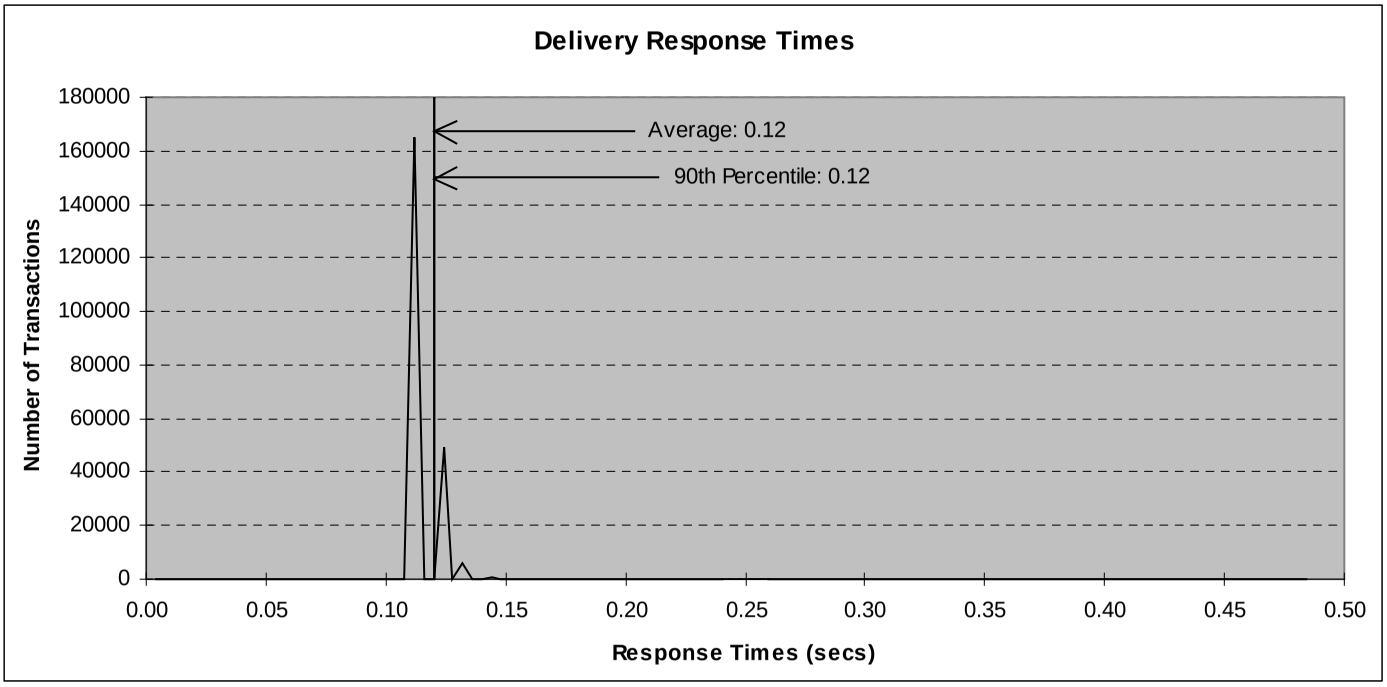
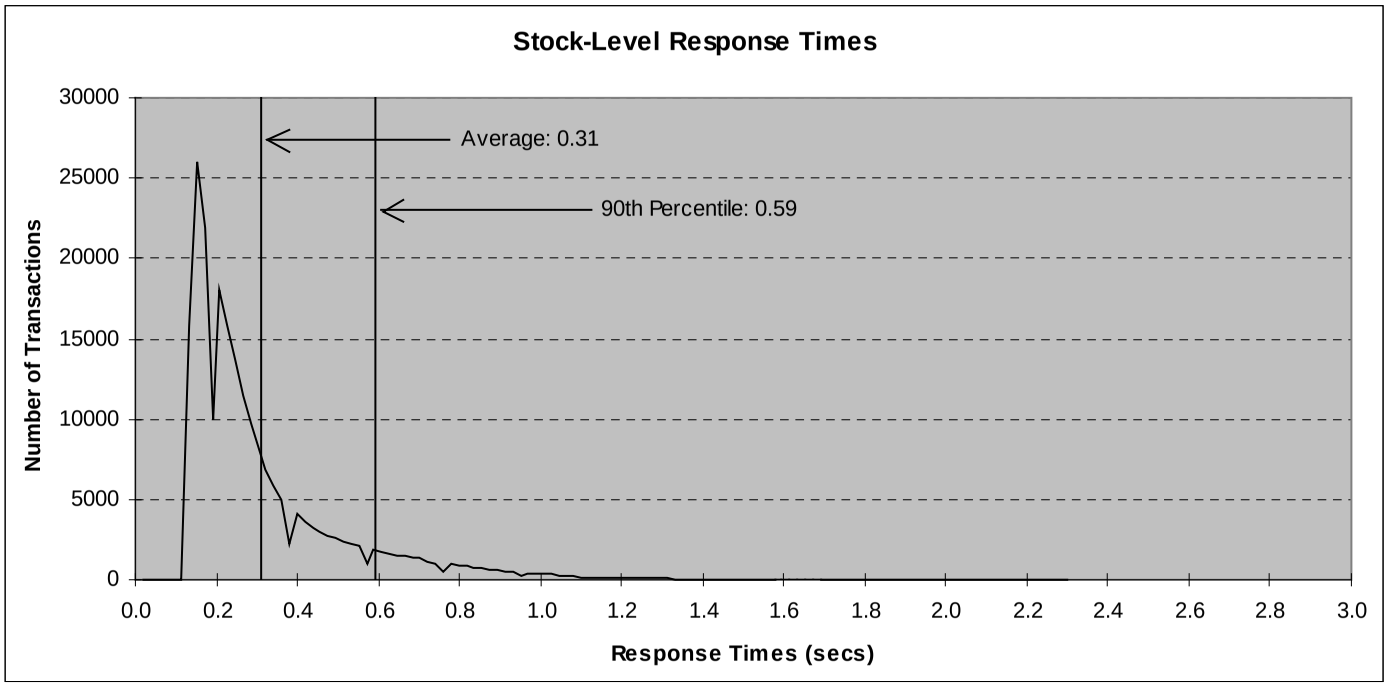


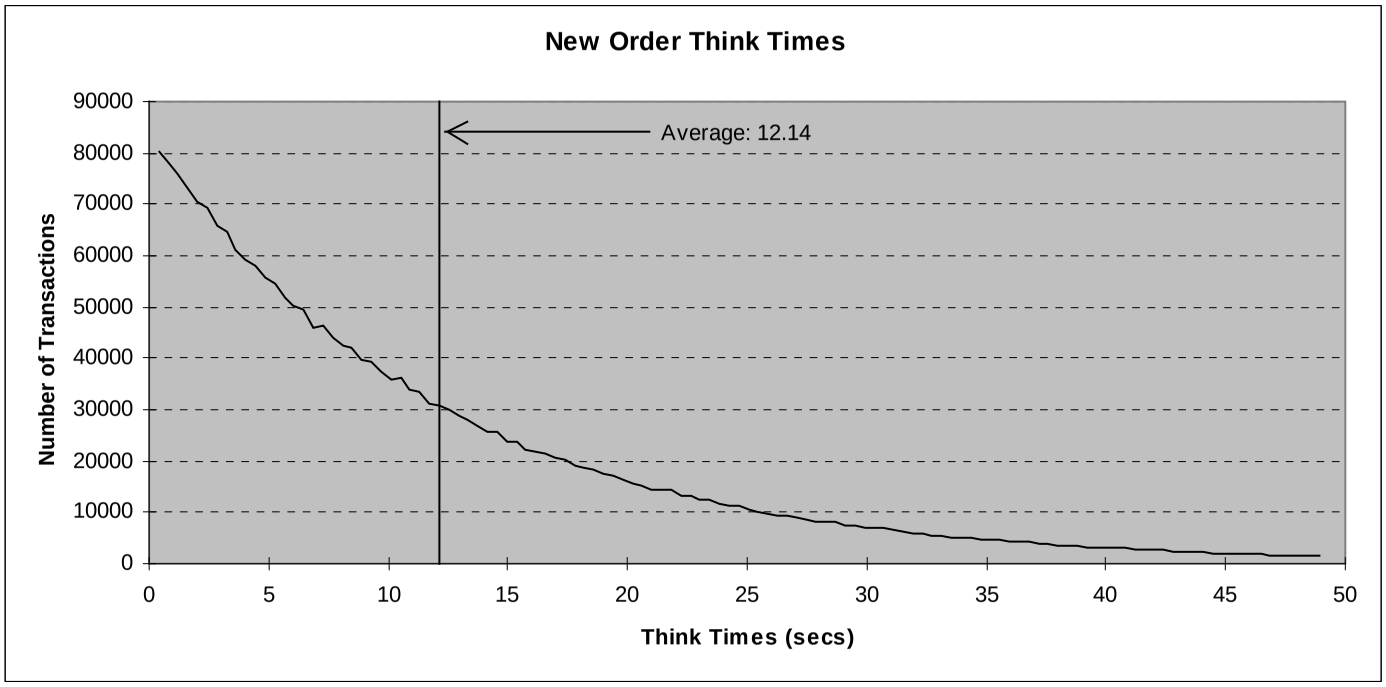
Figure 5.5: Stock Level Response Time Distribution



5.5. New Order Think Time Frequency Distribution Curve

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

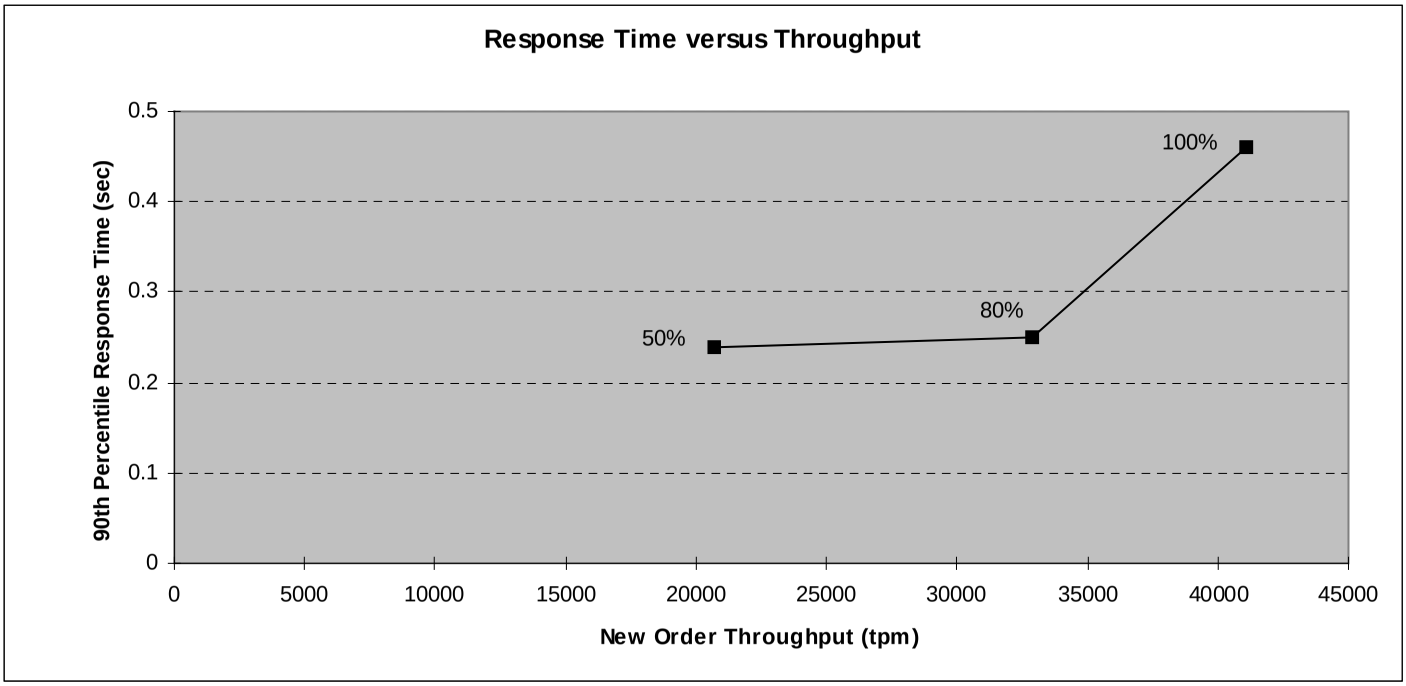
Figure 5.6: New Order Think Time Distribution



5.6. Response Time versus Throughput Performance Curve

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

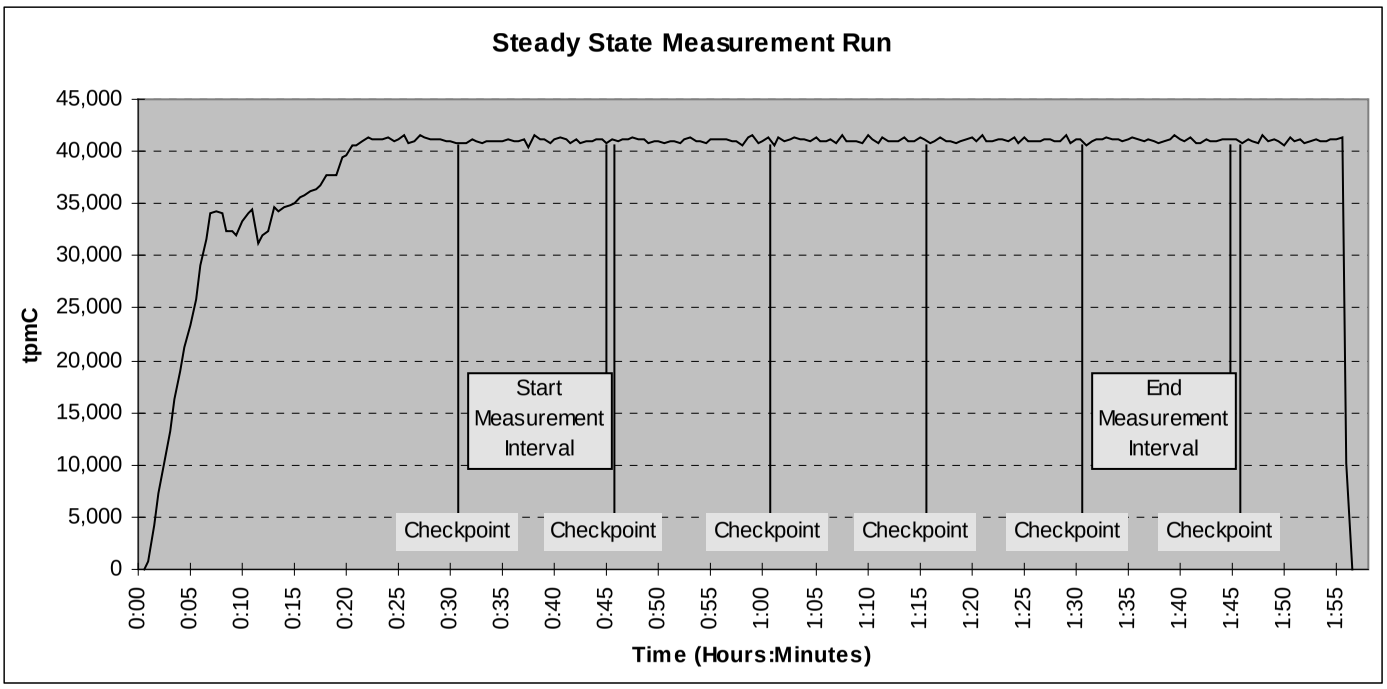
Figure 5.7: Response Time versus Throughput



5.7. New-Order Throughput vs. Time

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

Figure 5.8: Throughput (tpmC) versus Time



5.8. Determination of "Steady State"

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.

The transaction throughput rate (tpmC) and response time were relatively constant after the initial ramp-up period. The throughput and response time behavior were determined by examining data reported for each 30-second interval over the duration of the benchmark. Ramp-up and steady state are discernible in the graph presented in Figure 5.8.

5.9. Work Performed During Steady State

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

The RTE selects a transaction type from the menu and prepares to request the appropriate blank form. A timestamp is taken before the form request is sent and after the response is returned. The difference between the two is saved off as the menu response time. The RTE then generates input data for the transaction to create a completed form and waits the appropriate key time. A timestamp is taken before the completed form is sent and after the response is returned. The difference between these two is saved off as the transaction response time. Both response times are padded with a 0.1 second delay per spec to account for the web browser delay. The appropriate transaction data and response times are logged and the RTE waits the required think time interval before repeating the process. Each RTE driver maintains its own log file. Log file contents are consolidated for the reports.

The RTE emulates web browsers (not terminals) in this client-server implementation. The RTE sends and receives HTML formatted data using HTTP through Ethernet LANs to a client application running on the client machine. The client application (tpcc.dll) processes the request, sends the transaction to a Tuxedo TPC-C application server queue, waits for the transaction response, and returns an appropriately formatted HTML form back to the (emulated) web browser (RTE). The Tuxedo TPC-C application server retrieves a message from its queue, processes the request using Oracle OCI calls to the database, and returns a result to the client application (via Tuxedo).

Checkpoints were initiated at specified intervals via a shell script. The start and end time of each checkpoint was recorded. Oracles's logfiles were configured large enough to not cause an additional checkpoint within this specified enterval. The checkpoint script was started manually on the server after the RTE had all users logged in and sending transactions and a steady state had been achieved.

At each checkpoint, Oracle wrote to disk all database pages in memory that had been updated but not yet physically written to the disk. Upon completion of the checkpoint, Oracle also wrote records to the transaction log indicating that a checkpoint had completed.

5.10. Reproducibility

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

In a repeat test, carried out in the same manner as the primary test, a throughput of 41,010.06 tpmC was achieved on the same database during a 60-minute, steady state run. All required transaction statistics were met. See the Auditor's attestation letter for details.

5.11. Measurement Interval Duration

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

The measurement interval was 60 minutes.

5.12. Regulation of Transaction Mix

The method of regulation of the transaction mix (e.g. card decks or weighted random distribution) must be described. If weighted distribution is used and the RTE adjusts the weights associated with each transaction type, the maximum adjustments to the weight from the initial value must be disclosed.

The RTE was given a weighted random distribution that could not be adjusted during the run.

5.13. Transaction Statistics

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

The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed.

The average number of order-lines entered per New-Order transaction must be disclosed.

The percentage of remote order-lines entered per New-Order transaction must be disclosed.

The percentage of remote Payment transactions must be disclosed.

The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed.

The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed.

Table 5.4 shows this information.

Table 5.4: Transaction Statistics

Transaction Type	Statistics	Value
New Order	Rolledback transactions	1.01%
	Home warehouse	99.00%
	Remote warehouse	1.00%
	Average Items per Order	10.00
Payment	Home warehouse	85.00%
	Remote warehouse	15.00%
	Non-primary key access	60.01%
Order Status	Non-primary key access	59.93%
Delivery	Skipped transactions (Interactive)	0
	Skipped transaction counts (Deferred)	0
	Skipped District counts (Deferred)	0
Transaction Mix	New Order	44.84%
	Payment	43.06%
	Delivery	4.02%
	Stock-Level	4.03%
	Order-Status	4.03%

5.14. Checkpoint Statistics

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.

There are four checkpoints in the measurement interval. The first checkpoint starts 46 seconds into the measurement interval. The checkpoint interval is 15 minutes (from the start of one to the start of the next) and a checkpoint lasts approximately 8.0 minutes. In conformance with Clause 5.5.2.2, no guard zones were required since four checkpoints started and completed inside the measurement interval.

6. Clause 6: SUT, Driver & Communications Definition

6.1. Remote Terminal Emulator (RTE) Description

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

The RTE used is proprietary to Unisys. Appendix D contains the profile used as input to this RTE.

6.2. 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.

There were no emulated components in the benchmark configuration other than the emulated web browsers on the users' PCs.

6.3. Functional Diagrams

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

Section 0.7 describes and shows functional diagrams of the benchmarked and priced systems.

6.4. Network Configuration

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

Figures 0.1 and 0.2 in Section 0.7 also diagram the network configurations of the benchmark and configured systems and represent the RTEs connected via LAN replacing the user PCs that are directly connected via LAN.

6.5. Network Bandwidth

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

Ethernet local area networks (LAN) are used in the priced and tested configurations. The database server (SUT) contains a single 10/100 megabit per second LAN adapter. This LAN segment runs at 100 megabits per second full duplex in both the priced and tested configurations. The Tuxedo client contains two 10/100 megabit per second LAN adapters. Both LAN segments run at 100 megabits per second full duplex in the priced and tested configurations. The web client contains two 10/100 megabit per second LAN adapters. Both LAN segments run at 100 megabits per second full duplex in the priced and tested configurations. 36 (9 per client) user LAN segments run at 10 megabits per second half duplex in both the priced and tested configurations. Two eight port 10/100 megabit per second switches were concatenated and used to connect the web clients to users. An eight port 10/100 megabit per second switch was

used to connect the web clients to the Tuxedo client. An additional eight port 10/100 megabit per second switch was used to connect the Tuxedo client to the database server. In the priced configuration, the web client is connected to workstations (PCs running web browsers). In the tested configuration, the web client is connected to RTE driver systems emulating web browsers.

6.6. Operator Intervention

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

No operator intervention was required to sustain operation at the reported throughput.

7. Clause 7: Pricing

7.1. Pricing

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

The total 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.

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.

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

A detailed list of hardware and software components along with their part numbers and prices are given in the Executive Summary near the beginning of this document. The UnixWare 7.1.1 software has been supplemented by ptf7139a, ptf7442b, ptf7601b, ptf7603b, ptf7608b, ptf7619a, ptf7620a, and ptf7621a, which will be available from the SCO Support web site at www.sco.com.

7.1.1. System Pricing

Each priced configuration consists of an integrated system package, additional options, and components. Prices for all products are US list prices. A three year warranty is standard with this class of Unisys server products.

7.1.2. Maintenance Pricing

The five year support pricing for Unisys Corporation Enterprise Server products is based on a 36-month warranty on hardware, upgraded to service level Performance-Gold, plus an additional 24 months of support at service level Performance-Gold. Oracle, BEA and Microsoft support pricing is based on 5 years of annual support costs.

Unisys Corporation Standard Performance-Gold Support: four hour maximum response, onsite support for hardware provides service from 8:00 A.M. to 5:00 P.M., Monday through Friday. Service requests made as late as 5:00 P.M. will receive a response the same day.

Server disks are covered by Western Micro's 5 year, seven day return-to-factory warranty, and appropriate spares are included in the priced configuration. Netlux and Sofware House provide 5 year, seven day return-to-factory warranties, and appropriate spares are included in the priced configuration.

7.1.3. Discounts

Unisys provides a standard pre-pay discount for maintenance service of the client, server and storage components of the priced configuration.

Western Micro provides a standard dollar-volume discount to the client, server and storage components of the priced configuration.

7.2. Availability

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

The hardware, software and support/maintenance products priced in this benchmark are detailed on page vi.

Oracle 8i, Version 8.1.6.1 will be available by June 1, 2000. All hardware components are available now.

7.3. Measured tpmC, Pricing, Price/Performance, and Availability Date

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

The measured tpmC, plus pricing calculations, price/performance, and availability are shown on pages v and vi.

7.4. 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.

None.

7.5. Usage Pricing

For any usage pricing, the sponsor must disclose:

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

The component pricing based on usage is shown below:

- One (1) Oracle 8i Enterprise Edition license
- One (1) UnixWare 7 Data Center Edition license
- One (1) UnixWare 7 Business Edition license
- One (1) BEA Tuxedo 6.5 CFS for UnixWare license
- Four (4) BEA Tuxedo 6.4 CFS for NT licenses
- Four (4) Microsoft Windows NT Server 4.0 licenses

Oracle 8i and BEA Tuxedo were priced for an unlimited number of users.

8. Clause 8 : Full Disclosure Availability

8.1. Availability

The Full Disclosure Report must be readily available to the public at a reasonable charge, similar to charges for similar documents by that test sponsor.

Copies of this Full Disclosure Report may be downloaded from the Transaction Processing Performance Council web site at www.tpc.org or obtained by contacting:

TPC Benchmark Administrator
Systems Analysis, Modeling & Measurement Group
Unisys Corporation, M/S 262
25725 Jeronimo Road
Mission Viejo, CA 92691
USA

9. *Clause 9 : Audit*

9.1. Auditor's Report

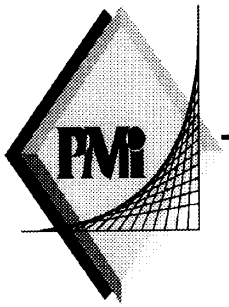
The auditor's name, address, phone number and a copy of the auditor's attestation letter indicating compliance must be included in the Full Disclosure Report.

This implementation of the TPC Benchmark C on the Unisys e-@ction ES2085R Enterprise Server was audited by Tom Sawyer, a TPC certified auditor of:

Performance Metrics, Inc.
137 Yankton St. Suite 101
Folsom, CA 95630

Phone: (916) 985-1131
Fax: (916) 985-1185
e-mail: Lorna@PerfMetrics.com

The attestation letter is shown on the next 2 pages.



PERFORMANCE METRICS INC.
TPC Certified Auditors

December 10, 1999

Jerrold Buggert
Director of Modeling and Measurement
Unisys Corporation
25725 Jeronimo Road
Mission Viejo, CA 92691

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

Platform:	Unisys e-@ction ES2085R Enterprise Server
Database Manager:	Oracle 8i Version 8.1.6
Server Operating System:	UnixWare 7 Data Center Edition Version 7.1.1
Client Operating Systems:	UnixWare 7 Business Edition Version 7.1.1
	Windows NT Server 4.0, IIS 3.0
Transaction Manager:	BEA TUXEDO CFS 6.5 for UnixWare & 6.4 for NT

Server: e-@ction ES2085R Enterprise Server				
CPU's	Memory	Disks	90% Response	tpmC
8 Pentium III Xeon @ 550 MHz	Main: 16 GB Cache: 2MB each	2 @ 4.3 GB 440 @ 18GB	**0.46 sec**	**41,085.43**
4 Web Clients: NetServer LC3				
2 Pentium III @ 500 MHz	Main: 1024 MB Cache: 512K	1 @ 4 GB	na	na
1 Tuxedo Client: e-@ction ES2023 Enterprise Server				
4 Pentium III Xeon @ 500 MHz	Main: 1024 MB Cache: 2MB	1 @ 4 GB	na	na

137 Yankton St. Suite 101, Folsom 95630
(916) 985-1131 fax: (916) 985-1185 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 was properly sized and populated.
- The database was properly scaled with 3,500 warehouses. There were 3,300 warehouses used in the measurement. I verified that d_next_o_id and w_ytd did not change for the unused warehouses.
- The ACID properties were met.
- The durability data loss and log loss tests were performed on the measured database.
- Input data was generated according to the specified percentages.
- Eight hours of dynamic table growth space was configured on the measured system.
- The 180-day space calculation was verified. The measured configuration has sufficient storage to satisfy this requirement.
- The measured system had an additional 100 18GB drives that were used for backup. I verified that these drives were inactive during the measurement. The cabinet for log drives had 10 extra drives that were not logically configured or visible to the OS during the measurement – see Auditor Notes.
- Measurement cycle times included a 0.1 second menu and a 0.1 second response time delay for an emulated Web browser.
- There were 33,000 user contexts present on the system.
- Each emulated user started with a different random number seed.
- The NURand constants used for database load and at run time were 1 and 87.
- The steady state portion of the test was 60 minutes.
- Four checkpoints were taken during the measurement interval.
- The system pricing was checked for major components and maintenance.

Auditor Notes:

Previous runs had shown no need for the additional log disks and I had requested that they be deconfigured to avoid any dispute that they might have been used. Subsequent runs demonstrated that the higher performance achieved would require the drives which were added to the priced configuration. An additional performance run was made with the connected drives which was 0.11% slower than the measured result.

The database binaries and the UnixWare binaries on the database server were each on a separate 4GB disk. These are combined on one 9GB disk in the priced system. I verified that there was little to no usage of these drives during the measurement.

Sincerely,

Tom Sawyer

Tom Sawyer
Auditor

137 Yankton St. Suite 101, Folsom 95630
(916) 985-1131 fax: (916) 985-1185 email: Lorna@PerfMetrics.com

Appendix A - Client/Server Source

Web Client

tpcc.def

```
EXPORTS
   GetExtensionVersion
   HttpExtensionProc
```

tpcc.h

```
// tpcc.h

#include <time.h>

// TPCCHandler return codes
#define TPCCSEND 1
#define TPCCSENDEND 2
#define TPCCENDNOW 3

// TPCC Service return codes
#define SVC_BADITEMID 1
#define SVC_NOERROR 0
#define SVCERR_DEADLOCK -1
#define SVCERR_NOCUSTOMER -2
#define SVCERR_NOORDERS -3
#define SVCERR_DBLIB -4
#define SVCERR_EXCEPTION -5
#define SVCERR_DQFULL -6
#define SVCERR_DQSTART -7

// Min/Max transaction data definitions
#define MIN_DId 1
#define MAX_DId 10
#define MIN_OL 5
#define MAX_OL 15
#define MIN_QUANTITY 1
#define MAX_QUANTITY 10
#define MIN_ITEM_ID 1
#define MAX_ITEM_ID 100000
#define MIN_CUST_ID 1
#define MAX_CUST_ID 3000
#define MIN_CARRIER 1
#define MAX_CARRIER 10
#define MIN_THRESHOLD 10
#define MAX_THRESHOLD 20

// pTPCC->iStatusId codes
#define INVALID_IID              1
#define STATUS_OK                0
#define ERR_CMD_UNKNOWN          -10
#define ERRTXT_CMD_UNKNOWN       "Unrecognized Command"
#define ERR_ALREADY_LOGGEDIN     -11
#define ERRTXT_ALREADY_LOGGEDIN  "Already Logged In"
#define ERR_TERMID               -12
#define ERRTXT_TERMID            "TermId or SyncId in Error"
#define ERR_FORM_UNKNOWN         -13
#define ERRTXT_FORM_UNKNOWN      "Unrecognized FormId"
#define ERR_WID_INVALID          -14
#define ERR_DID_INVALID          -15
#define ERR_MISSING_KEY          -16
#define ERR_NOT_NUMERIC          -17
#define ERR_THRESHOLD_RANGE      -18
#define ERR_EMBEDDED_EMPTY_OL    -19
#define ERR_QUANTITY_INVALID     -20
#define ERR_OL_INVALID           -21
#define ERR_OL_COUNT             -22
#define ERR_TM_INTERFACE         -23
#define ERR_SERVICE_RSLT         -24
#define ERR_INPUT_TOOLONG        -25
#define ERR_IDANDNAME_EMPTY      -26
#define ERR_IDANDNAME_ENTERED    -27
#define ERR_AMOUNT_BADFORM       -28
#define ERR_AMOUNT_INVALID       -29
#define ERR_CARRIER_INVALID      -30
#define ERR_TERM_ALLOC           -31
#define ERR_DQPOST_RSLT          -32

#define STATUS_LEN 200
#define NAME_LEN 16
#define ADDR_LEN 20
#define STATE_LEN 2
#define ZIP_LEN 9

#define MAX_MSG_SZ  5000
#define CTEXT "Content-length: "
#define HTTPHdr "Connection: keep-alive\r\nContent-type: text/html\r\n" \
                "Content-length:     \r\n\r\n"


#pragma pack(1)
typedef struct
{
   int year;
   int quarter;
   int month;
   int dayofyear;
   int day;
   int week;
   int weekday;
   int hour;
   int minute;
   int second;
   int millisecond;
} DBDATEREC;

typedef struct
{
   short ol_supply_w_id;
```

```
   long ol_i_id;
   char ol_i_name[25];
   short ol_quantity;
   char ol_brand_generic[2];
   double ol_i_price;
   double ol_amount;
   short ol_stock;
} OL_NEW_ORDER_DATA;

typedef struct
{
   short w_id;
   short d_id;
   long        c_id;
   short o_ol_cnt;
   char c_last[NAME_LEN + 1];
   char c_credit[3];
   double c_discount;
   double w_tax;
   double d_tax;
   long o_id;
   short o_commit_flag;
   DBDATEREC o_entry_d;
   short o_all_local;
   double total_amount;
   char execution_status[STATUS_LEN];
   OL_NEW_ORDER_DATA Ol[MAX_OL];
} NEW_ORDER_DATA;

typedef struct
{
   short w_id;
   short d_id;
   long c_id;
   short c_d_id;
   short c_w_id;
   double h_amount;
   DBDATEREC h_date;
   char w_street_1[ADDR_LEN + 1];
   char w_street_2[ADDR_LEN + 1];
   char w_city[ADDR_LEN + 1];
   char w_state[STATE_LEN + 1];
   char w_zip[ZIP_LEN + 1];
   char d_street_1[ADDR_LEN + 1];
   char d_street_2[ADDR_LEN + 1];
   char d_city[ADDR_LEN + 1];
   char d_state[STATE_LEN + 1];
   char d_zip[ZIP_LEN + 1];
   char c_first[NAME_LEN + 1];
   char c_middle[3];
   char c_last[NAME_LEN + 1];
   char c_street_1[ADDR_LEN + 1];
   char c_street_2[ADDR_LEN + 1];
   char c_city[ADDR_LEN + 1];
   char c_state[STATE_LEN + 1];
   char c_zip[ZIP_LEN + 1];
   char c_phone[16];
   DBDATEREC c_since;
   char c_credit[3];
   double c_credit_lim;
   double c_discount;
   double c_balance;
   char c_data[200+1];
   char execution_status[STATUS_LEN];
} PAYMENT_DATA;

typedef struct
{
   long ol_i_id;
   short ol_supply_w_id;
   short ol_quantity;
   double ol_amount;
   DBDATEREC ol_delivery_d;
} OL_ORDER_STATUS_DATA;

typedef struct
{
   short w_id;
   short d_id;
   long c_id;
   char c_first[NAME_LEN + 1];
   char c_middle[3];
   char c_last[NAME_LEN + 1];
   double c_balance;
   long o_id;
   DBDATEREC o_entry_d;
   short o_carrier_id;
   OL_ORDER_STATUS_DATA OlOrderStatusData[MAX_OL];
   short o_ol_cnt;
   char execution_status[STATUS_LEN];
} ORDER_STATUS_DATA;

typedef struct
{
   short w_id;
   short o_carrier_id;
   long o_id[10];
   int iComplete;
   SYSTEMTIME QTime;                // time delivery was queued
   SYSTEMTIME EndTime;              // time delivery completed
   char execution_status[STATUS_LEN];
} DELIVERY_DATA;

typedef struct
{
   short w_id;
   short d_id;
   short thresh_hold;
   long low_stock;
   char execution_status[STATUS_LEN];
} STOCK_LEVEL_DATA;

typedef struct
{
   LPVOID ConnID;                   // Active Connection Id
   SHORT sWId;                      // TPCC WareHouse Id
   SHORT sDId;                      // TPCC District Id
   INT  iSyncId;                    // TPCC Sync Id
   INT  iTermId;                    // TPCC Term Id
   UINT uFormId;                    // TPCC Form Id
   INT  iStatusId;                  // TPCC Status Id
   CHAR ErrTxt[500];                // Error text
```

```
   CHAR szWork[200];                    // Thread work area
   CHAR szHeader[100];                  // HTTP work area
   CHAR * RecvMsg;                      // HTML message from ECB
   CHAR SendMsg[MAX_MSG_SZ];            // HTML work area
   TMON_STATE tsTMon;                   // TMon Interface
} TPCC_STATE;
#pragma pack()
```

tpcc.c

```
// tpcc.c
//
// Copyright Unisys, 1997
//
#include <windows.h>
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <winreg.h>
#include <httpext.h>

#include "tmon.h"
#include "diagio.h"
#include "term.h"
#include "tpcchandler.h"
#include "delivery.h"

#define EXTN_VERSION MAKELONG(HSE_VERSION_MINOR,HSE_VERSION_MAJOR)
#define TLS_NULL 0xFFFFFFFF
DWORD dwTlsInx;
CHAR * pTitle = "IIS TPCC DLL";
CRITICAL_SECTION csDllMain;

// Diagnostic logging settings
BOOL bEventLog = TRUE;
BOOL bConsole = FALSE;
UINT uDiagLevel = DIAG_INFO;

// TMon Interface Settings
INT iTMMaxMsg = 0;

// Term Interface Settings
INT iMaxTerms = 3000;

// Delivery Settings
long lSetDThreads = 8;
long lSetDQSize = DEFAULTDQSIZE;
char szSetPath[200] = "\\tuxedo\\runtime\\";

static CHAR * szTPCCError =
   HTTPHdr "<HTML>"
   "<HEAD><TITLE>Welcome To TPC-C</TITLE></HEAD><BODY>"
   "<B>TPCC Extension Error (TPCC Array Not Allocated)</B><BR>"
   "</BODY></HTML>";

static CHAR * szTMInitError =
   HTTPHdr "<HTML>"
   "<HEAD><TITLE>Welcome To TPC-C</TITLE></HEAD><BODY>"
   "<B>TPCC Extension Error (TMInit Failed)</B><BR>"
   "</BODY></HTML>";
INT iHHdrLen = 0;
INT iCTextLen = 0;

BOOL ThreadAttach(TPCC_STATE * pTPCC,CHAR * pDiag);
VOID ThreadDetach(TPCC_STATE * pTPCC);
VOID SendResponse(EXTENSION_CONTROL_BLOCK * pECB,CHAR * pMsg,CHAR *
pWork);
BOOL ReadRegistry(VOID);

//====================================================================
//
// Function name: DllMain
//
//====================================================================
BOOL APIENTRY DllMain(HANDLE hInst, ULONG ul_reason_for_call,
                      LPVOID lpReserved)
{
   TPCC_STATE * pTPCC = NULL;
   CHAR szDiag[MAX_DIAG_SZ];
   UINT iTMMaxSz =0;
   switch(ul_reason_for_call)
   {
      case DLL_PROCESS_ATTACH:
         // Process initialization

         InitializeCriticalSection(&csDllMain);
         ReadRegistry();
         DiagIoInit(pTitle,bConsole,bEventLog,uDiagLevel);
         sprintf(szDiag,
            "EventLog = %d, Console = %d, DiagLevel = %d\n"
            "MaxTerms = %d\n",
            bEventLog,bConsole,uDiagLevel,iMaxTerms);
         DiagIoWrite(szDiag,DIAG_FORCE);
         dwTlsInx = TlsAlloc();
         if (dwTlsInx == TLS_NULL)
         {
            sprintf(szDiag,"PAttach(%ld): Tls Alloc Failed (%ld)\n",
               GetCurrentThreadId(),GetLastError);
            DiagIoWrite(szDiag,DIAG_ERROR);
            return(FALSE);
         };
         if (TermInit(iMaxTerms))
            return(FALSE);
         iTMMaxSz = max(iTMMaxSz,sizeof(NEW_ORDER_DATA));
         iTMMaxSz = max(iTMMaxSz,sizeof(PAYMENT_DATA));
         iTMMaxSz = max(iTMMaxSz,sizeof(ORDER_STATUS_DATA));
         iTMMaxSz = max(iTMMaxSz,sizeof(DELIVERY_DATA));
         iTMMaxSz = max(iTMMaxSz,sizeof(STOCK_LEVEL_DATA));
         iTMMaxSz += 10;
         TMonInit(iTMMaxSz);
         if (DeliveryInit(lSetDThreads,lSetDQSize,szSetPath))
         {
            DeliveryTerm();
            return(FALSE);
         };
         iHHdrLen = strlen(HTTPHdr);
         iCTextLen = strlen(CTEXT);
         break;
      case DLL_THREAD_ATTACH:
         // Move ThreadAttach call to HttpExt since the DllMain call
```

```
         // for Thread Attach did not reliably come before the first
         // call to HttpExtProc.
         break;
               case DLL_THREAD_DETACH:
         ThreadDetach(pTPCC);
                     break;
               case DLL_PROCESS_DETACH:
         ThreadDetach(pTPCC);
         DeleteCriticalSection(&csDllMain);
         DeliveryTerm();
         TMonTerm();
         TermTerm();
         TlsFree(dwTlsInx);
         dwTlsInx = TLS_NULL;
         DiagIoTerm();
                     break;
        };
        return TRUE;
};   // DllMain

//====================================================================
//
// Function name: ThreadAttach
//
// Result:
//    FALSE Thread state structure initialized
//    TRUE  Thread state structure initialization failure
//
//====================================================================
BOOL ThreadAttach(TPCC_STATE * pTPCC,CHAR * pDiag)
{
   BOOL bRslt;
   BOOL bTMRslt;
   INT iTMTry;
   UINT uLabelNoOp;
   EnterCriticalSection(&csDllMain);
   __try
   {
      pTPCC = (TPCC_STATE *) calloc(1,sizeof(TPCC_STATE));
      if (pTPCC == NULL)
      {
         sprintf(pDiag,"ThrAtt(%ld): pTPCC Alloc Failed (%ld)\n",
            GetCurrentThreadId(),GetLastError);
         DiagIoWrite(pDiag,DIAG_ERROR);
         bRslt = TRUE;
         goto TAttachXit;
      };
      TlsSetValue(dwTlsInx,pTPCC);
      pTPCC->tsTMon.pTMData = NULL;
      pTPCC->tsTMon.pszErrTxt = pTPCC->ErrTxt;
      for (iTMTry = 0;iTMTry < 5;iTMTry++)
      {
         Sleep(50);
         bTMRslt = TMInit(&pTPCC->tsTMon);
         if (bTMRslt)
         {
            sprintf(pDiag,"ThrAtt(%ld): TMInit %s\n",
               GetCurrentThreadId(),pTPCC->ErrTxt);
            DiagIoWrite(pDiag,DIAG_ERROR);
         }
         else
            break;
      };
      if (bTMRslt)
      {
         sprintf(pDiag,"ThrAtt(%ld): TMInit Retries(5) Exceeded\n",
            GetCurrentThreadId());
         DiagIoWrite(pDiag,DIAG_ERROR);
         bRslt = TRUE;
         goto TAttachXit;
      };
      sprintf(pDiag,"ThrAtt(%ld): Attach
Complete\n",GetCurrentThreadId());
      DiagIoWrite(pDiag,DIAG_FORCE);
      bRslt = FALSE;
TAttachXit:
      uLabelNoOp = 0;
   }
   __finally
   {
      LeaveCriticalSection(&csDllMain);
   };

   return(bRslt);

};   // ThreadAttach

//====================================================================
//
// Function name: ThreadDetach
//
//====================================================================
VOID ThreadDetach(TPCC_STATE * pTPCC)
{
   EnterCriticalSection(&csDllMain);
   __try
   {
      pTPCC = TlsGetValue(dwTlsInx);
      if (pTPCC != NULL)
      {
         TMDone(&pTPCC->tsTMon);
         free(pTPCC);
         pTPCC = NULL;
         TlsSetValue(dwTlsInx,pTPCC);
      };
   }
   __finally
   {
      LeaveCriticalSection(&csDllMain);
   };
};   // ThreadDetach

//====================================================================
//
// Function name: GetExtensionVersion
//
//====================================================================
BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO *pVersion)
{
   pVersion->dwExtensionVersion = EXTN_VERSION;
   strncpy(pVersion->lpszExtensionDesc,pTitle,HSE_MAX_EXT_DLL_NAME_LEN);
   return TRUE;
```

```
};   // GetExtensionVersion

//====================================================================
//
// Function name: HttpExtensionProc
//
//
// Returns:
//    HSE_STATUS_SUCCESS               send msg, drop connection
//    HSE_STATUS_SUCCESS_AND_KEEP_CONN       send msg, keep connection
//
//====================================================================
DWORD WINAPI HttpExtensionProc(EXTENSION_CONTROL_BLOCK * pECB)
{
   TPCC_STATE * pTPCC;
   DWORD dwRslt = HSE_STATUS_SUCCESS;
   UINT uRslt;

   pTPCC = TlsGetValue(dwTlsInx);
   if (pTPCC == NULL)
   {
      CHAR szWork[200];
      ThreadAttach(pTPCC,szWork);
      pTPCC = TlsGetValue(dwTlsInx);
      if (pTPCC == NULL)
      {
         SendResponse(pECB,szTPCCError,szWork);
         goto HttpXit;
      };
   };
   if (pTPCC->tsTMon.pTMData == NULL)
      SendResponse(pECB,szTMInitError,pTPCC->szHeader);
   TPCCClear(pTPCC);
   pTPCC->ConnID = pECB->ConnID;
   pTPCC->RecvMsg = pECB->lpszQueryString;
   uRslt = TPCCHandler(pTPCC);
   switch (uRslt)
   {
      case TPCCSEND:
         SendResponse(pECB,pTPCC->SendMsg,pTPCC->szHeader);
         dwRslt = HSE_STATUS_SUCCESS_AND_KEEP_CONN;
         break;
      case TPCCSENDEND:
         SendResponse(pECB,pTPCC->SendMsg,pTPCC->szHeader);
         break;
      case TPCCENDNOW:
      default:
         break;
   };   // switch (TPCCHandler result)

HttpXit:

       return(dwRslt);

};   // HttpExtensionProc

//====================================================================
//
// Function name: SendResponse
//
//====================================================================
VOID SendResponse(EXTENSION_CONTROL_BLOCK * pECB,CHAR * pMsg,CHAR * pWork)
{
   DWORD dwMsgBytes;
   CHAR * pCL;
   dwMsgBytes = strlen(pMsg);
   pCL=strstr(pMsg,CTEXT);
   dwMsgBytes -= iHHdrLen;
   sprintf(pWork,"%4ld",dwMsgBytes);
   pCL += iCTextLen;
   strncpy(pCL,pWork,4);
   (*pECB->ServerSupportFunction)
      (pECB->ConnID,
      HSE_REQ_SEND_RESPONSE_HEADER,
      NULL,
      &dwMsgBytes,
      (LPDWORD)pMsg);
};   // SendResponse

//====================================================================
//
// Function name: ReadRegistry
//
//    Sets global operational parameters from registry if they exist.
//    Otherwise, compiled in defaults apply.
//
// Result:
//    FALSE    Registry entry found
//    TRUE     Registry entry does not exist
//
//====================================================================
BOOL ReadRegistry(VOID)
{
   HKEY hkTPCC;
   DWORD       dwMax;
   DWORD       dwRT;
   INT i;
   CHAR szValue[100];
   if (RegOpenKeyEx(HKEY_LOCAL_MACHINE,"SOFTWARE\\Unisys\\TPCC",0,
      KEY_READ, &hkTPCC) != ERROR_SUCCESS )
      return(TRUE);
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"EVENTLOG",0,&dwRT,szValue,&dwMax)
      == ERROR_SUCCESS)
   {
      if (abs(atoi(szValue) == 0))
         bEventLog = FALSE;
      else
         bEventLog = TRUE;
   };
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"CONSOLE",0,&dwRT,szValue,&dwMax)
      == ERROR_SUCCESS )
   {
      if (abs(atoi(szValue) == 0))
         bConsole = FALSE;
      else
         bConsole = TRUE;
   };
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"DIAGLEVEL",0,&dwRT,szValue,&dwMax)
      == ERROR_SUCCESS )
```

```
   {
      i = atoi(szValue);
      if (i < DIAG_FORCE)
         i = DIAG_FORCE;
      else
      if (i > DIAG_INFO)
         i = DIAG_INFO;
      uDiagLevel = i;
   };
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"MAXTERMS",0,&dwRT,szValue,&dwMax)
      == ERROR_SUCCESS )
   {
      iMaxTerms = abs(atoi(szValue));
   };
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"DELIVERYTHREADS",0,&dwRT,(BYTE *)
&szValue,&dwMax)
      == ERROR_SUCCESS )
   {
      lSetDThreads = abs(atoi(szValue));
   };
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"DQSIZE",0,&dwRT,(BYTE *) &szValue,&dwMax)
      == ERROR_SUCCESS )
   {
      lSetDQSize = abs(atoi(szValue));
   };
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"DQPATH",0,&dwRT,(BYTE *) &szValue,&dwMax)
      == ERROR_SUCCESS )
   {
      strcpy(szSetPath,szValue);
   };
   RegCloseKey(hkTPCC);
   return(FALSE);
};   // ReadRegistry
```

tpcchandler.h

```
// tpcchandler.h

#include "tpcc.h"

BOOL TPCCClear(TPCC_STATE * pTPCC);
UINT TPCCHandler(TPCC_STATE * pTPCC);
```

tpcchandler.c

```
// tpcchandler.c
//
// Copyright Unisys, 1999
//
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "tmon.h"
#include "diagio.h"
#include "tpcchandler.h"
#include "delivery.h"
#include "term.h"

// pTPCC->iFormId - TPCC forms enumeration.
#define FORM_NULL        0
#define FORM_LOGON       1
#define FORM_MENU        2
#define FORM_NEWORDER    3
#define FORM_PAYMENT     4
#define FORM_DELIVERY    5
#define FORM_ORDERSTATUS 6
#define FORM_STOCKLEVEL  7
#define FORM_EXIT        8
#define FORM_MAX         9

// CMD= HTML Command Enumeration and Name
#define CMD_NULL              0
#define CMD_PROCESS           1
#define CMD_NEWORDER_FORM     2
#define CMD_PAYMENT_FORM      3
#define CMD_DELIVERY_FORM     4
#define CMD_ORDERSTATUS_FORM  5
#define CMD_STOCKLEVEL_FORM   6
#define CMD_EXIT              7
#define CMD_SUBMIT            8
#define CMD_MENU_FORM         9
#define CMD_MAX               10

static CHAR * szCmds[] =
{
   "Unknown",
   "Process",
   "..NewOrder..",
   "..Payment..",
   "..Delivery..",
   "..Order-Status..",
   "..Stock-Level..",
   "..Exit..",
   "Submit",
   "Menu"
};

static CHAR * szFormLogin =
   HTTPHdr "<HTML>"
   "<HEAD><TITLE>Welcome To TPC-C</TITLE></HEAD><BODY>"
   "Please Identify your Warehouse and District for this session.<BR>"
   "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
   "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
   "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"1\">"
   "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"-2\">"
   "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"0\">"
   "Warehouse ID <INPUT NAME=\"w_id\" SIZE=4><BR>"
   "District ID <INPUT NAME=\"d_id\" SIZE=2><BR>"
   "<HR>"
   "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Submit\">"
   "</FORM>";

static CHAR * szMenuList =
```

```
   "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
   "<INPUT TYPE=\"submit\" NAME=\"CMD\"VALUE=\"..Payment..\">"
   "<INPUT TYPE=\"submit\" NAME=\"CMD\"VALUE=\"..Delivery..\">"
   "<INPUT TYPE=\"submit\" NAME=\"CMD\"VALUE=\"..Order-Status..\">"
   "<INPUT TYPE=\"submit\" NAME=\"CMD\"VALUE=\"..Stock-Level..\">"
   "<INPUT TYPE=\"submit\" NAME=\"CMD\"VALUE=\"..Exit..\">";

static CHAR * HTMLTrailer =
   "</BODY></HTML>";

static CHAR * TERMIDTOKEN = "TERMID=";
static CHAR * SYNCIDTOKEN = "SYNCID=";
static CHAR * FORMIDTOKEN = "FORMID=";
static CHAR * STATUSIDTOKEN = "STATUSID=";
static CHAR * CMDTOKEN = "CMD=";
static CHAR * NEWORDER_SERVICE = "NEWORDER";
static CHAR * PAYMENT_SERVICE = "PAYMENT";
static CHAR * ORDERSTATUS_SERVICE = "ORDERSTS";
static CHAR * STOCKLEVEL_SERVICE = "STOCKLVL";
static CHAR * ZIPPIC = "XXXXX-XXXX";

BOOL ProcessLogin(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
BOOL ProcessForm(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
BOOL ProcessNewOrder(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
BOOL ProcessPayment(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
BOOL ProcessDelivery(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
BOOL ProcessOrderStatus(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
BOOL ProcessStockLevel(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
VOID FormatLogin(CHAR * pMsg,CHAR * pAddText);
BOOL GetHidden(CHAR * pMsg,UINT * uFormId,INT * iSyncId,INT * iTermId);
BOOL GetCmd(CHAR * pMsg,CHAR * pWork,UINT uLen);
BOOL GetLongKey(LONG * lRslt,CHAR * pHTML,CHAR * pKey,TPCC_STATE * pTPCC);
BOOL GetIntKey(INT * iRslt,CHAR * pHTML,CHAR * pKey,TPCC_STATE * pTPCC);
BOOL GetShortKey(SHORT * sRslt,CHAR * pHTML,CHAR * pKey,TPCC_STATE *
pTPCC);
BOOL GetStringKey(CHAR * szRslt,CHAR * pHTML,CHAR * pKey,
                  TPCC_STATE * pTPCC,UINT uMax);
BOOL GetAmountKey(DOUBLE * dRslt,CHAR * pHTML,CHAR * pKey,
                  TPCC_STATE * pTPCC);
BOOL GetKeyValue(CHAR * pHTML,CHAR * pKey,CHAR * pValue,UINT uMax);
VOID FormatLogin(CHAR * pOut,CHAR * pAddText);
VOID FormatMenu(CHAR * pOut,TPCC_STATE * pTPCC);
VOID FormatNewOrder(CHAR * pOut,TPCC_STATE * pTPCC);
VOID FormatPayment(CHAR * pOut,TPCC_STATE * pTPCC);
VOID FormatDelivery(CHAR * pOut,TPCC_STATE * pTPCC);
VOID FormatOrderStatus(CHAR * pOut,TPCC_STATE * pTPCC);
VOID FormatStockLevel(CHAR * pOut,TPCC_STATE * pTPCC);
VOID FormatFormHdr(CHAR * pOut,CHAR * pTitle,TPCC_STATE * pTPCC);
VOID FormatRespHdr(CHAR * pOut,CHAR * pTitle,TPCC_STATE * pTPCC);
VOID FormatHTMLString(CHAR * pOut,CHAR * pIn,UINT uLen);
VOID FormatString(CHAR * pOut,CHAR * pPic,CHAR * pIn);
VOID UtilStrCpy(CHAR * pDest,CHAR * pSrc,INT n);
BOOL CheckNumeric(CHAR * pNum);

//============================================================================
//
// Function name: TPCCClear
//
//============================================================================
BOOL TPCCClear(TPCC_STATE * pTPCC)
{
   pTPCC->ConnID = 0;
   pTPCC->sWId = 0;
   pTPCC->sDId = 0;
   pTPCC->iSyncId = 0;
   pTPCC->iTermId = -2;
   pTPCC->uFormId = FORM_NULL;
   pTPCC->iStatusId = 0;
   pTPCC->tsTMon.lTMDataLen = 0;
   strcpy(pTPCC->ErrTxt,"");
   return(FALSE);
};   // TPCCClear

//============================================================================
//
// Function name: TPCCHandler
//
//============================================================================
UINT TPCCHandler(TPCC_STATE * pTPCC)
{
   INT iSyncId;
   INT iTermId;
   UINT uCmdId;
   UINT uRslt = TPCCSENDEND;  // default error handling
   TERM_STATE * pTerm;

   pTPCC->iStatusId = STATUS_OK;
   if (GetHidden(pTPCC->RecvMsg,&pTPCC->uFormId,&iSyncId,&iTermId))
   {
      uRslt = TPCCSEND;
      FormatLogin(pTPCC->SendMsg,pTPCC->ErrTxt);
      goto HdlrXit;
   };
   if (iTermId > 0)
   {
      pTerm = TermGet(iTermId);
      if (pTerm == NULL)
      {
         uRslt = TPCCSEND;
         strcpy(pTPCC->ErrTxt,"Invalid Term Id");
         FormatLogin(pTPCC->SendMsg,pTPCC->ErrTxt);
         goto HdlrXit;
      };
      if (pTerm->ConnID != pTPCC->ConnID)
      {
         uRslt = TPCCSEND;
         strcpy(pTPCC->ErrTxt,"TermId vs ConnId Mismatch");
         FormatLogin(pTPCC->SendMsg,pTPCC->ErrTxt);
         goto HdlrXit;
      };
      pTPCC->sWId = pTerm->sWId;
      pTPCC->sDId = pTerm->sDId;
      pTPCC->iSyncId = pTerm->iSyncId;
      pTPCC->iTermId = pTerm->iTermId;
   };
   uCmdId = GetCmd(pTPCC->RecvMsg,pTPCC->szWork,sizeof(pTPCC->szWork));
   // Except for Submit(log in), sWId must already be set
   if (pTPCC->sWId == 0 && uCmdId != CMD_SUBMIT)
   {
      strcpy(pTPCC->ErrTxt,"Must log in first!");
      FormatLogin(pTPCC->SendMsg,pTPCC->ErrTxt);
      uRslt = TPCCSEND;
```

```
   goto HdlrXit;
};
// Check for multiple log in attempts
if (pTPCC->sWId != 0 && uCmdId == CMD_SUBMIT)
{
   strcpy(pTPCC->ErrTxt,ERRTXT_ALREADY_LOGGEDIN);
   pTPCC->iStatusId = ERR_ALREADY_LOGGEDIN;
   FormatMenu(pTPCC->SendMsg,pTPCC);
   uRslt = TPCCSEND;
   goto HdlrXit;
};
// If not logging in, validate hidden fields
if (uCmdId != CMD_SUBMIT)
{
   if (iTermId != pTPCC->iTermId || iTermId != iSyncId)
   {
      sprintf(pTPCC->ErrTxt,"%s: Received %ld, %ld (%ld)",
         ERRTXT_TERMID,iTermId,iSyncId,pTPCC->iTermId);
      pTPCC->iStatusId = ERR_TERMID;
      FormatMenu(pTPCC->SendMsg,pTPCC);
      goto HdlrXit;
   };
};
// Process the command
switch (uCmdId)
{
   case CMD_SUBMIT:
      ProcessLogin(pTPCC->RecvMsg,pTPCC->SendMsg,pTPCC);
      break;
   case CMD_MENU_FORM:
      FormatMenu(pTPCC->SendMsg,pTPCC);
      break;
   case CMD_PROCESS:
      ProcessForm(pTPCC->RecvMsg,pTPCC->SendMsg,pTPCC);
      break;
   case CMD_NEWORDER_FORM:
      FormatNewOrder(pTPCC->SendMsg,pTPCC);
      break;
   case CMD_PAYMENT_FORM:
      FormatPayment(pTPCC->SendMsg,pTPCC);
      break;
   case CMD_DELIVERY_FORM:
      FormatDelivery(pTPCC->SendMsg,pTPCC);
      break;
   case CMD_ORDERSTATUS_FORM:
      FormatOrderStatus(pTPCC->SendMsg,pTPCC);
      break;
   case CMD_STOCKLEVEL_FORM:
      FormatStockLevel(pTPCC->SendMsg,pTPCC);
      break;
   case CMD_EXIT:
      TermFree(pTPCC->iTermId);
      strcpy(pTPCC->ErrTxt,"Logged Off");
      FormatLogin(pTPCC->SendMsg,pTPCC->ErrTxt);
      goto HdlrXit;
   default:
      strcpy(pTPCC->ErrTxt,ERRTXT_CMD_UNKNOWN);
      pTPCC->iStatusId = ERR_CMD_UNKNOWN;
      if (pTPCC->sWId == 0)
         FormatLogin(pTPCC->SendMsg,pTPCC->ErrTxt);
      else
         FormatMenu(pTPCC->SendMsg,pTPCC);
      break;
   };   // switch (uCmdId)

   uRslt = TPCCSEND;

HdlrXit:

   return(uRslt);

};   // TPCCHandler

//================================================================================
//
// Function name: ProcessLogin
//
//    ProcessLogin extracts WId and DId from the incoming form.  Assumes
//    log in has not previously completed (sWId == 0 already verified).
//
//    Result:
//       FALSE - log in successful, sWId and sDId set in pTPCC,
//               pOut contains menu.
//       TRUE - log in failed, pOut contains log in form with
//              error message.
//
//================================================================================
BOOL ProcessLogin(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   SHORT sWId;
   SHORT sDId;
   TERM_STATE * pTerm;

   if (GetShortKey(&sWId,pIn,"w_id",pTPCC))
   {
     FormatLogin(pOut,pTPCC->ErrTxt);
     return(TRUE);
   };
   if (sWId < 1)
   {
      sprintf(pTPCC->ErrTxt,"Warehouse Id (%d) Invalid",sWId);
      pTPCC->iStatusId = ERR_WID_INVALID;
      FormatLogin(pOut,pTPCC->ErrTxt);
      return(TRUE);
   };
   if (GetShortKey(&sDId,pIn,"d_id",pTPCC))
   {
     FormatLogin(pOut,pTPCC->ErrTxt);
     return(TRUE);
   };
   if (sDId < MIN_DId || sDId > MAX_DId)
   {
      sprintf(pTPCC->ErrTxt,"DId Out of Range(%ld,%ld) - %ld",
         MIN_DId,MAX_DId,sDId);
      pTPCC->iStatusId = ERR_DID_INVALID;
      FormatLogin(pOut,pTPCC->ErrTxt);
      return(TRUE);
   };
   pTerm = TermAlloc();
   if (pTerm == NULL)
   {
      sprintf(pTPCC->ErrTxt,"Unable to Allocate Terminal Entry");
```

```
      pTPCC->iStatusId = ERR_TERM_ALLOC;
      FormatLogin(pOut,pTPCC->ErrTxt);
      return(TRUE);
   };
   pTerm->ConnID = pTPCC->ConnID;
   pTerm->iSyncId = pTerm->iTermId;
   pTerm->sWId = abs(sWId);
   pTerm->sDId = abs(sDId);
   pTPCC->iTermId = pTerm->iTermId;
   pTPCC->iSyncId = pTerm->iSyncId;
   pTPCC->sWId = pTerm->sWId;
   pTPCC->sDId = pTerm->sDId;
   FormatMenu(pOut,pTPCC);
   return(FALSE);
};   // ProcessLogin

//================================================================
//
// Function name: ProcessForm
//
//    ProcessForm uses pTPCC->uFormId to determine which form input is
//    present and ready for processing.  Actual processing is done by
//    the form specific routine.
//
//    Result:
//       FALSE - form processed, pOut contains response.
//       TRUE - error processing form input, pOut contains reason.
//
//================================================================
BOOL ProcessForm(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   switch (pTPCC->uFormId )
   {
      case FORM_NEWORDER:
               return(ProcessNewOrder(pIn,pOut,pTPCC));
      case FORM_PAYMENT:
               return(ProcessPayment(pIn,pOut,pTPCC));
      case FORM_DELIVERY:
               return(ProcessDelivery(pIn,pOut,pTPCC));
      case FORM_ORDERSTATUS:
               return(ProcessOrderStatus(pIn,pOut,pTPCC));
      case FORM_STOCKLEVEL:
               return(ProcessStockLevel(pIn,pOut,pTPCC));
      default:
         sprintf(pTPCC->ErrTxt,"%s (%ld)",
            ERRTXT_FORM_UNKNOWN,pTPCC->uFormId);
         pTPCC->iStatusId = ERR_FORM_UNKNOWN;
         FormatMenu(pOut,pTPCC);
         break;
      }
   return(TRUE);
};   // ProcessForm

//================================================================
//
// Function name: ProcessNewOrder
//
//    ProcessNewOrder extracts the input data fields from pIn, processes
//    the data, and returns a response in pOut.
//
//    Result:
//       FALSE - NewOrder processed successfully.
//       TRUE - NewOrder processing failed.
//
//================================================================
BOOL ProcessNewOrder(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   NEW_ORDER_DATA * pnod;
   TMON_STATE * pTMon;
   CHAR szKey[20];
   CHAR szCredit[14];
   CHAR * ptr;
   UINT u;
   UINT uLine;
   BOOL bTMRslt;
   BOOL bTPRslt;
   INT iTPRslt;

   pTMon = &pTPCC->tsTMon;
   pTMon->lTMDataLen = sizeof(NEW_ORDER_DATA);
   memset(pTMon->pTMData,0,pTMon->lTMDataLen);
   pnod = (NEW_ORDER_DATA *) pTMon->pTMData;
   pnod->w_id = pTPCC->sWId;
   if (GetShortKey(&pnod->d_id,pIn,"DID*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (pnod->d_id < MIN_DId || pnod->d_id > MAX_DId)
   {
      wsprintf(pTPCC->ErrTxt,"DId Out of Range(%ld,%ld) - %ld",
         MIN_DId,MAX_DId,pnod->d_id);
      pTPCC->iStatusId = ERR_DID_INVALID;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   if (GetLongKey(&pnod->c_id,pIn,"CID*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   pnod->o_ol_cnt = 0;
   ptr = pIn;
   for(u=0, uLine=0; u < MAX_OL; u++)
   {
      wsprintf(szKey,"SP%2.2d*",u);
      ptr = strstr(ptr,szKey);
      if (GetShortKey(&pnod->Ol[uLine].ol_supply_w_id,ptr,szKey,pTPCC))
      {
         FormatMenu(pOut,pTPCC);
         return(TRUE);
      };
      wsprintf(szKey,"IID%2.2d*",u);
      if (GetLongKey(&pnod->Ol[uLine].ol_i_id,ptr,szKey,pTPCC))
      {
         FormatMenu(pOut,pTPCC);
         return(TRUE);
      };
      wsprintf(szKey,"Qty%2.2d*",u);
      if (GetShortKey(&pnod->Ol[uLine].ol_quantity,ptr,szKey,pTPCC))
      {
         FormatMenu(pOut,pTPCC);
```

```
      return(TRUE);
   };
   if (pnod->Ol[uLine].ol_i_id != 0)
   {
      if (pnod->Ol[uLine].ol_supply_w_id < 1)
      {
         wsprintf(pTPCC->ErrTxt,
            "Order Line %ld Contains Invalid WId %d",
            u + 1,pnod->Ol[uLine].ol_supply_w_id);
         pTPCC->iStatusId = ERR_WID_INVALID;
         FormatMenu(pOut,pTPCC);
         return(TRUE);
      };
      if (pnod->Ol[uLine].ol_quantity < MIN_QUANTITY ||
          pnod->Ol[uLine].ol_quantity > MAX_QUANTITY)
      {
         wsprintf(pTPCC->ErrTxt,
            "Order Line %ld Contains Invalid Qty %d",
            u + 1,pnod->Ol[uLine].ol_quantity);
         pTPCC->iStatusId = ERR_QUANTITY_INVALID;
         FormatMenu(pOut,pTPCC);
         return(TRUE);
      };
      uLine++;
   }   // if (ol_i_id !=0)
   else
   {
      if (pnod->Ol[uLine].ol_supply_w_id != 0)
      {
         wsprintf(pTPCC->ErrTxt,
            "Order Line %ld WId Supplied with No Item",u + 1);
         pTPCC->iStatusId = ERR_OL_INVALID;
         FormatMenu(pOut,pTPCC);
         return(TRUE);
      };
      if (pnod->Ol[uLine].ol_quantity != 0)
      {
         wsprintf(pTPCC->ErrTxt,
            "Order Line %ld Qty Supplied with No Item",u + 1);
         pTPCC->iStatusId = ERR_OL_INVALID;
         FormatMenu(pOut,pTPCC);
         return(TRUE);
      };
   };   // empty order line
};   // for (u < MAX_OL)
pnod->o_ol_cnt = uLine;
if (pnod->o_ol_cnt < MIN_OL)
{
   wsprintf(pTPCC->ErrTxt,"Too Few Order Lines %d",pnod->o_ol_cnt);
   pTPCC->iStatusId = ERR_OL_COUNT;
   FormatMenu(pOut,pTPCC);
   return(TRUE);
};
bTMRslt = TMTran(NEWORDER_SERVICE,pTMon,&bTPRslt,&iTPRslt);
pnod = (NEW_ORDER_DATA *) pTMon->pTMData;
if (bTMRslt)
{
   pTPCC->iStatusId = ERR_TM_INTERFACE;
   FormatMenu(pOut,pTPCC);
   return(TRUE);
};
```

```
   // Exclude invalid item id case
   if (bTPRslt && iTPRslt < SVC_NOERROR)
   {
      sprintf(pTPCC->ErrTxt,
         "New Order Service Returned Error(%ld): %s",
         iTPRslt,pnod->execution_status);
      pTPCC->iStatusId = ERR_SERVICE_RSLT;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   if (iTPRslt == SVC_BADITEMID)
      pTPCC->iStatusId = INVALID_IID;

   FormatRespHdr(pOut,"TPC-C New Order",pTPCC);
   sprintf(pOut + strlen(pOut),
      "<PRE>                                  New Order<BR>"
      "Warehouse: %4.4d   District: %2.2d                          ",
      pnod->w_id,pnod->d_id);
   if (!bTPRslt)
      {
      sprintf(pOut + strlen(pOut),
         "Date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d <BR>",
         pnod->o_entry_d.day,pnod->o_entry_d.month,
         pnod->o_entry_d.year,pnod->o_entry_d.hour,
         pnod->o_entry_d.minute,pnod->o_entry_d.second);
   }
   else
   {
      sprintf(pOut + strlen(pOut),"Date:<BR>");
   };
   FormatHTMLString(pTPCC->szWork,pnod->c_last,NAME_LEN);
   FormatHTMLString(szCredit,pnod->c_credit,2);
   sprintf(pOut + strlen(pOut),
      "Customer:  %4.4d   Name: %s   Credit: %s   ",
      pnod->c_id,pTPCC->szWork,szCredit);
   if (!bTPRslt)
   {
      sprintf(pOut + strlen(pOut),
         "%%Disc: %5.2f            <BR>",pnod->c_discount * 100);
      sprintf(pOut + strlen(pOut),
         "Order Number: %8.8d  Number of Lines: %2.2d        W_tax: %5.2f
D_tax: %5.2f  <BR><BR>",
         pnod->o_id,pnod->o_ol_cnt,pnod->w_tax * 100,pnod->d_tax * 100);
      strcat(pOut," Supp_W  Item_Id  Item Name                    Qty  Stock
B/G  Price    Amount<BR>");
      for (u = 0; u < (UINT) pnod->o_ol_cnt; u++)
      {
         FormatHTMLString(pTPCC->szWork,pnod->Ol[u].ol_i_name,24);
         sprintf(pOut + strlen(pOut),
            "  %4.4d   %6.6d   %s  %2.2d    %3.3d    %1.1s   $%6.2f
$%7.2f  <BR>",
            pnod->Ol[u].ol_supply_w_id,pnod->Ol[u].ol_i_id,
            pTPCC->szWork,pnod->Ol[u].ol_quantity,pnod->Ol[u].ol_stock,
            pnod->Ol[u].ol_brand_generic,pnod->Ol[u].ol_i_price,
            pnod->Ol[u].ol_amount );
      }
   }   // if (!bTPRslt)
   else
   {
      strcat(pOut,"%Disc:<BR>");
      sprintf(pOut + strlen(pOut),
```

```
         "Order Number: %8.8d  Number of Lines:              W_tax:
D_tax:<BR><BR>",
         pnod->o_id);
      strcat(pOut,
         " Supp_W  Item_Id  Item Name                  Qty  Stock  B/G
Price    Amount<BR>");
      u = 0;
   };
   for(; u < MAX_OL; u++)
      strcat(pOut,"<BR>");
   if (!bTPRslt)
   {
      sprintf(pOut + strlen(pOut),
         "Execution Status: %24.24s                  Total:  $%8.2f  ",
         pnod->execution_status,pnod->total_amount);
   }
   else
   {
      sprintf(pOut + strlen(pOut),
         "Execution Status: %24.24s                  Total:",
         pnod->execution_status);
   };
   sprintf(pOut + strlen(pOut),
      "</PRE><HR><BR>%s</FORM>%s",szMenuList,HTMLTrailer);

   return(FALSE);

};   // ProcessNewOrder

//============================================================================
//
// Function name: ProcessPayment
//
//    ProcessPayment extracts the input data fields from pIn, processes
//    the data, and returns a response in pOut.
//
//    Result:
//       FALSE - Payment processed successfully.
//       TRUE - Payment processing failed.
//
//============================================================================
BOOL ProcessPayment(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   PAYMENT_DATA * ppd;
   TMON_STATE * pTMon;
   BOOL bTMRslt;
   BOOL bTPRslt;
   INT iTPRslt;
   CHAR * pCredit;
   INT iCDLines;
   CHAR szWork2[60];
   CHAR szWork3[60];
   CHAR szWork4[60];
   CHAR szZip1[20];
   CHAR szZip2[20];
   INT i;

   pTMon = &pTPCC->tsTMon;
   pTMon->lTMDataLen = sizeof(PAYMENT_DATA);
   memset(pTMon->pTMData,0,pTMon->lTMDataLen);
   ppd = (PAYMENT_DATA *) pTMon->pTMData;
   ppd->w_id = pTPCC->sWId;
   // Get and validate DId
   if (GetShortKey(&ppd->d_id,pIn,"DID*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (ppd->d_id < MIN_DId || ppd->d_id > MAX_DId)
   {
      sprintf(pTPCC->ErrTxt,"DId Out of Range(%ld,%ld) - %ld",
         MIN_DId,MAX_DId,ppd->d_id);
      pTPCC->iStatusId = ERR_DID_INVALID;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   // Get and validate customer Id and name
   if (GetLongKey(&ppd->c_id,pIn,"CID*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (GetStringKey(ppd->c_last,pIn,"CLT*",pTPCC,NAME_LEN))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (ppd->c_id == 0 && ppd->c_last[0] == 0)
   {
      strcpy(pTPCC->ErrTxt,"Error - Customer Id and Name Empty");
      pTPCC->iStatusId = ERR_IDANDNAME_EMPTY;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   if (ppd->c_id != 0 && ppd->c_last[0] != 0)
   {
      strcpy(pTPCC->ErrTxt,
         "Error - Specify Customer Id or Name, not Both");
      pTPCC->iStatusId = ERR_IDANDNAME_ENTERED;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   // Get and validate customer DId
   if (GetShortKey(&ppd->c_d_id,pIn,"CDI*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (ppd->c_d_id < MIN_DId || ppd->c_d_id > MAX_DId)
   {
      sprintf(pTPCC->ErrTxt,"Cust DId Out of Range(%ld,%ld) - %ld",
         MIN_DId,MAX_DId,ppd->d_id);
      pTPCC->iStatusId = ERR_DID_INVALID;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   // Get and validate customer WId
   if (GetShortKey(&ppd->c_w_id,pIn,"CWI*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
```

```
if (ppd->c_w_id < 1)
{
   sprintf(pTPCC->ErrTxt,
      "Payment Contains Invalid Customer WId %d",
      ppd->c_w_id);
   pTPCC->iStatusId = ERR_WID_INVALID;
   FormatMenu(pOut,pTPCC);
   return(TRUE);
};
// Get and validate amount
if (GetAmountKey(&ppd->h_amount,pIn,"HAM*",pTPCC))
{
  FormatMenu(pOut,pTPCC);
  return(TRUE);
};
if (ppd->h_amount <= 0)
{
   sprintf(pTPCC->ErrTxt,
      "Payment Amount Negative or Missing");
   pTPCC->iStatusId = ERR_AMOUNT_INVALID;
   FormatMenu(pOut,pTPCC);
   return(TRUE);
};
bTMRslt = TMTran(PAYMENT_SERVICE,pTMon,&bTPRslt,&iTPRslt);
ppd = (PAYMENT_DATA *) pTMon->pTMData;
if (bTMRslt)
{
   pTPCC->iStatusId = ERR_TM_INTERFACE;
   FormatMenu(pOut,pTPCC);
   return(TRUE);
};
if (bTPRslt)
{
   sprintf(pTPCC->ErrTxt,
      "Payment Service Returned Error(%ld): %s",
      iTPRslt,ppd->execution_status);
   pTPCC->iStatusId = ERR_SERVICE_RSLT;
   FormatMenu(pOut,pTPCC);
   return(TRUE);
};
FormatRespHdr(pOut,"TPC-C Payment",pTPCC);
sprintf(pOut + strlen(pOut),
       "<PRE>                                   Payment<BR>"
   "Date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d <BR><BR>"
   "Warehouse: %4.4d"
   "                        District: %2.2d<BR>",
        ppd->h_date.day,ppd->h_date.month,
   ppd->h_date.year,ppd->h_date.hour,
   ppd->h_date.minute,ppd->h_date.second,
   ppd->w_id,ppd->d_id);

FormatHTMLString(szWork2,ppd->w_street_1,ADDR_LEN);
FormatHTMLString(szWork3,ppd->d_street_1,ADDR_LEN);
sprintf(pOut + strlen(pOut),
   "%s                      %s<BR>",szWork2,szWork3);
FormatHTMLString(szWork2,ppd->w_street_2,ADDR_LEN);
FormatHTMLString(szWork3,ppd->d_street_2,ADDR_LEN);
sprintf(pOut + strlen(pOut),
   "%s                      %s<BR>",szWork2,szWork3);
FormatHTMLString(pTPCC->szWork,ppd->w_city,ADDR_LEN);
FormatHTMLString(szWork2,ppd->d_city,ADDR_LEN);
FormatHTMLString(szWork3,ppd->w_state,STATE_LEN);
FormatHTMLString(szWork4,ppd->d_state,STATE_LEN);
FormatString(szZip1,ZIPPIC,ppd->w_zip);
FormatString(szZip2,ZIPPIC,ppd->d_zip);
sprintf(pOut +  strlen(pOut),
   "%s %s %10.10s       %s %s %10.10s<BR><BR>",
   pTPCC->szWork,szWork3,szZip1,szWork2,szWork4,szZip2);
FormatHTMLString(szWork2,ppd->c_first,NAME_LEN);
FormatHTMLString(szWork3,ppd->c_middle,2);
FormatHTMLString(szWork4,ppd->c_last,NAME_LEN);
sprintf(pOut + strlen(pOut),
   "Customer: %4.4d  Cust-Warehouse: %4.4d  Cust-District: %2.2d<BR>"
   "Name:   %s %s %s     Since:  %2.2d-%2.2d-%4.4d<BR>",
   ppd->c_id,ppd->c_w_id,ppd->c_d_id,
   szWork2,szWork3,szWork4,
   ppd->c_since.day,ppd->c_since.month,ppd->c_since.year);
FormatHTMLString(pTPCC->szWork,ppd->c_street_1,ADDR_LEN);
FormatHTMLString(szWork2,ppd->c_credit,2);
FormatHTMLString(szWork3,ppd->c_street_2,ADDR_LEN);
sprintf(pOut + strlen(pOut),
   "        %s                      Credit: %s<BR>"
   "        %s                      %%Disc:  %5.2f<BR>",
   pTPCC->szWork,szWork2,szWork3,ppd->c_discount * 100);
FormatHTMLString(szWork2,ppd->c_city,ADDR_LEN);
FormatHTMLString(szWork3,ppd->c_state,STATE_LEN);
FormatString(szZip1,ZIPPIC,ppd->c_zip);
FormatString(szWork4,"XXXXXX-XXX-XXX-XXXX",ppd->c_phone);
sprintf(pOut + strlen(pOut),
   "        %s %s %10.10s       Phone:  %-19.19s<BR><BR>"
   "Amount Paid:          $%7.2f      New Cust Balance: $%14.2f<BR>"
   "Credit Limit:   $%13.2f<BR><BR>",
   szWork2,szWork3,szZip1,szWork4,
   ppd->h_amount,ppd->c_balance,ppd->c_credit_lim);
pCredit = ppd->c_credit;
if (*pCredit == 'B' && *(pCredit + 1) == 'C')
{
   pCredit = ppd->c_data;
   iCDLines = strlen(pCredit) / 50;
   for(i = 0; i < 4; i++, pCredit += 50)
   {
      if (i <= iCDLines)
         UtilStrCpy(szWork2,pCredit,50);
      else
         szWork2[0] = 0;
      FormatHTMLString(szWork3,szWork2,50);
      if (!i)
         sprintf(pOut + strlen(pOut),
            "Cust-Data: %s<BR>",szWork3);
      else
         sprintf(pOut + strlen(pOut),
            "           %s<BR>",szWork3);
   };
}
else
   strcat(pOut,"Cust-Data: <BR><BR><BR><BR>");
sprintf(pOut + strlen(pOut),
   "</PRE><HR><BR>%s</FORM>%s",szMenuList,HTMLTrailer);

return(FALSE);

};   // ProcessPayment
```

```
//=========================================================================
//
// Function name: ProcessDelivery
//
//    ProcessDelivery extracts the input data fields from pIn, processes
//    the data, and returns a response in pOut.
//
//    Result:
//       FALSE - Delivery processed successfully.
//       TRUE - Delivery processing failed.
//
//=========================================================================
BOOL ProcessDelivery(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   DELIVERY_DATA * pdd;
   TMON_STATE * pTMon;
   BOOL bTMRslt;

   pTMon = &pTPCC->tsTMon;
   pTMon->lTMDataLen = sizeof(DELIVERY_DATA);
   memset(pTMon->pTMData,0,pTMon->lTMDataLen);
   pdd = (DELIVERY_DATA *) pTMon->pTMData;
   pdd->w_id = pTPCC->sWId;
   // Get and validate carrier id
   if (GetShortKey(&pdd->o_carrier_id,pIn,"OCD*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (pdd->o_carrier_id < MIN_CARRIER ||
       pdd->o_carrier_id > MAX_CARRIER)
   {
      sprintf(pTPCC->ErrTxt,"Carrier Id Out of Range(%ld,%ld) - %ld",
         MIN_CARRIER,MAX_CARRIER,pdd->o_carrier_id);
      pTPCC->iStatusId = ERR_CARRIER_INVALID;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   GetLocalTime(&pdd->QTime);
   bTMRslt = DeliveryPost(pdd);
   if (bTMRslt)
   {
      wsprintf(pTPCC->ErrTxt,
         "Delivery Post Returned Error: Queue Request Failed");
      pTPCC->iStatusId = ERR_DQPOST_RSLT;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   strcpy(pdd->execution_status,"Delivery has been queued.");
   FormatRespHdr(pOut,"TPC-C Delivery",pTPCC);
   sprintf(pOut + strlen(pOut),
      "<PRE>                                  Delivery<BR>"
      "Warehouse: %4.4d<BR><BR>"
      "Carrier Number: %2.2d<BR><BR>"
      "Execution Status: %25.25s<BR>",
      pdd->w_id,pdd->o_carrier_id,pdd->execution_status);
   sprintf(pOut + strlen(pOut),
      "</PRE><HR><BR>%s</FORM>%s",szMenuList,HTMLTrailer);

   return(FALSE);
};   // ProcessDelivery

//=========================================================================
//
// Function name: ProcessOrderStatus
//
//    ProcessOrderStatus extracts the input data fields from pIn,
//    processes the data, and returns a response in pOut.
//
//    Result:
//       FALSE - OrderStatus processed successfully.
//       TRUE - OrderStatus processing failed.
//
//=========================================================================
BOOL ProcessOrderStatus(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   ORDER_STATUS_DATA * posd;
   TMON_STATE * pTMon;
   INT i;
   CHAR szWork2[50];
   CHAR szWork3[50];
   BOOL bTMRslt;
   BOOL bTPRslt;
   INT iTPRslt;

   pTMon = &pTPCC->tsTMon;
   pTMon->lTMDataLen = sizeof(ORDER_STATUS_DATA);
   memset(pTMon->pTMData,0,pTMon->lTMDataLen);
   posd = (ORDER_STATUS_DATA *) pTMon->pTMData;
   posd->w_id = pTPCC->sWId;
   if (GetShortKey(&posd->d_id,pIn,"DID*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (posd->d_id < MIN_DId || posd->d_id > MAX_DId)
   {
      sprintf(pTPCC->ErrTxt,"DId Out of Range(%ld,%ld) - %ld",
         MIN_DId,MAX_DId,posd->d_id);
      pTPCC->iStatusId = ERR_DID_INVALID;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   if (GetLongKey(&posd->c_id,pIn,"CID*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (GetStringKey(posd->c_last,pIn,"CLT*",pTPCC,NAME_LEN))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (posd->c_id == 0 && posd->c_last[0] == 0)
   {
      strcpy(pTPCC->ErrTxt,"Error - Customer Id and Name Empty");
      pTPCC->iStatusId = ERR_IDANDNAME_EMPTY;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
```

```
   if (posd->c_id != 0 && posd->c_last[0] != 0)
   {
      strcpy(pTPCC->ErrTxt,
         "Error - Specify Customer Id or Name, not Both");
      pTPCC->iStatusId = ERR_IDANDNAME_ENTERED;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   bTMRslt = TMTran(ORDERSTATUS_SERVICE,pTMon,&bTPRslt,&iTPRslt);
   posd = (ORDER_STATUS_DATA *) pTMon->pTMData;
   if (bTMRslt)
   {
      pTPCC->iStatusId = ERR_TM_INTERFACE;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   if (bTPRslt)
   {
      sprintf(pTPCC->ErrTxt,
         "Order Status Service Returned Error(%ld): %s",
         iTPRslt,posd->execution_status);
      pTPCC->iStatusId = ERR_SERVICE_RSLT;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   FormatRespHdr(pOut,"TPC-C Order-Status",pTPCC);
   sprintf(pOut + strlen(pOut),
      "<PRE>                                Order-Status<BR>"
      "Warehouse: %4.4d   District: %2.2d<BR>",
      posd->w_id,posd->d_id);
   FormatHTMLString(pTPCC->szWork,posd->c_first,NAME_LEN);
   FormatHTMLString(szWork2,posd->c_middle,2);
   FormatHTMLString(szWork3,posd->c_last,NAME_LEN);
   sprintf(pOut + strlen(pOut),
      "Customer: %4.4d   Name: %s %s %s<BR>"
      "Cust-Balance: $%9.2f<BR><BR>",
      posd->c_id,pTPCC->szWork,szWork2,szWork3,posd->c_balance);
   sprintf(pOut + strlen(pOut),
      "Order-Number: %8.8d   Entry-Date: %2.2d-%2.2d-%4.4d
%2.2d:%2.2d:%2.2d   Carrier-Number: %2.2d<BR>"
      "Supply-W     Item-Id    Qty     Amount      Delivery-Date<BR>",
      posd->o_id,posd->o_entry_d.day,posd->o_entry_d.month,
      posd->o_entry_d.year,posd->o_entry_d.hour,
      posd->o_entry_d.minute,posd->o_entry_d.second,
      posd->o_carrier_id);
   for(i = 0; i < posd->o_ol_cnt; i++)
   {
      sprintf(pOut + strlen(pOut),
         "  %4.4d       %6.6d     %2.2d     $%8.2f      %2.2d-%2.2d-
%4.4d<BR>",
         posd->OlOrderStatusData[i].ol_supply_w_id,
         posd->OlOrderStatusData[i].ol_i_id,
         posd->OlOrderStatusData[i].ol_quantity,
         posd->OlOrderStatusData[i].ol_amount,
         posd->OlOrderStatusData[i].ol_delivery_d.day,
         posd->OlOrderStatusData[i].ol_delivery_d.month,
         posd->OlOrderStatusData[i].ol_delivery_d.year);
   };
   sprintf(pOut + strlen(pOut),
      "<BR></PRE><HR><BR>%s</FORM>%s",szMenuList,HTMLTrailer);
   return(FALSE);

};   // ProcessOrderStatus

//================================================================================
//
// Function name: ProcessStockLevel
//
//    ProcessStockLevel extracts the input data fields from pIn,
//    processes the data, and returns a response in pOut.
//
//    Result:
//       FALSE - StockLevel processed successfully.
//       TRUE - StockLevel processing failed.
//
//================================================================================
BOOL ProcessStockLevel(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   STOCK_LEVEL_DATA * psld;
   TMON_STATE * pTMon;
   BOOL bTMRslt;
   BOOL bTPRslt;
   INT iTPRslt;

   pTMon = &pTPCC->tsTMon;
   pTMon->lTMDataLen = sizeof(STOCK_LEVEL_DATA);
   memset(pTMon->pTMData,0,pTMon->lTMDataLen);
   psld = (STOCK_LEVEL_DATA *) pTMon->pTMData;
   psld->w_id = pTPCC->sWId;
   psld->d_id = pTPCC->sDId;
   psld->low_stock = 0;
   psld->execution_status[0] = 0;
   if (GetShortKey(&psld->thresh_hold,pIn,"TT*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (psld->thresh_hold < MIN_THRESHOLD ||
       psld->thresh_hold > MAX_THRESHOLD)
   {
      sprintf(pTPCC->ErrTxt,"Threshold Out of Range(%ld,%ld) - %ld",
         MIN_THRESHOLD,MAX_THRESHOLD,psld->thresh_hold);
      pTPCC->iStatusId = ERR_THRESHOLD_RANGE;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };

   bTMRslt = TMTran(STOCKLEVEL_SERVICE,pTMon,&bTPRslt,&iTPRslt);
   psld = (STOCK_LEVEL_DATA *) pTMon->pTMData;
   if (bTMRslt)
   {
      pTPCC->iStatusId = ERR_TM_INTERFACE;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   if (bTPRslt)
   {
      sprintf(pTPCC->ErrTxt,
         "Stock Level Service Returned Error(%ld): %s",
         iTPRslt,psld->execution_status);
      pTPCC->iStatusId = ERR_SERVICE_RSLT;
```

```
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };

   FormatRespHdr(pOut,"TPC-C Stock Level",pTPCC);
   sprintf(pOut + strlen(pOut),
      "<PRE>                                Stock-Level<BR>"
      "Warehouse: %4.4d   District: %2.2d<BR><BR>"
      "Stock Level Threshold: %2.2d<BR><BR>"
      "low stock: %3.3ld</PRE><BR><HR>"
      "%s</FORM>%s",
      pTPCC->sWId,pTPCC->sDId,psld->thresh_hold,psld->low_stock,
      szMenuList,HTMLTrailer);

   return(FALSE);

};   // ProcessStockLevel

//================================================================================
//
// Function name: GetHidden
//
//================================================================================
BOOL GetHidden(CHAR * pMsg,UINT * uFormId,INT * iSyncId,INT * iTermId)
{
   CHAR * pPtr;
   BOOL bRslt = TRUE;

   // Extract TERMID
   pPtr = strstr(pMsg,TERMIDTOKEN);
   if (pPtr == NULL)
      goto xit;
   pPtr += strlen(TERMIDTOKEN);
   *iTermId = atoi(pPtr);

   // Extract SYNCID
   pPtr = strstr(pMsg,SYNCIDTOKEN);
   if (pPtr == NULL)
      goto xit;
   pPtr += strlen(SYNCIDTOKEN);
   *iSyncId = atoi(pPtr);

   // Extract FORMID
   pPtr = strstr(pMsg,FORMIDTOKEN);
   if (pPtr == NULL)
      goto xit;
   pPtr += strlen(FORMIDTOKEN);
   *uFormId = abs(atoi(pPtr));

   bRslt = FALSE;

xit:

   return(bRslt);

};   // GetHidden

//================================================================================
//
// Function name: GetCmd
//
//================================================================================
BOOL GetCmd(CHAR * pMsg,CHAR * pWork,UINT uLen)
{
   UINT u;
   CHAR * ptr;
   CHAR * pUpd;

   // Check for CMD key
   if (!(ptr = strstr(pMsg,CMDTOKEN)))
      return(CMD_NULL);
   ptr += sizeof(CMDTOKEN);
   pUpd = pWork;
   while (*ptr && *ptr != '&')
      *pUpd++ = *ptr++;
   *pUpd = 0;

   // Convert command name into command index
   for(u=0; u < CMD_MAX; u++)
   {
      if (!strcmp(szCmds[u],pWork))
         return(u);
   };

   // Command string not found
   return(CMD_NULL);

};   // GetCmd

//================================================================================
//
// Function name: GetLongKey
//
//================================================================================
BOOL GetLongKey(LONG * lRslt,CHAR * pHTML,CHAR * pKey,TPCC_STATE * pTPCC)
{
   if (GetKeyValue(pHTML,pKey,pTPCC->szWork,sizeof(pTPCC->szWork)))
   {
      sprintf(pTPCC->ErrTxt,"Error - Missing %s Key",pKey);
      pTPCC->iStatusId = ERR_MISSING_KEY;
      return(TRUE);
   };
   if (pTPCC->szWork[0] != 0 )
   {
      if (CheckNumeric(pTPCC->szWork))
      {
         sprintf(pTPCC->ErrTxt,"Error - %s Value Not Numeric",pKey);
         pTPCC->iStatusId = ERR_NOT_NUMERIC;
         return(TRUE);
      };
   };
   *lRslt = atol(pTPCC->szWork);
   return(FALSE);
};   // GetLongKey

//================================================================================
//
// Function name: GetIntKey
//
//================================================================================
BOOL GetIntKey(INT * iRslt,CHAR * pHTML,CHAR * pKey,TPCC_STATE * pTPCC)
{
```

```
   if (GetKeyValue(pHTML,pKey,pTPCC->szWork,sizeof(pTPCC->szWork)))
   {
      sprintf(pTPCC->ErrTxt,"Error - Missing %s Key",pKey);
      pTPCC->iStatusId = ERR_MISSING_KEY;
      return(TRUE);
   };
   if (pTPCC->szWork[0] != 0 )
   {
      if (CheckNumeric(pTPCC->szWork))
      {
         sprintf(pTPCC->ErrTxt,"Error - %s Value Not Numeric",pKey);
         pTPCC->iStatusId = ERR_NOT_NUMERIC;
         return(TRUE);
      };
   };
   *iRslt = atoi(pTPCC->szWork);
   return(FALSE);
};   // GetIntKey

//========================================================================
//
// Function name: GetShortKey
//
//========================================================================
BOOL GetShortKey(SHORT * sRslt,CHAR * pHTML,CHAR * pKey,TPCC_STATE *
pTPCC)
{
   if (GetKeyValue(pHTML,pKey,pTPCC->szWork,sizeof(pTPCC->szWork)))
   {
      sprintf(pTPCC->ErrTxt,"Error - Missing %s Key",pKey);
      pTPCC->iStatusId = ERR_MISSING_KEY;
      return(TRUE);
   };
   if (pTPCC->szWork[0] != 0 )
   {
      if (CheckNumeric(pTPCC->szWork))
      {
         sprintf(pTPCC->ErrTxt,"Error - %s Value Not Numeric",pKey);
         pTPCC->iStatusId = ERR_NOT_NUMERIC;
         return(TRUE);
      };
   };
   *sRslt = (SHORT) atoi(pTPCC->szWork);
   return(FALSE);
};   // GetShortKey

//========================================================================
//
// Function name: GetStringKey
//
//========================================================================
BOOL GetStringKey(CHAR * szRslt,CHAR * pHTML,CHAR * pKey,
                  TPCC_STATE * pTPCC,UINT uMax)
{
   UINT uLen;
   if (GetKeyValue(pHTML,pKey,pTPCC->szWork,sizeof(pTPCC->szWork)))
   {
      sprintf(pTPCC->ErrTxt,"Error - Missing %s Key",pKey);
      pTPCC->iStatusId = ERR_MISSING_KEY;
      return(TRUE);
   };
   uLen = strlen(pTPCC->szWork);
   if (uLen > uMax)
   {
      sprintf(pTPCC->ErrTxt,
         "Error - %s Key Input (%ld) Too Long (%ld)"
         ,pKey,uLen,uMax);
      pTPCC->iStatusId = ERR_INPUT_TOOLONG;
      return(TRUE);
   };
   _strupr(pTPCC->szWork);
   strcpy(szRslt,pTPCC->szWork);
   return(FALSE);
};   // GetStringKey

//========================================================================
//
// Function name: GetAmountKey
//
//========================================================================
BOOL GetAmountKey(DOUBLE * dRslt,CHAR * pHTML,CHAR * pKey,
                  TPCC_STATE * pTPCC)
{
   CHAR * ptr;
   BOOL bInvalid = FALSE;

   if (GetKeyValue(pHTML,pKey,pTPCC->szWork,sizeof(pTPCC->szWork)))
   {
      sprintf(pTPCC->ErrTxt,"Error - Missing %s Key",pKey);
      pTPCC->iStatusId = ERR_MISSING_KEY;
      return(TRUE);
   };
   ptr = pTPCC->szWork;
   while(*ptr)
   {
      if (*ptr == '.')
      {
         ptr++;
         if (!*ptr)
            break;
         if (*ptr < '0' || *ptr > '9')
         {
            bInvalid = TRUE;
            break;
         };
         ptr++;
         if (!*ptr)
            break;
         if (*ptr < '0' || *ptr > '9')
         {
            bInvalid = TRUE;
            break;
         };
         ptr++;
         if (*ptr)
         {
            bInvalid = TRUE;
            break;
         };
         break;
      }
      else
```

```
      if (*ptr < '0' || *ptr > '9')
      {
         bInvalid = TRUE;
         break;
      };
               ptr++;
       };   // while(!*ptr)

   if (!bInvalid)
      *dRslt = atof(pTPCC->szWork);
   else
   {
      sprintf(pTPCC->ErrTxt,
         "Error - Invalid Amount Format (%s)",pTPCC->szWork);
      pTPCC->iStatusId = ERR_AMOUNT_BADFORM;
   };

   return(bInvalid);

};   // GetAmountKey

//=========================================================================
//
// Function name: GetKeyValue
//   This function parses an HTTP formatted string for specific key
//   values.  HTTP keys terminate with '='.  HTTP values terminate
//   with an '&' or '\0'.
//
//    Result:
//       FALSE - Key found, string value return in pValue
//       TRUE - Key not found
//
//=========================================================================
BOOL GetKeyValue(CHAR * pHTML,CHAR * pKey,CHAR * pValue,UINT uMax)
{
   CHAR * ptr;
   if (!(ptr=strstr(pHTML,pKey)))
      return(TRUE);
   if (!(ptr=strchr(ptr,'=')))
      return(TRUE);
   ptr++;
   uMax--;
   while (*ptr && *ptr != '&' && uMax)
   {
      *pValue++ = *ptr++;
      uMax--;
   };
   *pValue = 0;
   return(FALSE);
};   // GetKeyValue

//=========================================================================
//
// Function name: FormatLogin
//
//=========================================================================
VOID FormatLogin(CHAR * pOut,CHAR * pAddText)
{
   sprintf(pOut,"%s<BR>%s<BR>%s",szFormLogin,pAddText,HTMLTrailer);
};   // FormatLogin

//=========================================================================
//
// Function name: FormatMenu
//
//=========================================================================
VOID FormatMenu(CHAR * pOut,TPCC_STATE * pTPCC)
{
   sprintf(pOut,
      "%s<HTML><HEAD><TITLE>TPC-C MainMenu</TITLE></HEAD><BODY>"
      "Select Desired Transaction.<BR><HR>"
      "<FORM ACTION=\"tpcc.dll\"METHOD=\"GET\">"
      "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
      "%s</FORM><BR>%s<BR>%s",
      HTTPHdr,pTPCC->iStatusId,pTPCC->iTermId,pTPCC->iSyncId,FORM_MENU,
      szMenuList,pTPCC->ErrTxt,HTMLTrailer);
};   // FormatMenu

//=========================================================================
//
// Function name: FormatNewOrder
//
//=========================================================================
VOID FormatNewOrder(CHAR * pOut,TPCC_STATE * pTPCC)
{
   pTPCC->uFormId = FORM_NEWORDER;
   FormatFormHdr(pOut,"TPC-C New Order",pTPCC);
   sprintf(pOut + strlen(pOut),
      "<PRE>                                   New Order<BR>"
      "Warehouse: %4.4d   District: <INPUT NAME=\"DID*\" SIZE=1>
Date:<BR>"
               "Customer:  <INPUT NAME=\"CID*\" SIZE=4>   Name:
Credit:      %Disc:<BR>"
      "Order Number:           Number of Lines:           W_tax:
D_tax:<BR><BR>"
      " Supp_W  Item_Id  Item Name                 Qty  Stock  B/G  Price
Amount<BR>"
      " <INPUT NAME=\"SP00*\" SIZE=4>  <INPUT NAME=\"IID00*\" SIZE=6>
<INPUT NAME=\"Qty00*\" SIZE=1><BR>"
      " <INPUT NAME=\"SP01*\" SIZE=4>  <INPUT NAME=\"IID01*\" SIZE=6>
<INPUT NAME=\"Qty01*\" SIZE=1><BR>"
      " <INPUT NAME=\"SP02*\" SIZE=4>  <INPUT NAME=\"IID02*\" SIZE=6>
<INPUT NAME=\"Qty02*\" SIZE=1><BR>"
      " <INPUT NAME=\"SP03*\" SIZE=4>  <INPUT NAME=\"IID03*\" SIZE=6>
<INPUT NAME=\"Qty03*\" SIZE=1><BR>"
      " <INPUT NAME=\"SP04*\" SIZE=4>  <INPUT NAME=\"IID04*\" SIZE=6>
<INPUT NAME=\"Qty04*\" SIZE=1><BR>"
      " <INPUT NAME=\"SP05*\" SIZE=4>  <INPUT NAME=\"IID05*\" SIZE=6>
<INPUT NAME=\"Qty05*\" SIZE=1><BR>"
      " <INPUT NAME=\"SP06*\" SIZE=4>  <INPUT NAME=\"IID06*\" SIZE=6>
<INPUT NAME=\"Qty06*\" SIZE=1><BR>"
      " <INPUT NAME=\"SP07*\" SIZE=4>  <INPUT NAME=\"IID07*\" SIZE=6>
<INPUT NAME=\"Qty07*\" SIZE=1><BR>"
      " <INPUT NAME=\"SP08*\" SIZE=4>  <INPUT NAME=\"IID08*\" SIZE=6>
<INPUT NAME=\"Qty08*\" SIZE=1><BR>"
      " <INPUT NAME=\"SP09*\" SIZE=4>  <INPUT NAME=\"IID09*\" SIZE=6>
<INPUT NAME=\"Qty09*\" SIZE=1><BR>"
      " <INPUT NAME=\"SP10*\" SIZE=4>  <INPUT NAME=\"IID10*\" SIZE=6>
<INPUT NAME=\"Qty10*\" SIZE=1><BR>"
```

```
      " <INPUT NAME=\"SP11*\" SIZE=4>  <INPUT NAME=\"IID11*\" SIZE=6>
<INPUT NAME=\"Qty11*\" SIZE=1><BR>"
      " <INPUT NAME=\"SP12*\" SIZE=4>  <INPUT NAME=\"IID12*\" SIZE=6>
<INPUT NAME=\"Qty12*\" SIZE=1><BR>"
      " <INPUT NAME=\"SP13*\" SIZE=4>  <INPUT NAME=\"IID13*\" SIZE=6>
<INPUT NAME=\"Qty13*\" SIZE=1><BR>"
      " <INPUT NAME=\"SP14*\" SIZE=4>  <INPUT NAME=\"IID14*\" SIZE=6>
<INPUT NAME=\"Qty14*\" SIZE=1><BR>"
      "Execution Status:
Total:<BR><HR>"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Process\">"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Menu\">"
      "</FORM>%s",
      pTPCC->sWId,HTMLTrailer);
};   // FormatNewOrder

//================================================================================
//
// Function name: FormatPayment
//
//================================================================================
VOID FormatPayment(CHAR * pOut,TPCC_STATE * pTPCC)
{
   pTPCC->uFormId = FORM_PAYMENT;
   FormatFormHdr(pOut,"TPC-C Payment",pTPCC);
   sprintf(pOut + strlen(pOut),
      "<PRE>                                    Payment<BR>"
      "Date:<BR><BR>"
      "Warehouse: %4.4d"
      "                        District: <INPUT NAME=\"DID*\"
SIZE=1><BR><BR><BR><BR><BR>"
      "Customer: <INPUT NAME=\"CID*\" SIZE=4>"
      "Cust-Warehouse: <INPUT NAME=\"CWI*\" SIZE=4>  "
      "Cust-District: <INPUT NAME=\"CDI*\" SIZE=1><BR>"
      "Name:                       <INPUT NAME=\"CLT*\" SIZE=16>
Since:<BR>"
      "                                                 Credit:<BR>"
      "                                                 Disc:<BR>"
      "                                                 Phone:<BR><BR>"
      "Amount Paid:          $<INPUT NAME=\"HAM*\" SIZE=7>      New Cust
Balance:<BR>"
      "Credit Limit:<BR><BR>Cust-Data: <BR><BR><BR><BR></PRE><HR>"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Process\">"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Menu\">"
      "</FORM>%s",
      pTPCC->sWId,HTMLTrailer);
};   // FormatPayment

//================================================================================
//
// Function name: FormatDelivery
//
//================================================================================
VOID FormatDelivery(CHAR * pOut,TPCC_STATE * pTPCC)
{
   pTPCC->uFormId = FORM_DELIVERY;
   FormatFormHdr(pOut,"TPC-C Delivery",pTPCC);
   sprintf(pOut + strlen(pOut),
      "<PRE>                                    Delivery<BR>"
      "Warehouse: %4.4d<BR><BR>"
      "Carrier Number: <INPUT NAME=\"OCD*\" SIZE=1><BR><BR>"
      "Execution Status:<BR></PRE><HR>"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Process\">"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Menu\">"
      "</FORM>%s",
      pTPCC->sWId,HTMLTrailer);
};   // FormatDelivery

//================================================================================
//
// Function name: FormatOrderStatus
//
//================================================================================
VOID FormatOrderStatus(CHAR * pOut,TPCC_STATE * pTPCC)
{
   pTPCC->uFormId = FORM_ORDERSTATUS;
   FormatFormHdr(pOut,"TPC-C Order-Status",pTPCC);
   sprintf(pOut + strlen(pOut),
      "<PRE>                                 Order-Status<BR>"
      "Warehouse: %4.4d   "
      "District: <INPUT NAME=\"DID*\" SIZE=1><BR>"
               "Customer: <INPUT NAME=\"CID*\" SIZE=4>   Name:
<INPUT NAME=\"CLT*\" SIZE=23><BR>"
      "Cust-Balance:<BR><BR>"
      "Order-Number:            Entry-Date:                        Carrier-
Number:<BR>"
      "Supply-W     Item-Id    Qty     Amount      Delivery-
Date<BR></PRE><HR>"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Process\">"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Menu\">"
      "</FORM>%s",
      pTPCC->sWId,HTMLTrailer);
};   // FormatOrderStatus

//================================================================================
//
// Function name: FormatStockLevel
//
//================================================================================
VOID FormatStockLevel(CHAR * pOut,TPCC_STATE * pTPCC)
{
   pTPCC->uFormId = FORM_STOCKLEVEL;
   FormatFormHdr(pOut,"TPC-C Stock Level",pTPCC);
   sprintf(pOut + strlen(pOut),
      "<PRE>                                 Stock-Level<BR>"
      "Warehouse: %4.4d   District: %2.2d<BR><BR>"
      "Stock Level Threshold: <INPUT NAME=\"TT*\"SIZE=2><BR><BR>"
      "low stock:    <BR><HR>"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Process\">"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Menu\">"
      "</FORM>%s",
      pTPCC->sWId,pTPCC->sDId,HTMLTrailer);
};   // FormatStockLevel

//================================================================================
//
// Function name: FormatFormHdr
//
//================================================================================
VOID FormatFormHdr(CHAR * pOut,CHAR * pTitle,TPCC_STATE * pTPCC)
{
   sprintf(pOut,
```

```
      "%s<HTML><HEAD><TITLE>%s</TITLE></HEAD>"
      "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
      "<INPUT TYPE=\"hidden\" NAME=\"PI*\" VALUE=\"\">"
      "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
      "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">",
      HTTPHdr,pTitle,pTPCC->uFormId,pTPCC->iTermId,pTPCC->iSyncId);
};   // FormatFormHdr

//========================================================================
//
// Function name: FormatRespHdr
//
//========================================================================
VOID FormatRespHdr(CHAR * pOut,CHAR * pTitle,TPCC_STATE * pTPCC)
{
   sprintf(pOut,
      "%s<HTML><HEAD><TITLE>%s</TITLE></HEAD>"
      "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
      "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">",
      HTTPHdr,pTitle,pTPCC->iStatusId,pTPCC->uFormId,
      pTPCC->iTermId,pTPCC->iSyncId);
};   // FormatRespHdr

//========================================================================
//
// Function name: FormatHTMLString
//
//    Encodes HTML special characters.  If necessary, space fills
//    to pOut to total uLen characters.
//
//========================================================================
VOID FormatHTMLString(CHAR * pOut,CHAR * pIn,UINT uLen)
{
   while (uLen && *pIn)
   {
      *pOut++ = *pIn++;
      uLen--;
   };   // while (uLen && *pIn)
   while(uLen--)
      *pOut++ = ' ';
       *pOut = 0;
};   // FormatHTMLString

//========================================================================
//
// Function name: FormatString
//
//    Encodes formatted string for HTML transmission.
//
//========================================================================
VOID FormatString(CHAR * pOut,CHAR * pPic,CHAR * pIn)
{
   while(*pPic)
   {
      if (*pPic == 'X' )
      {
         if (*pIn)
            *pOut++ = *pIn++;
         else
            *pOut++ = ' ';
      }
      else
         *pOut++ = *pPic;
      pPic++;
   };
   *pOut = 0;
};   // FormatString

//========================================================================
// FUNCTION: UtilStrCpy
//
//    Copies n characters from string pSrc to pDst and places a null
//    null character at the end of the destination string.  Unlike
//    strncpy this function ensures that the result string is always
//    null terminated.
//
//========================================================================
VOID UtilStrCpy(CHAR * pDest,CHAR * pSrc,INT n)
{
   strncpy(pDest,pSrc,n);
   pDest[n] = '\0';
   return;
};   // UtilStrCpy

//========================================================================
//
// Function name: CheckNumeric
//
//    Result
//       FALSE - string is all numeric
//       TRUE - sting contains non-numeric characters
//
//========================================================================
BOOL CheckNumeric(CHAR * pNum)
{
   if (*pNum == 0 )
      return(TRUE);
   while (*pNum && isdigit(*pNum))
      pNum++;
   return(*pNum);
};   // CheckNumeric
```

delivery.h

```
// delivery.h
//
// Copyright Unisys, 1999

#define DEFAULTDQSIZE 2000

BOOL DeliveryInit(long lSetThreads,long lSetQSize,char * pszPath);
void DeliveryTerm(void);
BOOL DeliveryPost(DELIVERY_DATA * pdd);
```

delivery.c

```
// delivery.c
//
// Copyright Unisys, 1999

#include <windows.h>
#include <stdio.h>
#include <time.h>
#include <sys\timeb.h>
#include <process.h>

#include "tmon.h"
#include "tpcc.h"
#include "diagio.h"
#include "delivery.h"

CRITICAL_SECTION csDQRead;
CRITICAL_SECTION csDQWrite;
HANDLE hDQRead;
HANDLE hDQStart;
BOOL bDQStarted = FALSE;
BOOL bDQQuit = FALSE;
long lDeliveryThreads;
long lDQSize;
char szPath[200];
long lDeliveryActive = 0;
typedef struct
{
   BOOL bInUse;
   DELIVERY_DATA ddEntry;
} DELIVERY_QUEUE;
DELIVERY_QUEUE * pDQ;
long lDQNextWrite = 0;
long lDQNextRead = 0;
static CHAR * DELIVERY_SERVICE = "DELIVERY";

BOOL DoDQStart(void);
UINT WINAPI DoDelivery(void * Unused);
void CalculateElapsed(int * pElapsed,LPSYSTEMTIME lpBegin,
                      LPSYSTEMTIME lpEnd);

//=========================================================================
//
// Function name: DeliveryInit
//
//=========================================================================
BOOL DeliveryInit(long lSetThreads,long lSetQSize,char * pszPath)
{
   char szDiag[MAX_DIAG_SZ];

   lDeliveryThreads = lSetThreads;
   lDQSize = lSetQSize;
   if (lDQSize <= 0)
      lDQSize = DEFAULTDQSIZE;
   strcpy(szPath,pszPath);
   InitializeCriticalSection(&csDQRead);
   InitializeCriticalSection(&csDQWrite);
   hDQRead = CreateEvent(NULL,TRUE,FALSE,NULL);
   if (!hDQRead)
   {
      wsprintf(szDiag,"DeliveryInit: Create DQRead Event Failure (%ld)\n",
         GetLastError());
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(TRUE);
   };
   pDQ = (DELIVERY_QUEUE *) calloc(lDQSize,sizeof(DELIVERY_QUEUE));
   if (pDQ == NULL)
   {
      DiagIoWrite("DeliveryInit: Allocate Delivery Queue
Failure\n",DIAG_ERROR);
      return(TRUE);
   };
   wsprintf(szDiag,
      "DeliveryInit: Threads = %ld, DQSize(entries) = %ld\n",
      lDeliveryThreads,lDQSize);
   DiagIoWrite(szDiag,DIAG_FORCE);
   return(FALSE);
};   // DeliveryInit

//=========================================================================
//
// Function name: DoDQStart
//
//=========================================================================
BOOL DoDQStart(void)
{
   UINT uThread;
   ULONG hThread;
   DWORD dwRslt;
   char szDiag[MAX_DIAG_SZ];
   void * Unused = NULL;
   int i;

   bDQStarted = TRUE;
   hDQStart = CreateEvent(NULL,TRUE,FALSE,NULL);
   if (!hDQStart)
   {
      wsprintf(szDiag,"DoDQStart: Create Event Failure (%ld)\n",
         GetLastError());
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(TRUE);
   };
   for (i = 0; i < lDeliveryThreads; i++)
   {
      hThread =
         _beginthreadex(NULL,
                        0,
                        DoDelivery,
                        Unused,
                        0,
                        &uThread);
      if (hThread == 0)
      {
         wsprintf(szDiag,
            "DoDQStart: Begin Delivery Thread(%d) Failed(%ld)\n",
            i + 1,errno);
         DiagIoWrite(szDiag,DIAG_ERROR);
         return(TRUE);
      };
      dwRslt = WaitForSingleObject(hDQStart,60000);
```

```
      if (dwRslt == WAIT_TIMEOUT)
      {
         DiagIoWrite("DoDQStart: Wait Delivery Start Timed
Out\n",DIAG_ERROR);
         return(TRUE);
      };
      if (lDeliveryActive != (i + 1))
      {
         wsprintf(szDiag,
            "DoDQStart: Delivery Thread Initialization Failed(%ld)\n",
            i + 1);
         DiagIoWrite(szDiag,DIAG_ERROR);
         return(TRUE);
      };
      ResetEvent(hDQStart);
   };   // for (lDeliveryThreads)
   CloseHandle(hDQStart);
   return(FALSE);
};   // DoDQStart

//============================================================================
//
// Function name: DeliveryTerm
//
//============================================================================
void DeliveryTerm(void)
{
   int i = 0;
   bDQQuit = TRUE;
   while (i < 12 && lDeliveryActive > 0)
   {
      SetEvent(hDQRead);
      Sleep(5000);
      i++;
   };
   if (lDeliveryActive != 0)
   {
      CHAR szDiag[MAX_DIAG_SZ];
      wsprintf(szDiag,
         "DeliveryTerm: %ld DeliveryThreads still active\n",
         lDeliveryThreads);
      DiagIoWrite(szDiag,DIAG_ERROR);
   };
   free(pDQ);
   CloseHandle(hDQRead);
   DeleteCriticalSection(&csDQWrite);
   DeleteCriticalSection(&csDQRead);
   return;
};   // DeliveryTerm

//============================================================================
//
// Function name: DeliveryPost
//
//============================================================================
BOOL DeliveryPost(DELIVERY_DATA * pPost)
{
   DELIVERY_QUEUE * pDQSlot;
   DELIVERY_DATA * pddEntry;
   if (!bDQStarted)
   {
      if (DoDQStart())
      {
         return(TRUE);
      };
   };
   __try
   {
      EnterCriticalSection(&csDQWrite);
      pDQSlot = &pDQ[lDQNextWrite];
      if (pDQSlot->bInUse)
      {
         char szDiag[MAX_DIAG_SZ];
         wsprintf(szDiag,
            "Delivery Post: Queue Limit (%d) Exceeded\n",
            lDQSize);
         DiagIoWrite(szDiag,DIAG_ERROR);
         return(TRUE);
      };
      pddEntry = &pDQSlot->ddEntry;
      memcpy(pddEntry,pPost,sizeof(DELIVERY_DATA));
      pDQSlot->bInUse = TRUE;
      if (lDQNextWrite == lDQNextRead)
         SetEvent(hDQRead);
      lDQNextWrite++;
      if (lDQNextWrite == lDQSize)
         lDQNextWrite = 0;
   }
   __finally
   {
      LeaveCriticalSection(&csDQWrite);
   };
   return(FALSE);
};   // DeliveryPost

//============================================================================
//
// Function name: DoDelivery
//
//============================================================================
UINT WINAPI DoDelivery(void * Unused)
{
   FILE *fpLog;
   char  szLogTitle[300];
   BOOL bFlush = FALSE;
   DELIVERY_QUEUE * pDQSlot;
   DELIVERY_DATA * pddEntry;
   DELIVERY_DATA * pdd;
   TMON_STATE tsState;
   TMON_STATE * pTMon;
   BOOL bTMRslt;
   BOOL bTPRslt;
   INT iTPRslt;
   long lMyId;
   char szTMErrTxt[500];
   char szDiag[MAX_DIAG_SZ];
   int iElapsed;
   int iInx;

   lMyId = InterlockedIncrement(&lDeliveryActive);
   pTMon = &tsState;
   pTMon->pszErrTxt = szTMErrTxt;
```

```
if (TMInit(pTMon))
{
   wsprintf(szDiag,"DoDelivery(%ld): TMInit %s\n",lMyId,szTMErrTxt);
   DiagIoWrite(szDiag,DIAG_ERROR);
   InterlockedDecrement(&lDeliveryActive);
   SetEvent(hDQStart);
   return(1);
};
wsprintf(szLogTitle,"%sdelilog%ld",szPath,lMyId);
fpLog = fopen(szLogTitle,"ab");
     if (!fpLog)
{
   wsprintf(szDiag,
      "DoDelivery(%ld): LogFile %s Open Failed (%ld)\n",
      lMyId,szLogTitle,GetLastError());
   DiagIoWrite(szDiag,DIAG_ERROR);
   InterlockedDecrement(&lDeliveryActive);
   SetEvent(hDQStart);
   return(1);
};
wsprintf(szDiag,"DoDelivery(%ld): Initialized\n",lMyId);
DiagIoWrite(szDiag,DIAG_FORCE);
SetEvent(hDQStart);

while (!bDQQuit)
{
   EnterCriticalSection(&csDQRead);
   WaitForSingleObject(hDQRead,INFINITE);
   if (bDQQuit)
   {
      LeaveCriticalSection(&csDQRead);
      break;
   };
   pDQSlot = &pDQ[lDQNextRead];
   if (!pDQSlot->bInUse)
   {
      wsprintf(szDiag,
         "DoDelivery(%ld): QSlot for Read Not InUse (%ld)\n",
      lMyId);
      DiagIoWrite(szDiag,DIAG_ERROR);
      LeaveCriticalSection(&csDQRead);
      break;
   };
   pddEntry = &pDQSlot->ddEntry;
   pdd = (DELIVERY_DATA *) pTMon->pTMData;
   memcpy(pdd,pddEntry,sizeof(DELIVERY_DATA));
   EnterCriticalSection(&csDQWrite);
   pDQSlot->bInUse = FALSE;
   lDQNextRead++;
   if (lDQNextRead == lDQSize)
      lDQNextRead = 0;
   if (lDQNextRead == lDQNextWrite)
      ResetEvent(hDQRead);
   LeaveCriticalSection(&csDQWrite);
   LeaveCriticalSection(&csDQRead);
   // Process delivery transaction
   bTMRslt = TMTran(DELIVERY_SERVICE,pTMon,&bTPRslt,&iTPRslt);
   pdd = (DELIVERY_DATA *) pTMon->pTMData;
   if (bTMRslt)
   {
         wsprintf(szDiag,"DoDelivery(%ld): Error
(%s)\n",lMyId,szTMErrTxt);
         DiagIoWrite(szDiag,DIAG_ERROR);
         break;
      };
      pdd = (DELIVERY_DATA *) pTMon->pTMData;
      GetLocalTime(&pdd->EndTime);
      iElapsed = 9999999;
      if (!bTPRslt)
         CalculateElapsed(&iElapsed,&pdd->QTime,&pdd->EndTime);
      iInx =  wsprintf(szDiag,

"%2.2d/%2.2d/%2.2d,%2.2d:%2.2d:%2.2d:%3.3d,%2.2d:%2.2d:%2.2d:%3.3d,"
         "%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d\r\n",
         pdd->EndTime.wYear - 1900,pdd->EndTime.wMonth,pdd->EndTime.wDay,
         pdd->QTime.wHour,pdd->QTime.wMinute,
         pdd->QTime.wSecond,pdd->QTime.wMilliseconds,
         pdd->EndTime.wHour,pdd->EndTime.wMinute,
         pdd->EndTime.wSecond,pdd->EndTime.wMilliseconds,
         iElapsed,pdd->w_id,pdd->o_carrier_id,
         pdd->o_id[0],pdd->o_id[1],pdd->o_id[2],pdd->o_id[3],pdd->o_id[4],
         pdd->o_id[5],pdd->o_id[6],pdd->o_id[7],pdd->o_id[8],pdd->o_id[9]
);
      fwrite(szDiag,iInx,1,fpLog);
   };   // while !bDQQuit

   if (fpLog)
      fclose(fpLog);
   TMDone(pTMon);
   InterlockedDecrement(&lDeliveryActive);
   wsprintf(szDiag,"DoDelivery(%ld): Shutdown\n",lMyId);
   DiagIoWrite(szDiag,DIAG_FORCE);
   return(0);
};   // DoDelivery

//=========================================================================
//
// Function name: CalculateElapsed (milliseconds)
//
//=========================================================================
void CalculateElapsed(int * pElapsed,LPSYSTEMTIME lpBegin,
                      LPSYSTEMTIME lpEnd)
{
   int tmBegin;
   int tmEnd;
   tmBegin = (lpBegin->wHour * 3600000) + (lpBegin->wMinute * 60000) +
      (lpBegin->wSecond * 1000) + lpBegin->wMilliseconds;
   tmEnd = (lpEnd->wHour * 3600000) + (lpEnd->wMinute * 60000) +
      (lpEnd->wSecond * 1000) + lpEnd->wMilliseconds;
   *pElapsed = tmEnd - tmBegin;
   // Check for day boundry, this will function for 24 hour period but
   // will fail over a 48 hours period.
   if (*pElapsed < 0)
      *pElapsed = *pElapsed + (24 * 60 * 60 * 1000);
   return;
};   // CalculateElapsed
```

tmon.h

```
// tmon.h

typedef struct
{
   CHAR * pszErrTxt;                    // Error text
   CHAR * pTMData;                      // TM buffer area
   LONG lTMDataLen;                     // TM buffer len
} TMON_STATE;

VOID TMonInit(INT iSetMaxMsg);
VOID TMonTerm(VOID);
BOOL TMInit(TMON_STATE * pTMon);
VOID TMDone(TMON_STATE * pTMon);
BOOL TMTran(CHAR * pService,TMON_STATE * pTMon,
            BOOL * bTPRslt,INT * iTPRslt);
BOOL TMPost(CHAR * pService,TMON_STATE * pTMon);
```

tmon.c

```
// tmon.c
//
// Copyright Unisys, 1997
//
#include <windows.h>
#include <stdio.h>
#include <atmi.h>
#include "tmon.h"

INT iTMMaxSz;

//========================================================================
//
// Function name: TMonInit
//
//========================================================================
VOID TMonInit(INT iSetMaxMsg)
{
   iTMMaxSz = iSetMaxMsg;
};   // TMonInit


//========================================================================
//
// Function name: TMonTerm
//
//========================================================================
VOID TMonTerm(VOID)
{
};   // TMonTerm

//========================================================================
//
// Function name: TMInit
//
//    Result:
//       FALSE    Initialization completed successfully
//       TRUE     Initialization failed
//
//========================================================================
BOOL TMInit(TMON_STATE * pTMon)
{
   BOOL bRslt = FALSE;
   TPINIT * tpinfo;

   // Must have ErrTxt message area set before init
   if (pTMon->pszErrTxt == NULL)
      return(TRUE);
   tpinfo = (TPINIT *) tpalloc("TPINIT",NULL,TPINITNEED(20));
   memset(tpinfo,0,sizeof(TPINIT));
   tpinfo->flags=TPMULTICONTEXTS;
   sprintf(tpinfo->cltname,"tpcc%d",GetCurrentThreadId());

   if (tpinit(tpinfo) == -1)
   {
      sprintf(pTMon->pszErrTxt,"TPInit Failed(%ld)",tperrno);
      bRslt = TRUE;
   }
   else
   {
      pTMon->pTMData = tpalloc("CARRAY",NULL,iTMMaxSz);
      if (pTMon->pTMData == NULL)
      {
         sprintf(pTMon->pszErrTxt,"TPAlloc Failed(%ld)",tperrno);
         bRslt = TRUE;
      };
   };

   return(bRslt);

};   // TMInit

//========================================================================
//
// Function name: TMDone
//
//========================================================================
VOID TMDone(TMON_STATE * pTMon)
{
   tpfree(pTMon->pTMData);
   tpterm();
};   // TMDone

//========================================================================
//
// Function name: TMTran
//
//    Result:
//       FALSE    call completed. bTPRslt contains outcome (FALSE tran
//                success).  iTPRslt contains application returned
//                result code.
//       TRUE     TM interface error, ErrTxt has diagnostic.
//
//========================================================================
BOOL TMTran(CHAR * pService,TMON_STATE * pTMon,
            BOOL * bTPRslt,INT * iTPRslt)
{
   BOOL bRslt = FALSE;
   INT iGRply;
```

```
   iGRply = tpcall(pService,pTMon->pTMData,iTMMaxSz,
      &pTMon->pTMData,&pTMon->lTMDataLen,TPNOTIME | TPSIGRSTRT);
   if (iGRply != -1)
   {
      *iTPRslt = tpurcode;
      *bTPRslt = FALSE;
   }
   else
   if (tperrno == TPESVCFAIL)
   {
      *iTPRslt = tpurcode;
      *bTPRslt = TRUE;
   }
   else
   {
      sprintf(pTMon->pszErrTxt,"TPCall Failed (%ld)",tperrno);
      bRslt = TRUE;
   };
   return(bRslt);
};   // TMTran

//====================================================================
//
// Function name: TMPost
//
//    Result:
//       FALSE    transaction submitted with no response expected
//       TRUE     tpacall failed, ErrTxt has diagnostic
//
//====================================================================
BOOL TMPost(CHAR * pService,TMON_STATE * pTMon)
{
   BOOL bRslt = FALSE;
   INT iCD;

   iCD = tpacall(pService,pTMon->pTMData,iTMMaxSz,TPNOREPLY);
   if (iCD == -1)
   {
      sprintf(pTMon->pszErrTxt,"TPACall Failed (%ld)",tperrno);
      bRslt = TRUE;
   };
   return(bRslt);
};   // TMPost
```

term.h

```
// term.h

#include <sys\timeb.h>

#define TMILLI_TIMEOUT 3600000       // One hour

typedef struct
{
   BOOL bInUse;                        // In use flag
   INT iTermId;                        // TermId
   LPVOID ConnID;                      // Connection Id
   INT  iSyncId;                       // Sync Id
   SHORT sWId;                         // TPCC WareHouse Id
   SHORT sDId;                         // TPCC District Id
   struct _timeb tbLastAccess;         // Last activity timestamp
} TERM_STATE;

BOOL TermInit(INT iSetMaxTerm);
VOID TermTerm(VOID);
TERM_STATE * TermAlloc(VOID);
TERM_STATE * TermGet(INT iTermId);
BOOL TermFree(INT iTermId);
```

term.c

```
// term.c
//
// Copyright Unisys, 1997
//
#include <windows.h>
#include <stdio.h>
#include "diagio.h"
#include "timesupp.h"
#include "term.h"

TERM_STATE * pTArray;
INT iNextTerm = 0;
INT iMaxTerm = 0;
CRITICAL_SECTION csTerm;

VOID TermMaint(VOID);

//====================================================================
//
// Function name: TermInit
//    Creates and initializes the first TERMINITIAL TArray entries.
//    Initializes critical section to control access to TArray.  Assumes
//    access to function is single threaded, no other threads will start
//    until this function completes and that function is called once
//    (DLL_PROCESS_ATTACH).
//
// Returns:
//    FALSE TArray allocated and initialized
//    TRUE  TArray allocation failure
//
//====================================================================
BOOL TermInit(INT iSetMaxTerm)
{
   INT iTermId;
   CHAR szDiag[MAX_DIAG_SZ];
   if (pTArray != NULL)
   {
      sprintf(szDiag,"TermInit(%ld): TArray Already Initialized\n",
         GetCurrentThreadId());
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(TRUE);
   };
   InitializeCriticalSection(&csTerm);
   iMaxTerm = iSetMaxTerm;
   pTArray = (TERM_STATE *) malloc(sizeof(TERM_STATE) * (iMaxTerm + 1));
   if (pTArray == NULL)
   {
      sprintf(szDiag,"TermInit(%ld): malloc failed (%ld)\n",
         GetCurrentThreadId(),GetLastError());
```

```
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(TRUE);
   }
   for (iTermId = 1; iTermId <= iMaxTerm; iTermId++)
      TermFree(iTermId);
   iNextTerm = 1;
   return(FALSE);
};   // TermInit

//=========================================================================
//
// Function name: TermTerm
//    Frees TArray and deletes csTerm critical section.  Assumes access
//    to function is single threaded and no other threads are actively
//    accessing TArray entries (DLL_PROCESS_DETACH).
//
//=========================================================================
VOID TermTerm(VOID)
{
   DeleteCriticalSection(&csTerm);
   if (pTArray != NULL)
      free(pTArray);
   iNextTerm = 0;
   iMaxTerm =0;
};   // TermTerm

//=========================================================================
//
// Function name: TermAlloc
//    Allocates empty TArray.  Uses iNextTerm to start search.
//
// Returns:
//    > 0   TArray entry index (iTermId)
//    < 0   Empty TArray entry not available
//
//=========================================================================
TERM_STATE * TermAlloc(VOID)
{
   INT iTermId = -1;
   if (pTArray == NULL)
   {
      CHAR szDiag[MAX_DIAG_SZ];
      sprintf(szDiag,"TermAlloc(%ld): Term Array Not Allocated\n",
         GetCurrentThreadId());
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(NULL);
   };
   EnterCriticalSection(&csTerm);
   __try
   {
      while(iNextTerm <= iMaxTerm)
      {
         if (!pTArray[iNextTerm].bInUse)
         {
            pTArray[iNextTerm].bInUse = TRUE;
            _ftime(&pTArray[iNextTerm].tbLastAccess);
            iTermId = iNextTerm;
            iNextTerm++;
            break;
         };
         iNextTerm++;
      };   // while(iNextTerm <= iMaxTerm) (1st Try)
      if (iTermId <= 0)
      {
         // No entry found.  Perform maint and try again
         TermMaint();
         iNextTerm = 1;
         while(iNextTerm <= iMaxTerm)
         {
            if (!pTArray[iNextTerm].bInUse)
            {
               pTArray[iNextTerm].bInUse = TRUE;
               _ftime(&pTArray[iNextTerm].tbLastAccess);
               iTermId = iNextTerm;
               iNextTerm++;
               break;
            };
            iNextTerm++;
         };   // while(iNextTerm <= iMaxTerm) (2nd Try)
      };   // if (iTermId <= 0)
      if (iTermId <= 0)
         iNextTerm = 1;
   }
   __finally
   {
      LeaveCriticalSection(&csTerm);
   };

   if (iTermId > 0)
      return(&pTArray[iTermId]);
   else
      return(NULL);


};   // TermAlloc

//=========================================================================
//
// Function name: TermMaint
//    Clears entries whose last access time exceeds TMILLI_TIMEOUT.
//    Assumes caller has entered csTerm.
//
//=========================================================================
VOID TermMaint(VOID)
{
   INT iTermId;
   TMILLI tmElapsed;
   // Free entries that have timed out
   for (iTermId = 1; iTermId <= iMaxTerm; iTermId++)
   {
      if (pTArray[iTermId].bInUse)
      {
         tmElapsed = TimebElapsed(&pTArray[iTermId].tbLastAccess);
         if (tmElapsed > TMILLI_TIMEOUT)
            TermFree(iTermId);
      };
   };
};   // TermMaint

//=========================================================================
//
// Function name: TermGet
```

```
//    Returns pointer to TArray slot at iTermId.
//
// Returns:
//    FALSE TArray entry made available
//    TRUE  iTermId invalid.
//
//=========================================================================
TERM_STATE * TermGet(INT iTermId)
{
   TERM_STATE * pTerm;
   TMILLI tmElapsed;
   if (iTermId <= 0 || iTermId > iMaxTerm)
   {
      CHAR szDiag[MAX_DIAG_SZ];
      sprintf(szDiag,"TermGet(%ld): Invalid TermId (%ld)\n",
         GetCurrentThreadId(),iTermId);
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(NULL);
   };
   pTerm = &pTArray[iTermId];
   if (!pTerm->bInUse)
      return(NULL);
   tmElapsed = TimebElapsed(&pTerm->tbLastAccess);
   if (tmElapsed > TMILLI_TIMEOUT)
      return(NULL);   // Entry destined to be freed by maint
   _ftime(&pTArray[iTermId].tbLastAccess);
   return(&pTArray[iTermId]);
};   // TermGet

//=========================================================================
//
// Function name: TermFree
//    Initializes contents of TArray slot at iTermId.
//
// Returns:
//    FALSE TArray entry made available
//    TRUE  iTermId invalid.
//
//=========================================================================
BOOL TermFree(INT iTermId)
{
   TERM_STATE * pTerm;
   if (iTermId <= 0 || iTermId > iMaxTerm)
   {
      CHAR szDiag[MAX_DIAG_SZ];
      sprintf(szDiag,"TermFree(%ld): Invalid TermId (%ld)\n",
         GetCurrentThreadId(),iTermId);
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(TRUE);
   };
   pTerm = &pTArray[iTermId];
   pTerm->ConnID = 0;
   pTerm->sWId = 0;
   pTerm->sDId = 0;
   pTerm->iSyncId = 0;
   pTerm->iTermId = iTermId;
   TimebClear(&pTerm->tbLastAccess);
   pTerm->bInUse = FALSE;
};   // TermFree
```

timesupp.h

```
// timesupp.h

#include <windows.h>
#include <time.h>
#include <sys\timeb.h>

#define TIMEBSEED_MOD 10000
#define TIMEBSEED_SHIFT 1000
#define TIMEB_STRING_SZ 23
#define TIMEB_STRING_DATESZ 10
#define TIMEB_STRING_TIMEOFFSET 11
#define TIMEB_STRING_TIMESZ 12

typedef ULONG TMILLI;

TMILLI TimebDiff(struct _timeb * p_tb1, struct _timeb * p_tb2);
VOID TimebCopy(struct _timeb * p_tbDest, struct _timeb * p_tbSource);
TMILLI TimebElapsed(struct _timeb * p_tb1);
VOID TimebClear(struct _timeb * p_tb1);
CHAR * TimebToString(struct _timeb * p_tb1,CHAR * psz,BOOL bMillis);
BOOL TimebFromString(struct _timeb * p_tb1,CHAR * psz);
VOID TimebAddSecs(struct _timeb * p_tb1,INT iSeconds);
ULONG TimebSeed(VOID);
```

timesupp.c

```
// timesupp.c
//
// Copyright Unisys, 1997
//

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

//=========================================================================
//
// Function name: TimebCopy
//    Structure contents copy of _timeb source to _timeb dest.
//
//=========================================================================
VOID TimebCopy(struct _timeb * p_tbDest, struct _timeb * p_tbSource)
{
   p_tbDest->time = p_tbSource->time;
   p_tbDest->millitm = p_tbSource->millitm;
   p_tbDest->dstflag = p_tbSource->dstflag;
   p_tbDest->timezone = p_tbSource->timezone;

};   // TimebCopy

//=========================================================================
//
// Function name: TimebDiff
//    Time difference in milliseconds between _timeb _t1 and _timeb _t2.
//
//=========================================================================
TMILLI TimebDiff(struct _timeb * p_tb1, struct _timeb * p_tb2)
{
   LONG lRslt;
```

```
   lRslt = ((p_tb2->time - p_tb1->time) * 1000) +
               (p_tb2->millitm - p_tb1->millitm);
   if (lRslt < 0)
      return(0);
   else
      return((TMILLI) lRslt);

};   // TimebDiff

//=========================================================================
//
// Function name: TimebElaspsed
//
//=========================================================================
TMILLI TimebElapsed(struct _timeb * p_tb1)
{
   struct _timeb _tb2;
   _ftime(&_tb2);
   return (TimebDiff(p_tb1,&_tb2));

};   // TimebElapsed

//=========================================================================
//
// Function name: TimebClear
//
//=========================================================================
VOID TimebClear(struct _timeb * p_tb1)
{
   p_tb1->time = 0;
   p_tb1->millitm = 0;

};   // TimebClear

//=========================================================================
//
// Function name: TimebToString
//    Converts timeb to yyyy:mm:dd,hh:mm:ss.sss format
//
//=========================================================================
CHAR * TimebToString(struct _timeb * p_tb1,CHAR * psz,BOOL bMillis)
{
   struct tm * ptm;
   ptm = localtime(&p_tb1->time);
   sprintf(psz,"%4.4d/%2.2d/%2.2d,%2.2d:%2.2d:%2.2d",
      ptm->tm_year + 1900,ptm->tm_mon + 1,ptm->tm_mday,
      ptm->tm_hour,ptm->tm_min,ptm->tm_sec);
   if (bMillis)
      sprintf(psz + strlen(psz),".%3.3d",p_tb1->millitm);
   return(psz);
};   // TimebToString

//=========================================================================
//
// Function name: TimebFromString
//    Converts yyyy:mm:dd,hh:mm:ss.sss (TimebToString) format to timeb
//
//=========================================================================
BOOL TimebFromString(struct _timeb * p_tb1,CHAR * psz)
{
   struct tm tmTime;
   struct tm * ptm;
   UINT uLen;

   ptm = &tmTime;
   uLen = strlen(psz);
   if (uLen < (TIMEB_STRING_SZ - 4))   // millis are optional
   {
      p_tb1->time = 0;
      p_tb1->millitm = 0;
      return (TRUE);
   };
   // Clear fields that won't be set
   ptm->tm_wday = 0;
   ptm->tm_yday = 0;
   ptm->tm_isdst = -1;
   // Set tm struct fields from string
   ptm->tm_year = (atoi(psz)) - 1900;
   psz += 5;
   ptm->tm_mon = (atoi(psz)) - 1;
   psz += 3;
   ptm->tm_mday = atoi(psz);
   psz += 3;
   ptm->tm_hour = atoi(psz);
   psz += 3;
   ptm->tm_min = atoi(psz);
   psz +=3;
   ptm->tm_sec = atoi(psz);
   if (uLen >= TIMEB_STRING_SZ)   // Millis present
   {
      psz += 3;
      p_tb1->millitm = atoi(psz);
   };
   p_tb1->time = mktime(ptm);
   return(FALSE);
};   // TimebFromString

//=========================================================================
//
// Function name: TimebAddSecs
//
//=========================================================================
VOID TimebAddSecs(struct _timeb * p_tb1,INT iSeconds)
{
   p_tb1->time += iSeconds;
};   // TimebAddSecs

//=========================================================================
//
// Function name: TimebSeed
//
//=========================================================================
ULONG TimebSeed(VOID)
{
   ULONG ulSeed;
   struct _timeb tb_1;
   _ftime(&tb_1);
   ulSeed = ((tb_1.time % TIMEBSEED_MOD) * TIMEBSEED_SHIFT) +
tb_1.millitm;
   return(ulSeed);
};   // TimebSeed
```

diagio.h

```
// diagio.h

// Environment variable defaults
#define DEFAULTDIAGLEVEL DIAG_INFO
#define DEFAULTEVENTLOG 0

#define DIAGNOSTICS TRUE
#define MAX_DIAG_SZ 2000

// Severity level of diagnostic report
#define DIAG_FORCE 1
#define DIAG_ERROR 2
#define DIAG_STATE 3
#define DIAG_INFO 4

VOID DiagIoInit(CHAR * pDiagId,BOOL bConsole,BOOL bEvent,UINT uLevel);
VOID DiagIoTerm(VOID);
VOID DiagIoWrite(CHAR * pDiagBuffer, UINT uSeverity);
```

diagio.c

```
// diagio.c
//
// Copyright Unisys, 1997
//
#include <windows.h>
#include <stdio.h>
#include "diagio.h"

CRITICAL_SECTION csDiagIo;
HANDLE hEventLog = NULL;
UINT uDiagLevel;
BOOL bEventLog;
BOOL bConsoleLog;
CHAR * pDiagHdr;
CHAR * pEventHost;
CHAR * pErrHdr =
   {"*** ERROR *** ERROR *** ERROR *** ERROR *** ERROR ***"};

INT WriteEventLog(CHAR * pDMsgs[],UINT uMsgCnt,UINT uSeverity);

//============================================================================
//
// Function name: DiagIoInit
//
//============================================================================
VOID DiagIoInit(CHAR * pDiagId,BOOL bConsole,BOOL bEvent,UINT uLevel)
{
   if (DIAGNOSTICS)
   {
      InitializeCriticalSection(&csDiagIo);

      uDiagLevel = uLevel;
      bEventLog = bEvent;
      bConsoleLog = bConsole;
      pEventHost = (CHAR *) malloc(10);
      strcpy(pEventHost,"");  // local host
      pDiagHdr = (CHAR *) malloc(strlen(pDiagId) + 1);
      strcpy(pDiagHdr,pDiagId);
      if (bEventLog)
      {
         hEventLog = RegisterEventSource(pEventHost,pDiagId);
         if (hEventLog == NULL)
         {
            bEventLog = FALSE;
            if (bConsoleLog)
               fprintf(stdout,
                  "%s: Event Log Register Failed (%ld)\n"
                  "Event Log Will NOT be Used\n",
                  pDiagHdr,GetLastError());
         }
         else
         {
            if (bConsoleLog)
               fprintf(stdout,"%s: Event Logging to LocalHost as %s\n",
                  pDiagHdr,pDiagHdr);
         };
      };   // if bEventLog
   };   // if Diagnostics
};   // DiagIoInit


//============================================================================
//
// Function name: DiagIoTerm
//
//============================================================================
VOID DiagIoTerm(VOID)
{
   if (DIAGNOSTICS)
   {
      DeleteCriticalSection(&csDiagIo);
      if (hEventLog != NULL)
         DeregisterEventSource(hEventLog);
      free(pDiagHdr);
      free(pEventHost);
   };
};   // DiagIoTerm

//============================================================================
//
// Function name: DiagIoWrite
//
//============================================================================
VOID DiagIoWrite(CHAR * pDiagBuffer, UINT uSeverity)
{
   CHAR * pDMsgs[3];
   UINT uMsgCnt = 0;
   INT iERslt = 0;
   if (DIAGNOSTICS)
   {
      if (uDiagLevel >= uSeverity)
      {
         EnterCriticalSection(&csDiagIo);
         __try
         {
            if (uSeverity == DIAG_ERROR)
            {
               pDMsgs[0] = pDiagHdr;
```

```
               pDMsgs[1] = pErrHdr;
               pDMsgs[2] = pDiagBuffer;
               uMsgCnt = 3;
            }
            else
            {
               pDMsgs[0] = pDiagHdr;
               pDMsgs[1] = pDiagBuffer;
               uMsgCnt = 2;
            };
            if (bEventLog)
               iERslt = WriteEventLog(pDMsgs,uMsgCnt,uSeverity);
            if (bConsoleLog)
            {
               if (uMsgCnt == 3)
                  fprintf(stdout,"\n%s:
%s\n%s",pDMsgs[0],pDMsgs[1],pDMsgs[2]);
               else
                  fprintf(stdout,"\n%s: %s",pDMsgs[0],pDMsgs[1]);
               if (iERslt != 0)
                  fprintf(stdout,
                     "EventLog Write Failed (%ld), No Longer in Use\n",
                     iERslt);
            };
         }
         __finally
         {
            LeaveCriticalSection(&csDiagIo);
         };
      };   // if uDiagLevel >= uSeverity
   };   // if Diagnostics
};   // DiagIoWrite

INT WriteEventLog(CHAR * pDMsgs[],UINT uMsgCnt,UINT uSeverity)
{
   WORD wType;
   WORD wCount;
   wCount = uMsgCnt;
   switch (uSeverity)
   {
      case DIAG_ERROR:
         wType = EVENTLOG_ERROR_TYPE;
         break;
      default:
         wType = EVENTLOG_INFORMATION_TYPE;
         break;
   };
   if (wType != 0)
   {
      if (!ReportEvent(hEventLog,           // event log handle
                       wType,               // event type
                       0,                   // category zero
                       uSeverity,           // no event identifier
                       NULL,                // no user security identifier
                       wCount,              // # of substitution strings
                       0,                   // no binary data
                       (LPCTSTR *) pDMsgs,  // address of string array
                       NULL))               // address of binary
      {
         DeregisterEventSource(hEventLog);
         hEventLog = NULL;
         bEventLog = FALSE;
         return(GetLastError());
      };   // ReportEvent failed
   };   // if wType != 0
   return(0);
};   // WriteEventLog
```

Tuxedo Client

makefile

```
#==============================================================+
#        Copyright (c) 1999  Unisys Corporation                |
#==============================================================+
# FILENAME
#      Makefile
# DESCRIPTION
#      Makefile for tpccsvr - Tuxedo server
#==============================================================+

#==============================================================+
#  MACROS - from prefix.mk
#==============================================================+

# DESCRIPTION: Remove a file.  No error even if file does not exist.
REMOVE=rm -f

# DESCRIPTION: Copy a file.
CP=cp

# DESCRIPTION: The linker.
LD=ld

# DESCRIPTION: Command for linking executables.
LINK=cc -o $@

# DESCRIPTION: The flag for include path entries.
I_SYM=-I

# DESCRIPTION: C compiler flags.
CFLAGS=-O

# DESCRIPTION: Libraries for linking executables.
LIBS=-lclntsh -ljava \
       -lsocket -lnsl -lgen -ldl -lmw_UXW -lelf -lm \
       -lmw_UXW -lgen -ldshm -Kthread -ltux -lbuft -ltux2 -lfml -lfml32 -
lgp

# DESCRIPTION: Flags for linking executables.
LDFLAGS=-L$(ORACLE_HOME)/lib -L$(ORACLE_HOME)/rdbms/lib

#==============================================================+
all: tpccsvr

tidy:
       $(REMOVE) *.o
```

```
clean:
	$(REMOVE) *.o tpccsvr

#include ../../../prefix.mk
SDIR=.

INCLUDE=$(I_SYM). $(I_SYM)$(SDIR) $(I_SYM)$(ORACLE_HOME)/rdbms/demo \
	 $(I_SYM)$(ORACLE_HOME)/rdbms/public \
	 $(I_SYM)$(ORACLE_HOME)/rdbms/include \
	 $(I_SYM)$(ORACLE_HOME)/network/public \
	 $(I_SYM)$(ORACLE_HOME)/plsql/public \
	$(I_SYM)$(TUXDIR)/include

MEMBS=
OBJS=orasvr.o pldel.o plnew.o plord.o plpay.o plsto.o svrjack.o tpccpl.o

files:

compile: $(OBJS)

cleanup:
	$(REMOVE) $(OBJS) *.exe

orasvr.o: orasvr.c tpcc.h tpccflags.h
	$(CC) $(CFLAGS) $(INCLUDE) -c orasvr.c

pldel.o: pldel.c tpcc.h tpccflags.h
	$(CC) $(CFLAGS) $(INCLUDE) -c pldel.c

plnew.o: plnew.c tpcc.h tpccflags.h
	$(CC) $(CFLAGS) $(INCLUDE) -c plnew.c

plord.o: plord.c tpcc.h tpccflags.h
	$(CC) $(CFLAGS) $(INCLUDE) -c plord.c

plpay.o: plpay.c tpcc.h tpccflags.h
	$(CC) $(CFLAGS) $(INCLUDE) -c plpay.c

plsto.o: plsto.c tpcc.h tpccflags.h
	$(CC) $(CFLAGS) $(INCLUDE) -c plsto.c

svrjack.o: svrjack.c tpcc.h tpccflags.h
	$(CC) $(CFLAGS) $(INCLUDE) -c svrjack.c

tpccpl.o: tpccpl.c tpcc.h tpccflags.h
	$(CC) $(CFLAGS) $(INCLUDE) -c tpccpl.c

tpccsvr: $(OBJS)
	$(LINK) $(LDFLAGS) $(OBJS) $(LIBS)
```

tpcc.h

```
// tpcc.h
// Copyright (c) 1998 1999 Unisys Corporation
// Copyright (c) 1995  Oracle Corp, Redwood Shores, CA

#ifndef TPCC_H
#define TPCC_H

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <ctype.h>
#include <string.h>
#define boolean int
#include <oratypes.h>
#include <oci.h>
#include <ocidfn.h>
#include "tpccflags.h"
#ifdef ORA_NT
#include <windows.h>
#else
#define BOOL boolean
#endif

#ifndef ORA_NT
typedef struct
{
    long wYear;
    long wMonth;
    long wDayOfWeek;
    long wDay;
    long wHour;
    long wMinute;
    long wSecond;
    long wMillisecond;
}SYSTEMTIME;
#define LPSYSTEMTIME SYSTEMTIME*
#endif

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 ();
#ifndef TUX
extern void userlog();
#endif

// TPCC Service return codes
#define SVC_BADITEMID 1
#define SVC_NOERROR 0
#define SVCERR_DEADLOCK -1
#define SVCERR_NOCUSTOMER -2
#define SVCERR_NOORDERS -3
#define SVCERR_OCI -4

/* 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

#define FULLDATE "dd-mon-yy.hh:mi:ss"
#define SHORTDATE "dd-mm-yyyy"

extern int tkvcninit ();
extern int tkvcpinit ();
extern int tkvcoinit ();
extern int tkvcdinit ();
extern int tkvcsinit ();

extern int tkvcn ();
extern int tkvcp ();
extern int tkvco ();
extern int tkvcd ();
extern int tkvcs ();

extern void tkvcndone ();
extern void tkvcpdone ();
extern void tkvcodone ();
extern void tkvcddone ();
extern void tkvcsdone ();

extern int tkvcss (); /* for alter session to get memory size and trace */
extern boolean multitranx;
extern int ord_init;


extern int errrpt (csrdef *,csrdef *);
extern int ocierror(char *fname, int lineno,OCIError *errhp, sword
status);
extern int sqlfile(char *fname, text *linebuf);

extern FILE *fopen ();
extern int proc_no;
extern int doid[];

extern int execstatus;
extern int errcode;

extern OCIEnv *tpcenv;
extern OCIServer *tpcsrv;
extern OCIError *errhp;
extern OCISvcCtx *tpcsvc;
extern OCISession *tpcusr;
extern OCIStmt *curntest;
/* The bind and define handles for each transaction are
   included in their respective header files. */



/* for stock-level transaction */

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

/* for delivery transaction */

extern int del_o_id[10];
extern int carrier_id;
extern int retries;

/* for order-status transaction */

extern int bylastname;
extern char c_last[17];
extern char c_first[17];
extern char c_middle[3];
extern double c_balance;
extern int o_id;
extern text o_entry_d[20];
extern int o_carrier_id;
extern int o_ol_cnt;
extern int ol_supply_w_id[15];
extern int ol_i_id[15];
extern int ol_quantity[15];
extern int ol_amount[15];
ub4    ol_del_len[15];
extern OCIDate ol_delivery_d[15];

/* for payment transaction */

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

/* for new order transaction */

extern int nol_i_id[15];
extern int nol_supply_w_id[15];
extern int nol_quantity[15];
extern int nol_quanti10[15];
extern int nol_quanti91[15];
```

```
extern int nol_ytdqty[15];
extern int nol_amount[15];
extern int o_all_local;
#ifdef SVRUNIX
extern float w_tax;
extern float d_tax;
#else
extern int w_tax;
extern int d_tax;
#endif
extern float total_amount;
extern char i_name[15][25];
extern int i_name_strlen[15];
extern ub2 i_name_strlen_len[15];
extern ub2 i_name_strlen_rcode[15];
extern ub4 i_name_strlen_csize;
extern int s_quantity[15];
//extern char brand_gen[15];
//extern ub2 brand_gen_len[15];
//extern ub2 brand_gen_rcode[15];
//extern ub4 brand_gen_csize;
extern int i_price[15];
extern char brand_generic[15][1];
extern int status;

/* Miscellaneous */
extern OCIDate cr_date;
extern OCIDate c_since;
extern OCIDate o_entry_d_base;
extern OCIDate ol_d_base[15];

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#define VER7            2

#define NA              -1      /* ANSI SQL NULL */
#define NLT             1       /* length for string null terminator */
#define DEADLOCK        60      /* ORA-00060: deadlock */
#define NO_DATA_FOUND   1403    /* ORA-01403: no data found */
#define NOT_SERIALIZABLE  8177  /* ORA-08177: transaction not serializable
*/
#define SNAPSHOT_TOO_OLD  1555  /* ORA-01555: snapshot too old */

#ifndef NULLP
# define NULLP (void *)NULL
#endif /* NULLP */

#define ADR(object) ((ub1 *)&(object))
#define SIZ(object) ((sword)sizeof(object))

typedef char date[24+NLT];
typedef char varchar2;

#define min(x,y)  (((x) < (y)) ? (x) : (y))
```

```
#define OCIERROR(errp,function)\
        ocierror(__FILE__,__LINE__,(errp),(function));

#define OCIBND(stmp, bndp, errp, sqlvar, progv, progvl, ftype)\
       ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
       ocierror(__FILE__,__LINE__, (errp), \
             OCIBindByName((stmp), &(bndp), (errp), \
                    (text *)(sqlvar), strlen((sqlvar)),\
                    (progv), (progvl), (ftype),0,0,0,0,0,OCI_DEFAULT));

#define
OCIBNDRA(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode) \
       ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
       ocierror(__FILE__,__LINE__,(errp), \
            OCIBindByName((stmp),&(bndp),(errp),(text
*)(sqlvar),strlen((sqlvar)),\

(progv),(progvl),(ftype),(indp),(alen),(arcode),0,0,OCI_DEFAULT));

#define
OCIBNDRAD(stmp,bndp,errp,sqlvar,progvl,ftype,indp,ctxp,cbf_nodata,cbf_data
) \
       ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
       ocierror(__FILE__,__LINE__,(errp), \
            OCIBindByName((stmp),&(bndp),(errp),(text *)(sqlvar), \
                          strlen((sqlvar)),0,(progvl),(ftype), \
                          indp,0,0,0,0,OCI_DATA_AT_EXEC)); \
       ocierror(__FILE__,__LINE__,(errp), \

OCIBindDynamic((bndp),(errp),(ctxp),(cbf_nodata),(ctxp),(cbf_data)));



#define OCIBNDR(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode)
\
       ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
       ocierror(__FILE__,__LINE__,(errp), \
          OCIBindByName((stmp),&(bndp),(errp),(text
*)(sqlvar),strlen((sqlvar)),\

(progv),(progvl),(ftype),(indp),(alen),(arcode),0,0,OCI_DEFAULT));

#define
OCIBNDRAA(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode,ms,cu)
\
       ocierror(__FILE__,__LINE__, (errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0));\
       ocierror(__FILE__,__LINE__,(errp),\
         OCIBindByName((stmp),&(bndp),(errp),(text
*)(sqlvar),strlen((sqlvar)),\

(progv),(progvl),(ftype),(indp),(alen),(arcode),(ms),(cu),OCI_DEFAULT));
```

```
#define OCIDEFINE(stmp,dfnp,errp,pos,progv,progvl,ftype)\

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),(progvl),(ftype),\
                     0,0,0,OCI_DEFAULT);


#define OCIDEF(stmp,dfnp,errp,pos,progv,progvl,ftype) \
       OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTYPE_DEFINE,0,\
                                  (dvoid**)0);\
       OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),(progvl),\
                           (ftype),NULL,NULL,NULL,OCI_DEFAULT); \


#define OCIDFNRA(stmp,dfnp,errp,pos,progv,progvl,ftype,indp,alen,arcode) \
       OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTYPE_DEFINE,0,\
                                  (dvoid**)0);\
       OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),\
                                  (progvl),(ftype),(indp),(alen),\
                                  (arcode),OCI_DEFAULT);

#define
OCIDFNDYN(stmp,dfnp,errp,pos,progv,progvl,ftype,indp,ctxp,cbf_data) \
       ocierror(__FILE__,__LINE__,(errp), \
       OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTYPE_DEFINE,0,\
                                  (dvoid**)0));\
       ocierror(__FILE__,__LINE__,(errp), \
       OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),
(progvl),(ftype),\
                                    (indp),NULL,NULL,
OCI_DYNAMIC_FETCH));\
       ocierror(__FILE__,__LINE__,(errp), \
       OCIDefineDynamic((dfnp),(errp),(ctxp),(cbf_data)));




// Min/Max transaction data definitions
#define MIN_DId 1
#define MAX_DId 10
#define MIN_OL 5
#define MAX_OL 15
#define MIN_QUANTITY 1
#define MAX_QUANTITY 10
#define MIN_ITEM_ID 1
#define MAX_ITEM_ID 100000
#define MIN_CUST_ID 1
#define MAX_CUST_ID 3000
#define MIN_CARRIER 1
#define MAX_CARRIER 10
#define MIN_THRESHOLD 10
#define MAX_THRESHOLD 20

#define STATUS_LEN 200
#define NAME_LEN 16
#define ADDR_LEN 20
#define STATE_LEN 2
#define ZIP_LEN 9

#pragma pack(1)
```

```
typedef struct
{
   int year;
   int quarter;
   int month;
   int dayofyear;
   int day;
   int week;
   int weekday;
   int hour;
   int minute;
   int second;
   int millisecond;
} DBDATEREC;

typedef struct
{
   short ol_supply_w_id;
   long ol_i_id;
   char ol_i_name[25];
   short ol_quantity;
   char ol_brand_generic[2];
   double ol_i_price;
   double ol_amount;
   short ol_stock;
} OL_NEW_ORDER_DATA;

typedef struct
{
   short w_id;
   short d_id;
   long        c_id;
   short o_ol_cnt;
   char c_last[NAME_LEN + 1];
   char c_credit[3];
   double c_discount;
   double w_tax;
   double d_tax;
   long o_id;
   short o_commit_flag;
   DBDATEREC o_entry_d;
   short o_all_local;
   double total_amount;
   char execution_status[STATUS_LEN];
   OL_NEW_ORDER_DATA Ol[MAX_OL];
} NEW_ORDER_DATA;

typedef struct
{
   short w_id;
   short d_id;
   long c_id;
   short c_d_id;
   short c_w_id;
   double h_amount;
   DBDATEREC h_date;
   char w_street_1[ADDR_LEN + 1];
   char w_street_2[ADDR_LEN + 1];
   char w_city[ADDR_LEN + 1];
   char w_state[STATE_LEN + 1];
   char w_zip[ZIP_LEN + 1];
```

```
   char d_street_1[ADDR_LEN + 1];
   char d_street_2[ADDR_LEN + 1];
   char d_city[ADDR_LEN + 1];
   char d_state[STATE_LEN + 1];
   char d_zip[ZIP_LEN + 1];
   char c_first[NAME_LEN + 1];
   char c_middle[3];
   char c_last[NAME_LEN + 1];
   char c_street_1[ADDR_LEN + 1];
   char c_street_2[ADDR_LEN + 1];
   char c_city[ADDR_LEN + 1];
   char c_state[STATE_LEN + 1];
   char c_zip[ZIP_LEN + 1];
   char c_phone[16];
   DBDATEREC c_since;
   char c_credit[3];
   double c_credit_lim;
   double c_discount;
   double c_balance;
   char c_data[200+1];
   char execution_status[STATUS_LEN];
} PAYMENT_DATA;

typedef struct
{
   short w_id;
   short o_carrier_id;
   long o_id[10];
   int iComplete;
   SYSTEMTIME QTime;                // time delivery was queued
   SYSTEMTIME EndTime;              // time delivery completed
   char execution_status[STATUS_LEN];
} DELIVERY_DATA;

typedef struct
{
   long ol_i_id;
   short ol_supply_w_id;
   short ol_quantity;
   double ol_amount;
   DBDATEREC ol_delivery_d;
} OL_ORDER_STATUS_DATA;

typedef struct
{
   short w_id;
   short d_id;
   long c_id;
   char c_first[NAME_LEN + 1];
   char c_middle[3];
   char c_last[NAME_LEN + 1];
   double c_balance;
   long o_id;
   DBDATEREC o_entry_d;
   short o_carrier_id;
   OL_ORDER_STATUS_DATA OlOrderStatusData[MAX_OL];
   short o_ol_cnt;
   char execution_status[STATUS_LEN];
} ORDER_STATUS_DATA;

typedef struct
{
   short w_id;
   short d_id;
   short thresh_hold;
   long low_stock;
   char execution_status[STATUS_LEN];
} STOCK_LEVEL_DATA;

#pragma pack()

#endif
```

tpccflags.h

```
// tpccflags.h
// Copyright (c) 1998 1999 Unisys Corporation
// Copyright (c) 1995  Oracle Corp, Redwood Shores, CA
#define SVRUNIX
#define PLSQLNO
#define DMLRETDEL
#define TUX
```

svrjack.c

```
// svrjack.c
// Copyright (c) 1998 1999  Unisys Corporation

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

#if defined(__cplusplus)
extern "C" {
#endif
extern int _tmrunserver _((int));
extern void NEWORDER _((TPSVCINFO *));
extern void PAYMENT _((TPSVCINFO *));
extern void DELIVERY _((TPSVCINFO *));
extern void ORDERSTS _((TPSVCINFO *));
extern void STOCKLVL _((TPSVCINFO *));
#if defined(__cplusplus)
}
#endif

static struct tmdsptchtbl_t _tmdsptchtbl[] = {
        { "NEWORDER", "NEWORDER", (void (*) _((TPSVCINFO *))) NEWORDER, 0,
0 },
        { "PAYMENT",  "PAYMENT",  (void (*) _((TPSVCINFO *))) PAYMENT,  1,
0 },
        { "DELIVERY", "DELIVERY", (void (*) _((TPSVCINFO *))) DELIVERY, 2,
0 },
        { "ORDERSTS", "ORDERSTS", (void (*) _((TPSVCINFO *))) ORDERSTS, 3,
0 },
        { "STOCKLVL", "STOCKLVL", (void (*) _((TPSVCINFO *))) STOCKLVL, 4,
0 },
        { NULL, NULL, NULL, 0, 0 }
};

#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()));
}
```

orasvr.c

```
// orasvr.c
// Copyright (c) 1998 1999  Unisys Corporation
// Copyright (c) 1995  Oracle Corp, Redwood Shores, CA

#include <stdio.h>
#ifdef ORA_NT
#include <windows.h>
#include <process.h>
#endif

#include "tpcc.h"

#include <atmi.h>
#include <userlog.h>

char    szDBServer[32] = "tpcc";
char    szUser[32]             = "tpcc";
char    szPassword[32] = "tpcc";
char  szService[16]  = "orasvr";
char  szWork[200];
FILE *fpLog;
char  szLogTitle[32];
int   iServerNo = 0;

void GetArgs(int argc, char **argv);

//============================================================================
//
// Function name: tpsvrinit
//
//============================================================================
tpsvrinit(int argc, char *argv[])
{
   int iRslt;
   GetArgs(argc,argv);
   iServerNo=_getpid();
   iRslt = TPCinit(iServerNo,szUser,szPassword,szDBServer);
   if (iRslt)
      sprintf(szWork,"%s Init Failed (%s,%s,%s)",
         szService,szUser,szPassword,szDBServer);
   else
      sprintf(szWork,"%s Init Complete (%s,%s,%s)",
         szService,szUser,szPassword,szDBServer);
   userlog(szWork);
   return(iRslt);
} // tpsvrinit

//============================================================================
//
// Function name: tpsvrdone
//
//============================================================================
void tpsvrdone ()
{
   userlog("Shutdown request for %s",szService);
   TPCexit (0);
}   // tpsvrdone

//============================================================================
//
// Function name: NEWORDER
//
//============================================================================
void NEWORDER(TPSVCINFO * msg)
{
   int iRslt;

   NEW_ORDER_DATA * newinfo;
   newinfo = (NEW_ORDER_DATA *) msg->data;
   iRslt = TPCnew(newinfo);
   if (iRslt == SVC_NOERROR)
      tpreturn(TPSUCCESS,0,msg->data,msg->len,0);
   else
      tpreturn(TPFAIL,iRslt,msg->data,msg->len,0);
```

```
}   // NEWORDER

//============================================================================
//
// Function name: PAYMENT
//
//============================================================================
void PAYMENT(TPSVCINFO * msg)
{
   int iRslt;
   PAYMENT_DATA * payinfo;
   payinfo = (PAYMENT_DATA *) msg->data;
   iRslt = TPCpay(payinfo);
   if (iRslt == SVC_NOERROR)
      tpreturn(TPSUCCESS,0,msg->data,msg->len,0);
   else
      tpreturn(TPFAIL,iRslt,msg->data,msg->len,0);
}   // PAYMENT

//============================================================================
//
// Function name: DELIVERY
//
//    Entry point called by tuxedo for DELIVERY service requests.
//
//============================================================================
void DELIVERY(TPSVCINFO * msg)
{
   int iRslt;
   DELIVERY_DATA * delinfo;

   delinfo = (DELIVERY_DATA *) msg->data;
   iRslt = TPCdel(delinfo);

   if (iRslt == SVC_NOERROR)
      tpreturn(TPSUCCESS,0,msg->data,msg->len,0);
   else
      tpreturn(TPFAIL,iRslt,msg->data,msg->len,0);
}   // DELIVERY

//============================================================================
//
// Function name: ORDERSTS
//
//============================================================================
void ORDERSTS(TPSVCINFO * msg)
{
   int iRslt;
   ORDER_STATUS_DATA * ordinfo;
   ordinfo = (ORDER_STATUS_DATA *) msg->data;
   iRslt = TPCord(ordinfo);
   if (iRslt == SVC_NOERROR)
      tpreturn(TPSUCCESS,0,msg->data,msg->len,0);
   else
      tpreturn(TPFAIL,iRslt,msg->data,msg->len,0);
}   // ORDERSTS

//============================================================================
//
// Function name: STOCKLVL
//
//============================================================================
void STOCKLVL(TPSVCINFO * msg)
{
   int iRslt;
   STOCK_LEVEL_DATA * stoinfo;
   stoinfo = (STOCK_LEVEL_DATA *) msg->data;
   iRslt = TPCsto(stoinfo);
   if (iRslt == SVC_NOERROR)
      tpreturn(TPSUCCESS,0,msg->data,msg->len,0);
   else
      tpreturn(TPFAIL,iRslt,msg->data,msg->len,0);
}   // STOCKLVL

//============================================================================
//
// Function name: GetArgs
//
//============================================================================
void GetArgs(int argc, char **argv)
{
   int j;
   char * ptr;
   BOOL bRslt = TRUE;

   for (j = 1; j < argc; ++j)
   {
      ptr = argv[j];
      switch (ptr[1])
      {
         case 's':
         case 'S':
            strcpy(szDBServer,ptr+2);
            break;

         case 'u':
         case 'U':
            strcpy(szUser,ptr+2);
            break;

         case 'p':
         case 'P':
            strcpy(szPassword,ptr+2);
            break;

      };   // switch(ptr[1])
   };   // for (j = 1; j < argc; ++j)
}   // GetArgs
```

tpccpl.c

```
// tpccpl.c
// Copyright (c) 1998 1999  Unisys Corporation
// Copyright (c) 1995  Oracle Corp, Redwood Shores, CA

#include <stdio.h>
#include <time.h>
#include "tpcc.h"
#include <userlog.h>

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

```
#ifdef SQL_TRACE
#define SQLTXT1 "alter session set sql_trace = true"
#endif

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 res_init = 0;

int execstatus;
int errcode;

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


/* for stock-level transaction */

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

/* for delivery transaction */
int retries;
int w_id;
int o_carrier_id;
int del_o_id[10];
OCIDate cr_date;

/* for order-status transaction */

int bylastname;
char c_last[17];
char c_first[17];
char c_middle[3];
double c_balance;
int o_id;
text o_entry_d[20];
ub4  datelen;
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];
ub4  ol_del_len[15];
OCIDate ol_delivery_d[15];

/* 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];
ub4  sincelen;
text c_since_d[11];
float c_discount;
char c_credit[3];
int c_credit_lim;
char c_data[201];
ub4 hlen;
text h_date[20];

/* for new order transaction */

int nol_i_id[15];
int nol_supply_w_id[15];
int nol_quantity[15];
int nol_quanti10[15];
int nol_quanti91[15];
int nol_ytdqty[15];
int nol_amount[15];
int o_all_local;
#ifdef SVRUNIX
float w_tax;
float d_tax;
#else
int w_tax;
int d_tax;
#endif
float total_amount;
char i_name[15][25];
int s_quantity[15];
//char brand_gen[15];
int i_price[15];
char brand_generic[15][1];
int status;

OCIDate cr_date;
OCIDate c_since;
OCIDate o_entry_d_base;
OCIDate ol_d_base[15];
dvoid *xmem;

/* NewOrder Binding stuff */
```

```
extern void cvtdmyhms(OCIDate * oradt,DBDATEREC * pdd);

void cvtdmyhms(OCIDate * oradt,DBDATEREC * pdd)
{
   OCIDateGetTime(oradt,&pdd->hour,&pdd->minute,&pdd->second);
   OCIDateGetDate(oradt,&pdd->year,&pdd->month,&pdd->day)
        return;
}   // cvtdmyhms

int ocierror(char * fname,int lineno,
                       OCIError * errhp,sword status)
{
  text errbuf[512];
  sb4 errcode;
  sb4 lstat;
  ub4 recno=2;

  switch (status) {
  case OCI_SUCCESS:
    break;
  case OCI_SUCCESS_WITH_INFO:
    userlog("Module %s Line %d\n", fname, lineno);
    userlog("Error - OCI_SUCCESS_WITH_INFO\n");
    lstat = OCIErrorGet (errhp, recno++, (text *) NULL, &errcode, errbuf,
                         (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
    userlog("Error - %s\n", errbuf);
    break;
  case OCI_NEED_DATA:
    userlog("Module %s Line %d\n", fname, lineno);
    userlog("Error - OCI_NEED_DATA\n");
    return (IRRECERR);
  case OCI_NO_DATA:
//    userlog("Module %s Line %d\n", fname, lineno);
//    userlog("Error - OCI_NO_DATA\n");
    return (OCI_NO_DATA);
  case OCI_ERROR:
    lstat = OCIErrorGet (errhp, (ub4) 1,
                  (text *) NULL, &errcode, errbuf,
                      (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
    if (errcode == NOT_SERIALIZABLE) return (errcode);
    if (errcode == SNAPSHOT_TOO_OLD) return (errcode);
     while (lstat != OCI_NO_DATA)
    {
      userlog("Module %s Line %d\n", fname, lineno);
      userlog("Error - %s\n", errbuf);
      lstat = OCIErrorGet (errhp, recno++, (text *) NULL, &errcode,
errbuf,
                         (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
    }
    return (errcode);
  case OCI_INVALID_HANDLE:
    userlog("Module %s Line %d\n", fname, lineno);
    userlog("Error - OCI_INVALID_HANDLE\n");
    TPCexit();
    exit(-1);
  case OCI_STILL_EXECUTING:
    userlog("Module %s Line %d\n", fname, lineno);
    userlog("Error - OCI_STILL_EXECUTE\n");
    return (IRRECERR);
  case OCI_CONTINUE:
    userlog("Module %s Line %d\n", fname, lineno);
    userlog("Error - OCI_CONTINUE\n");
    return (IRRECERR);
  default:
    userlog("Module %s Line %d\n", fname, lineno);
    userlog("Status - %s\n", status);
    return (IRRECERR);
  }
  return (RECOVERR);
}   // ocierror

FILE  *vopen(char * fnam,char * mode)
{
FILE *fd;

#ifdef  DEBUG
  fprintf(stderr, "tkvuopen() fnam: %s,   mode: %s\n", fnam, mode);
#endif

    fd = fopen((char *)fnam,(char *)mode);
    if (!fd){
        userlog(" fopen on %s failed %d\n",fnam,fd);
        exit(-1);
    }
    return(fd);
}  // vopen

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

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

    fd = vopen(fnam,"r");
    while (fgets((char *)linebuf+nulpt, SQL_BUF_SIZE,fd))
    {
        nulpt = strlen((char *)linebuf);
    }
    return(nulpt);
}   // sqlfile

TPCexit()
{

   if (new_init) {
      tkvcndone();
      new_init = 0;
   }
   if (pay_init) {
      tkvcpdone();
      pay_init = 0;
   }
   if (del_init) {
      tkvcddone();
      del_init = 0;
   }
   if (ord_init) {
      tkvcodone();
      ord_init = 0;
```

```
   }
   if (sto_init) {
      tkvcsdone();
      sto_init = 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);
   return(0);
}   // TPCexit

int TPCinit(int id,char * uid,char * pwd,char * dbserver)
{
   text stmbuf[100];
   proc_no = id;
   OCIInitialize(OCI_DEFAULT|OCI_OBJECT,(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);
   OCIERROR(errhp,OCIServerAttach(tpcsrv, errhp, (text
*)(dbserver),strlen(dbserver),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 *)uid,
(ub4)strlen(uid),OCI_ATTR_USERNAME, errhp);
   OCIAttrSet((dvoid *)tpcusr, OCI_HTYPE_SESSION, (dvoid *)pwd,
(ub4)strlen(pwd),
             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);

   /* run all transaction in serializable mode */

   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);
   if (ocierror(__FILE__,__LINE__,errhp,OCIStmtExecute(tpcsvc, curi,
errhp,1,0,0,0,OCI_DEFAULT)) != RECOVERR)
   {
      OCIHandleFree(curi, OCI_HTYPE_STMT);
      TPCexit();
      return (-1);
   };
   OCIHandleFree(curi, OCI_HTYPE_STMT);

   logon = 1;

   OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));

   if (tkvcninit ()) {  /* new order */
      TPCexit ();
      return (-1);
   }
   else
      new_init = 1;

   if (tkvcpinit ()) {   /* payment */
      TPCexit ();
      return (-1);
   }
   else
      pay_init = 1;

   if (tkvcdinit ())  {   /* delivery */
      TPCexit ();
      return (-1);
   }
   else
      del_init = 1;

   if (tkvcoinit ()) {   /* order status */
      TPCexit ();
      return (-1);
   }
   else
      ord_init = 1;

   if (tkvcsinit ()) {   /* stock level */
      TPCexit ();
      return (-1);
   }
   else
      sto_init = 1;

   return (0);

}   // TPCinit

int TPCnew (NEW_ORDER_DATA * pnod)
{
   int i;
   int terror;
   w_id = pnod->w_id;
   d_id = pnod->d_id;
   c_id = pnod->c_id;
   for (i = 0; i < 15; i++)
   {
      nol_i_id[i] = pnod->Ol[i].ol_i_id;
      nol_supply_w_id[i] = pnod->Ol[i].ol_supply_w_id;
      nol_quantity[i] = pnod->Ol[i].ol_quantity;
   }
   retries = 0;
   OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));
   terror = tkvcn();
   if (terror < SVC_NOERROR)
   {
      if (terror == SVCERR_NOCUSTOMER)
         strcpy(pnod->execution_status,"Invalid Customer Id");
      else
         strcpy(pnod->execution_status,"Irrecov See ULOG");
```

```
      return(terror);
   };
   cvtdmyhms(&cr_date,&pnod->o_entry_d);
   pnod->o_id = o_id;
   pnod->o_ol_cnt = o_ol_cnt;
   strncpy(pnod->c_last,c_last,17);
   strncpy(pnod->c_credit,c_credit,3);
   pnod->c_discount = c_discount;
#ifdef SVRUNIX
       pnod->w_tax = w_tax;
   pnod->d_tax = d_tax;
#else
   pnod->w_tax = (float)(w_tax)/10000;
   pnod->d_tax = (float)(d_tax)/10000;
#endif
       pnod->total_amount = total_amount;
   for (i = 0; i < o_ol_cnt; i++)
   {
      strncpy(pnod->Ol[i].ol_i_name,i_name[i],25);
      pnod->Ol[i].ol_stock = s_quantity[i];
      pnod->Ol[i].ol_brand_generic[0] = brand_generic[i][0];
      pnod->Ol[i].ol_brand_generic[1] = '\0';
      pnod->Ol[i].ol_i_price = (float)(i_price[i])/100;
      pnod->Ol[i].ol_amount = (float)(nol_amount[i])/100;
   }
   if (terror == SVC_BADITEMID)
      strcpy(pnod->execution_status,"Item number is not valid");
   else
      strcpy(pnod->execution_status,"Transaction commited.");
   return(terror);
}   // TPCnew



int TPCpay (PAYMENT_DATA * ppd)
{
   int terror;
   w_id = ppd->w_id;
   d_id = ppd->d_id;
   c_w_id = ppd->c_w_id;
   c_d_id = ppd->c_d_id;
   h_amount = (int) (ppd->h_amount * 100);
   OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));
   if (ppd->c_id == 0)
   {
      c_id = 0;
      strncpy(c_last,ppd->c_last,17);
      bylastname = TRUE;
   }
   else
   {
      c_id = ppd->c_id;
      strcpy(c_last," ");
      bylastname = FALSE;
   };
   retries = 0;
   terror = tkvcp();
   if (terror < SVC_NOERROR)
   {
      if (terror == SVCERR_NOCUSTOMER)
         strcpy(ppd->execution_status,"Invalid Customer Name or Number");
      else
         strcpy(ppd->execution_status,"Irrecov See ULOG");
      return(terror);
   };
   cvtdmyhms(&cr_date,&ppd->h_date);
   cvtdmyhms(&c_since,&ppd->c_since);
   strncpy (ppd->w_street_1,w_street_1,21);
   strncpy (ppd->w_street_2,w_street_2,21);
   strncpy (ppd->w_city,w_city,21);
   strncpy (ppd->w_state,w_state,3);
   strncpy (ppd->w_zip,w_zip,10);
   strncpy (ppd->d_street_1,d_street_1,21);
   strncpy (ppd->d_street_2,d_street_2,21);
   strncpy (ppd->d_city,d_city,21);
   strncpy (ppd->d_state,d_state,3);
   strncpy (ppd->d_zip,d_zip,10);
   ppd->c_id = c_id;
   strncpy (ppd->c_first,c_first,17);
   strncpy (ppd->c_middle,c_middle,3);
   strncpy (ppd->c_last,c_last,17);
   strncpy (ppd->c_street_1,c_street_1,21);
   strncpy (ppd->c_street_2,c_street_2,21);
   strncpy (ppd->c_city,c_city,21);
   strncpy (ppd->c_state,c_state,3);
   strncpy (ppd->c_zip,c_zip,10);
   strncpy (ppd->c_phone,c_phone,17);
   strncpy (ppd->c_credit,c_credit,3);
   ppd->c_credit_lim = (float)(c_credit_lim)/100;
   ppd->c_discount = c_discount;
   ppd->c_balance = (float)(c_balance)/100;
   strncpy(ppd->c_data,c_data,201);
   strcpy(ppd->execution_status,"Transaction commited.");
   return(terror);
}   // TPCpay



int TPCdel(DELIVERY_DATA * pdd)
{
   int i;
   int terror;
   w_id = pdd->w_id;
   o_carrier_id = pdd->o_carrier_id;
   pdd->iComplete = 0;
   OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));
   retries = 0;
   terror = tkvcd();
   if (terror)
   {
      if (terror == DEL_ERROR)
         return DEL_ERROR;
      if (terror != RECOVERR)
            terror = IRRECERR;
      strcpy(pdd->execution_status,"Irrecov");
      return (-1);
   };
   for (i = 0; i < NDISTS; i++)
      pdd->o_id[i] = del_o_id[i];
   strcpy(pdd->execution_status,"Transaction commited.");
   return (0);
}   // TPCdel
```

```
int TPCord (ORDER_STATUS_DATA * posd)
{
   int i;
   int terror;
   w_id = posd->w_id;
   d_id = posd->d_id;
   if (posd->c_id == 0)
   {
      c_id = 0;
      strncpy(c_last,posd->c_last,17);
      bylastname = TRUE;
   }
   else
   {
      c_id = posd->c_id;
      strcpy(c_last," ");
      bylastname = FALSE;
   };
   retries = 0;
   terror = tkvco();
   if (terror < SVC_NOERROR)
   {
      if (terror == SVCERR_NOCUSTOMER)
         strcpy(posd->execution_status,"Invalid Customer Name or Number");
      else
         strcpy(posd->execution_status,"Irrecov See ULOG");
      return(terror);
   };
   cvtdmyhms(&o_entry_d_base,&posd->o_entry_d);
   posd->c_id = c_id;
   strncpy(posd->c_last,c_last,17);
   strncpy(posd->c_first,c_first,17);
   strncpy(posd->c_middle,c_middle,3);
   posd->c_balance = c_balance/100;
   posd->o_id = o_id;
   if (o_carrier_id == 11)
      posd->o_carrier_id = 0;
   else
      posd->o_carrier_id = o_carrier_id;
   posd->o_ol_cnt = o_ol_cnt;
   for (i = 0; i < o_ol_cnt; i++)
   {
      cvtdmyhms(&ol_delivery_d[i],&posd-
>OlOrderStatusData[i].ol_delivery_d);
      posd->OlOrderStatusData[i].ol_supply_w_id = ol_supply_w_id[i];
      posd->OlOrderStatusData[i].ol_i_id = ol_i_id[i];
      posd->OlOrderStatusData[i].ol_quantity = ol_quantity[i];
      posd->OlOrderStatusData[i].ol_amount = (float)(ol_amount[i])/100;
   }
   return(terror);
}   // TPCord




int TPCsto (STOCK_LEVEL_DATA * psld)
{
   int terror;
   w_id = psld->w_id;
   d_id = psld->d_id;
   threshold = psld->thresh_hold;
   retries = 0;
   terror = tkvcs();
   if (terror < SVC_NOERROR)
   {
      strcpy(psld->execution_status,"Irrecov See ULOG");
         return(terror);
   };
   psld->low_stock = low_stock;
   return(terror);
}   // TPCsto
```

plnew.c

```
// plnew.c
// Copyright (c) 1998 1999 Unisys Corporation
// Copyright (c) 1995  Oracle Corp, Redwood Shores, CA

#include "tpcc.h"
#ifdef TUX
#include <userlog.h>
#endif

#ifndef PLSQLNO
int SelItemStk();
int UpdStk2();
#endif

#ifdef PLSQLNO
#define SQLTXT2 "BEGIN initnew.new_init(:idx1arr); END;"
#else
#define SQLTXT2 "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 \
   WHERE rowid = :s_rowid"

#define SQLTXT3_1 "\
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 \
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 \
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 \
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 \
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 \
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 SQLTXT3_2 "\
```

```
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 \
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 \
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 \
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 \
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 SQLTXT3_3 "\
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 \
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 \
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 \
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 SQLTXT4 "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)\
   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)"
#endif  /* PLSQLNO */

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

void shiftitemstock (int i, int j);

sb4 no_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
               dvoid **bufpp, ub4 *alenp, ub1 *piecep,
               dvoid **indpp)
{
  *bufpp = (dvoid*)0;
  *alenp =0;
  *indpp = (dvoid*)0;
  *piecep =OCI_ONE_PIECE;
  return (OCI_CONTINUE);
}
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];
#ifdef PLSQLNO
  sb2 s_bg_ind[NITEMS];
#endif

  ub2 nol_i_id_len[NITEMS];
  ub2 nol_supply_w_id_len[NITEMS];
  ub2 nol_quantity_len[NITEMS];
  ub2 nol_amount_len[NITEMS];
  ub2 s_quantity_len[NITEMS];
  ub2 i_name_len[NITEMS];
  ub2 i_price_len[NITEMS];
  ub2 i_data_len[NITEMS];
  ub2 s_dist_info_len[NITEMS];
  ub2 s_data_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 ol_dist_info_len[NITEMS];
  ub2 null_date_len[NITEMS];
#ifdef PLSQLNO
  ub2 s_bg_len[NITEMS];
#endif

  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];
#ifdef PLSQLNO
  ub2 s_bg_rcode[NITEMS];
#endif

  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];
  OCIDate null_date[NITEMS]; /* base date for null date entry */
  OCIStmt *curn1;
#ifdef PLSQLNO
  OCIBind *ol_i_id_bp;
  OCIBind *ol_supply_w_id_bp;
  OCIBind *i_price_bp;
  OCIBind *i_name_bp;
  OCIBind *s_bg_bp;
  OCIBind *s_data_bp;
  OCIBind *i_data_bp;
  ub4 nol_i_count;
  ub4 nol_s_count;
  ub4 nol_q_count;
  ub4 nol_item_count;
  ub4 nol_name_count;
  ub4 nol_qty_count;
  ub4 nol_bg_count;
  ub4 nol_am_count;
  ub4 s_remote_count;
  ub4 s_data_count;
  ub4 i_data_count;
#endif
  OCIStmt *curn2;
  OCIStmt *curn3[10];
  OCIBind *ol_i_id_bp4;
  OCIBind *ol_supply_w_id_bp4;
  OCIBind *ol_quantity_bp;
  OCIBind *ol_quantity_bp4;
  OCIBind *s_remote_bp;
  OCIBind *s_quantity_bp;
  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 *s_rowid_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;

/* context holders */
  int i_name_ctx;
  int i_data_ctx;
  int i_price_ctx;
  int s_data_ctx;
  int s_dist_info_ctx;
  int s_quantity_ctx;
};

typedef struct newctx newctx;

newctx *nctx;

tkvcninit ()
{
  int i;
  text stmbuf[16*1024];

#ifndef PLSQLNO
  char sd[4];
  char id[4];
  int j;
  char sqltxt[SQL_BUF_SIZE];
# endif /* !PLSQLNO */

  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));
#ifdef PLSQLNO
//  sqlfile("\\tuxedo\\runtime\\sql816\\tkvcpnew.sql",stmbuf);
  sqlfile("/home/tuxedo/runtime/sql816/tkvcpnew.sql",stmbuf);
#else
//  sqlfile("\\tuxedo\\runtime\\sql816\\tkvcbnew.sql",stmbuf);
  sqlfile("/home/tuxedo/runtime/sql816/tkvcbnew.sql",stmbuf);
#endif
  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);
#ifdef SVRUNIX
    OCIBNDR(nctx->curn1, nctx->w_tax_bp, errhp,
":w_tax",ADR(w_tax),SIZ(w_tax),
          SQLT_FLT, &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_FLT, &nctx->d_tax_ind, &nctx->d_tax_len, &nctx->d_tax_rc);
#else
  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);
```

```
#endif
  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_FLT,
       &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(OCIDate),
 SQLT_ODT, &nctx->cr_date_ind, &nctx->cr_date_len, &nctx->cr_date_rc);

#ifdef PLSQLNO
  OCIBNDRAA(nctx->curn1, 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,NITEMS,&nctx->nol_i_count);
  OCIBNDRAA(nctx->curn1, 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,
            NITEMS, &nctx->nol_s_count);
  OCIBNDRAA(nctx->curn1, 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,NITEMS,&nctx->nol_q_count);
  OCIBNDRAA(nctx->curn1, nctx-
>i_price_bp,errhp,":i_price",i_price,SIZ(int),
             SQLT_INT, nctx->i_price_ind,nctx->i_price_len,nctx-
>i_price_rcode,
             NITEMS, &nctx->nol_item_count);
  OCIBNDRAA(nctx->curn1, nctx->i_name_bp,errhp,":i_name",i_name,
             SIZ(i_name[0]),SQLT_STR, nctx->i_name_ind,nctx->i_name_len,
             nctx->i_name_rcode,NITEMS,&nctx->nol_name_count);
  OCIBNDRAA(nctx->curn1, 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,NITEMS,&nctx->nol_qty_count);
  OCIBNDRAA(nctx->curn1, nctx-
>s_bg_bp,errhp,":brand_generic",brand_generic,
            SIZ(char), SQLT_CHR,nctx->s_bg_ind,nctx->s_bg_len,
            nctx->s_bg_rcode,NITEMS,&nctx->nol_bg_count);
  OCIBNDRAA(nctx->curn1, 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,NITEMS,&nctx->nol_am_count);
  OCIBNDRAA(nctx->curn1, 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,NITEMS,&nctx->s_remote_count);

  /* open second cursor */
  OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)(&nctx->curn2),
OCI_HTYPE_STMT,
                                0, (dvoid**)0));
  sprintf ((char *) stmbuf, SQLTXT2);
  OCIERROR(errhp,OCIStmtPrepare(nctx->curn2, errhp, stmbuf,
           strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));


  /* execute second cursor to init newinit package */
  {
    int idx1arr[NITEMS];
    OCIBind *idx1arr_bp;
    ub2 idx1arr_len[NITEMS];
    ub2 idx1arr_rcode[NITEMS];
    sb2 idx1arr_ind[NITEMS];
    ub4 idx1arr_count;
    ub2 idx;

    for (idx = 0; idx < NITEMS; idx++) {
      idx1arr[idx] = idx + 1;
      idx1arr_ind[idx] = TRUE;
      idx1arr_len[idx] = sizeof(int);
    }
    idx1arr_count = NITEMS;
       o_ol_cnt = NITEMS;


    /* Bind array */
    OCIBNDRAA(nctx->curn2, idx1arr_bp,errhp,":idx1arr",idx1arr,
              SIZ(int), SQLT_INT, idx1arr_ind,idx1arr_len,
              idx1arr_rcode,NITEMS,&idx1arr_count);

     execstatus = OCIStmtExecute(tpcsvc,nctx-
>curn2,errhp,1,0,0,0,OCI_DEFAULT);
     if(execstatus != OCI_SUCCESS) {
       OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
       errcode = OCIERROR(errhp,execstatus);
       return -1;
     }
   }
#else

  /* open second cursor */
  OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)(&nctx->curn2),
OCI_HTYPE_STMT,
                                0, (dvoid**)0));
  sprintf ((char *) stmbuf, SQLTXT2);
  OCIERROR(errhp,OCIStmtPrepare(nctx->curn2, errhp, stmbuf,
           strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));



  /* bind variables */

  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);
```

```
  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));

    strcpy(sqltxt,SQLTXT3_1);
    strcat(sqltxt,SQLTXT3_2);
    strcat(sqltxt,SQLTXT3_3);
    sprintf ((char *) stmbuf, sqltxt, 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);
    }

    OCIDEF((nctx->curn3)[i],(nctx->Dcons)[i],errhp,1,&(nctx->cons[0]),
             SIZ(nctx->cons[0]),SQLT_INT);
    OCIDEF((nctx->curn3)[i], (nctx->Ds_rowid)[i],errhp,2,
             nctx->s_rowid_ptr, sizeof(nctx->s_rowid_ptr[0]), SQLT_RDD);
    OCIDEF((nctx->curn3)[i], (nctx->Di_price)[i],errhp,3,i_price,SIZ(int),
             SQLT_INT);

    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, NULL,nctx->i_data_len,NULL);
    OCIDFNRA((nctx->curn3)[i], (nctx->Ds_dist_info)[i],errhp,6,
              nctx->s_dist_info, SIZ(nctx->s_dist_info[0]),SQLT_STR,
              NULL,nctx->s_dist_info_len, NULL);
    OCIDFNRA((nctx->curn3)[i],(nctx->Ds_data)[i],errhp,7,nctx->s_data,
             SIZ(nctx->s_data[0]),SQLT_STR,NULL,nctx->s_data_len,NULL);
    OCIDEF((nctx->curn3)[i],(nctx->Ds_quantity)[i],errhp,8,s_quantity,
             SIZ(int),SQLT_INT);

   }

  /* open fourth cursor */
  OCIHandleAlloc(tpcenv, (dvoid **)(&nctx->curn4), OCI_HTYPE_STMT, 0,
                 (dvoid**)0);
  sprintf ((char *) stmbuf, SQLTXT4);
  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, NULL,nctx->ol_o_id_len,
           NULL);
  OCIBNDRA(nctx->curn4, nctx->ol_d_id_bp,errhp,":ol_d_id",nctx->ol_d_id,
           SIZ(int),SQLT_INT, NULL,nctx->ol_d_id_len,
           NULL);
  OCIBNDRA(nctx->curn4, nctx->ol_w_id_bp,errhp,":ol_w_id",nctx->ol_w_id,
           SIZ(int),SQLT_INT, NULL,nctx->ol_w_id_len,
           NULL);
  OCIBNDRA(nctx->curn4, nctx->ol_number_bp,errhp,":ol_number",nctx-
>ol_number,
           SIZ(int),SQLT_INT, NULL,nctx->ol_number_len,
           NULL);
  OCIBNDRA(nctx->curn4, nctx-
>ol_i_id_bp4,errhp,":ol_i_id",nol_i_id,SIZ(int),
           SQLT_INT, NULL,nctx->nol_i_id_len, NULL);
  OCIBNDRA(nctx->curn4, nctx->ol_supply_w_id_bp4,errhp,":ol_supply_w_id",
           nol_supply_w_id,SIZ(int),SQLT_INT, NULL,
           nctx->nol_supply_w_id_len, NULL);
  OCIBNDRA(nctx->curn4, nctx-
>ol_quantity_bp4,errhp,":ol_quantity",nol_quantity,
           SIZ(int),SQLT_INT, NULL,nctx->nol_quantity_len,
          NULL);
  OCIBNDRA(nctx->curn4, nctx->ol_amount_bp,errhp,":ol_amount",nol_amount,
           SIZ(int),SQLT_INT, NULL,nctx->nol_amount_len,
           NULL);
  OCIBNDRA(nctx->curn4, nctx->ol_dist_info_bp,errhp,":ol_dist_info",
           nctx->s_dist_info, SIZ(nctx->s_dist_info[0]),SQLT_AFC,
           NULL, nctx->ol_dist_info_len,
           NULL);
  OCIBNDRA(nctx->curn4, nctx->null_date_bp,errhp,":null_date",nctx-
>null_date,
           SIZ(OCIDate), SQLT_ODT,NULL,
           nctx->null_date_len, NULL);

  /* set up the null date Null date is 01-Jan-1811 */
  for (i=0;i<NITEMS;i++)
  {
     OCIDateSetDate(&nctx->null_date[i],(sb2)1811,(ub1)1,(ub1)1);
  }
```

```
#endif
  return (0);

}  // tkvcninit



tkvcn ()
{

   int i;
   int rcount;
# ifndef PLSQLNO
   ub4 flags;
   int rpc,rpc3;
# endif /* !PLSQLNO */
   int failed = 0;

retry:

   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);
#ifdef PLSQLNO
   /* this is the row count */
   rcount = o_ol_cnt;
   nctx->nol_i_count = o_ol_cnt;
   nctx->nol_q_count = o_ol_cnt;
   nctx->nol_s_count = o_ol_cnt;
   nctx->s_remote_count = o_ol_cnt;

   nctx->nol_qty_count  = 0;
   nctx->nol_bg_count = 0;
   nctx->nol_item_count = 0;
   nctx->nol_name_count = 0;
   nctx->nol_am_count   = 0;
   /* following not relevant */
   nctx->s_data_count = o_ol_cnt;
   nctx->i_data_count = o_ol_cnt;

   /* 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] = 0;
      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_data_ind[i] = TRUE;
      nctx->i_data_ind[i] = TRUE;
      nctx->s_quant_ind[i] = TRUE;
      nctx->s_bg_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(OCIDate);
      nctx->s_remote_len[i] = sizeof(int);
      nctx->s_data_len[i] = sizeof(int);
      nctx->i_data_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);
      nctx->i_name_len[i]=0;
      nctx->s_bg_len[i] = 0;
```

```
   }
   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_data_ind[i] = NA;
      nctx->i_data_ind[i] = NA;
      nctx->s_quant_ind[i] = NA;
      nctx->s_bg_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->i_data_len[i] = 0;
      nctx->s_data_len[i] = 0;
      nctx->s_quant_len[i] = 0;
      nctx->s_rowid_len[i] = 0;
      nctx->cons_len[i] = 0;
      nctx->i_name_len[i]=0;
      nctx->s_bg_len[i] = 0;
   }

   execstatus = OCIStmtExecute(tpcsvc,nctx->curn1,errhp,1,0,0,0,
                               OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);

#else
   execstatus = OCIStmtExecute(tpcsvc,nctx-
>curn1,errhp,1,0,0,0,OCI_DEFAULT);
#endif


   if(execstatus != OCI_SUCCESS)
   {
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      errcode = OCIERROR(errhp,execstatus);
      if (errcode == OCI_NO_DATA)
         return(SVCERR_NOCUSTOMER);
      if(errcode == NOT_SERIALIZABLE)
      {
         retries++;
         goto retry;
      }
      else
      if (errcode == RECOVERR)
      {
         retries++;
         goto retry;
      }
               else
               if (errcode == SNAPSHOT_TOO_OLD)
               {
                        retries++;
                        goto retry;
               }
      else
      {
         return(SVCERR_OCI);
      }
   };

#ifdef PLSQLNO
   /* did the txn succeed ? */
   if (rcount != o_ol_cnt)
   {
     status = rcount - o_ol_cnt;
     o_ol_cnt = rcount;
   }
#endif

#ifndef PLSQLNO
   /* 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(OCIDate);
      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;
   }

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

   /* 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) {
         s_quantity[i] -= nol_quantity[i];
         if (s_quantity[i] < 10)
            s_quantity[i] += 91;
         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_generic[i][0] = 'B';
         else
            brand_generic[i][0] = 'G';
      }

    }
#ifdef SVRUNIX
      total_amount *= (float)(1 - c_discount) * (float)(1.0 + d_tax +
w_tax);
#else
   total_amount *= ((float)(1 - c_discount)) * (1.0 +
((float)(d_tax)/10000) + ((float)(w_tax)/10000));
#endif
   total_amount = total_amount/100;

   rpc = UpdStk2 ();
   if (rpc == -2)
      goto retry;
   else if (rpc == -1)
      return(SVCERR_OCI);

   /* error processing - will keep it separated for readablity */
   /* number of items selected != number of stock updated */

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


/* common code for insert into order_line */
   for (i=0; i< o_ol_cnt; i++)  /* move district info in place */
   {
     nctx->ol_dist_info_len[i]=nctx->s_dist_info_len[i];
   }

   /* array insert into order line table */
   flags= (status ? OCI_DEFAULT : (OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS));
   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 || errcode ==
RECOVERR
                               || errcode == SNAPSHOT_TOO_OLD)
                       {
                               retries++;
                               goto retry;
                       }
                       else
                       {
                               return(SVCERR_OCI);
                       }
               }

      OCIAttrGet(nctx->curn4,OCI_HTYPE_STMT,&rcount,NULL,
OCI_ATTR_ROW_COUNT, errhp);
```

```
      if (rcount != (o_ol_cnt - status))
      {
                        userlog ("Error in TPC-C server %d: array insert
failed\n", proc_no);
                        /* rollback */
                        OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
                        return(SVCERR_OCI);
                }
   }

   /* commit if no invalid item */

   if (status) {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
   }

#endif
   total_amount = 0.0;
   for (i = 0; i < o_ol_cnt; i++)
   {
      if (nctx->nol_amount_ind[i] != NA) {
         total_amount += nol_amount[i];
      }

    }
#ifdef SVRUNIX
   total_amount *= (float)(1 - c_discount) * (float)(1.0 + d_tax + w_tax);
#else
   total_amount *= ((float)(1 - c_discount)) * (float)(1.0 +
((float)(d_tax)/10000) + ((float)(w_tax)/10000));
#endif
   total_amount = total_amount/100;

   if (status)
      return(SVC_BADITEMID);
   else
      return(SVC_NOERROR);

}   // tkvcn



void tkvcndone ()

{

   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 */

void 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];
}

void swapitemstock (i, j)

int i, j;

{

   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);

}



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 if (errcode == SNAPSHOT_TOO_OLD) {
        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)
```

```
    {
       userlog ("TPC-C server %d: more than 1 invalid item?\n", proc_no);
      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 */
    }


             userlog ("TPC-C server %d: reordering items and stocks\n",
                      proc_no);

    /* 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);

}



UpdStk2 ()
{

    int rpc, 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 if (errcode == SNAPSHOT_TOO_OLD) {
         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)) {
       userlog ("Error in TPC-C server %d: array update failed\n",
                proc_no);
       OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
       return (-1);
    }

    return (rpc);

}
```

plpay.c

```
// plpay.c
// Copyright (c) 1998 1999 Unisys Corporation
// Copyright (c) 1995  Oracle Corp, Redwood Shores, CA

#include "tpcc.h"

#define SQLTXT_INIT "BEGIN initpay.pay_init; END;"
#define SQLTXT_STP  "begin payment.dopayment(:w_id,:d_id,:c_w_id,:c_d_id,
\

:c_id,:by_lname,:h_amount,:c_last,:w_street_1,:w_street_2,\

:w_city,:w_state,:w_zip,:d_street_1,:d_street_2,:d_city,  \
                    :d_state,:d_zip,:c_first,:c_middle,:c_street_1,
\
                    :c_street_2,:c_city,:c_state,:c_zip,:c_phone,:c_since,
\

:c_credit,:c_credit_lim,:c_discount,:c_balance,:c_data,   \
                    :cr_date,:retry); end;"
```

```
struct payctx {
  OCIStmt *curpi;
  OCIStmt *curp0;
  OCIStmt *curp1;
  OCIBind *w_id_bp;
  OCIBind *w_id_bp1;
  sb2 w_id_ind;
  ub2 w_id_len;
  ub2 w_id_rc;

  OCIBind *d_id_bp;
  OCIBind *d_id_bp1;
  sb2 d_id_ind;
  ub2 d_id_len;
  ub2 d_id_rc;

  OCIBind *c_w_id_bp;
  OCIBind *c_w_id_bp1;
  sb2 c_w_id_ind;
  ub2 c_w_id_len;
  ub2 c_w_id_rc;

  OCIBind *c_d_id_bp;
  OCIBind *c_d_id_bp1;
  sb2 c_d_id_ind;
  ub2 c_d_id_len;
  ub2 c_d_id_rc;

  OCIBind *c_id_bp;
  OCIBind *c_id_bp1;
  sb2 c_id_ind;
  ub2 c_id_len;
  ub2 c_id_rc;

  OCIBind *by_lname_bp;

  OCIBind *h_amount_bp;
  OCIBind *h_amount_bp1;
  sb2 h_amount_ind;
  ub2 h_amount_len;
  ub2 h_amount_rc;

  OCIBind *c_last_bp;
  OCIBind *c_last_bp1;
  sb2 c_last_ind;
  ub2 c_last_len;
  ub2 c_last_rc;

  OCIBind *w_street_1_bp;
  OCIBind *w_street_1_bp1;
  sb2 w_street_1_ind;
  ub2 w_street_1_len;
  ub2 w_street_1_rc;

  OCIBind *w_street_2_bp;
  OCIBind *w_street_2_bp1;
  sb2 w_street_2_ind;
  ub2 w_street_2_len;
  ub2 w_street_2_rc;

  OCIBind *w_city_bp;
  OCIBind *w_city_bp1;
  sb2 w_city_ind;
  ub2 w_city_len;
  ub2 w_city_rc;

  OCIBind *w_state_bp;
  OCIBind *w_state_bp1;
  sb2 w_state_ind;
  ub2 w_state_len;
  ub2 w_state_rc;

  OCIBind *w_zip_bp;
  OCIBind *w_zip_bp1;
  sb2 w_zip_ind;
  ub2 w_zip_len;
  ub2 w_zip_rc;

  OCIBind *d_street_1_bp;
  OCIBind *d_street_1_bp1;
  sb2 d_street_1_ind;
  ub2 d_street_1_len;
  ub2 d_street_1_rc;

  OCIBind *d_street_2_bp;
  OCIBind *d_street_2_bp1;
  sb2 d_street_2_ind;
  ub2 d_street_2_len;
  ub2 d_street_2_rc;

  OCIBind *d_city_bp;
  OCIBind *d_city_bp1;
  sb2 d_city_ind;
  ub2 d_city_len;
  ub2 d_city_rc;

  OCIBind *d_state_bp;
  OCIBind *d_state_bp1;
  sb2 d_state_ind;
  ub2 d_state_len;
  ub2 d_state_rc;

  OCIBind *d_zip_bp;
  OCIBind *d_zip_bp1;
  sb2 d_zip_ind;
  ub2 d_zip_len;
  ub2 d_zip_rc;

  OCIBind *c_first_bp;
  OCIBind *c_first_bp1;
  sb2 c_first_ind;
  ub2 c_first_len;
  ub2 c_first_rc;

  OCIBind *c_middle_bp;
  OCIBind *c_middle_bp1;
  sb2 c_middle_ind;
  ub2 c_middle_len;
  ub2 c_middle_rc;

  OCIBind *c_street_1_bp;
  OCIBind *c_street_1_bp1;
```

```
sb2 c_street_1_ind;
ub2 c_street_1_len;
ub2 c_street_1_rc;

OCIBind *c_street_2_bp;
OCIBind *c_street_2_bp1;
sb2 c_street_2_ind;
ub2 c_street_2_len;
ub2 c_street_2_rc;

OCIBind *c_city_bp;
OCIBind *c_city_bp1;
sb2 c_city_ind;
ub2 c_city_len;
ub2 c_city_rc;

OCIBind *c_state_bp;
OCIBind *c_state_bp1;
sb2 c_state_ind;
ub2 c_state_len;
ub2 c_state_rc;

OCIBind *c_zip_bp;
OCIBind *c_zip_bp1;
sb2 c_zip_ind;
ub2 c_zip_len;
ub2 c_zip_rc;

OCIBind *c_phone_bp;
OCIBind *c_phone_bp1;
sb2 c_phone_ind;
ub2 c_phone_len;
ub2 c_phone_rc;

OCIBind *c_since_bp;
OCIBind *c_since_bp1;
sb2 c_since_ind;
ub2 c_since_len;
ub2 c_since_rc;

OCIBind *c_credit_bp;
OCIBind *c_credit_bp1;
sb2 c_credit_ind;
ub2 c_credit_len;
ub2 c_credit_rc;

OCIBind *c_credit_lim_bp;
OCIBind *c_credit_lim_bp1;
sb2 c_credit_lim_ind;
ub2 c_credit_lim_len;
ub2 c_credit_lim_rc;

OCIBind *c_discount_bp;
OCIBind *c_discount_bp1;
sb2 c_discount_ind;
ub2 c_discount_len;
ub2 c_discount_rc;

OCIBind *c_balance_bp;
OCIBind *c_balance_bp1;
sb2 c_balance_ind;
ub2 c_balance_len;
ub2 c_balance_rc;

OCIBind *c_data_bp;
OCIBind *c_data_bp1;
sb2 c_data_ind;
ub2 c_data_len;
ub2 c_data_rc;

OCIBind *h_date_bp;
OCIBind *h_date_bp1;
sb2 h_date_ind;
ub2 h_date_len;
ub2 h_date_rc;

OCIBind *retries_bp;
OCIBind *retries_bp1;
sb2 retries_ind;
ub2 retries_len;
ub2 retries_rc;

OCIBind *cr_date_bp;
OCIBind *cr_date_bp1;
sb2 cr_date_ind;
ub2 cr_date_len;
ub2 cr_date_rc;

OCIBind *byln_bp;
sb2 byln_ind;
ub2 byln_len;
ub2 byln_rc;
};

typedef struct payctx payctx;



payctx *pctx;

int tkvcpinit (void)

{

   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));
#ifdef PLSQLPAY
  /*   prepare the stub for calling plsql stored procedure */
    sprintf ((char*)stmbuf, SQLTXT_STP);
    OCIERROR(errhp,OCIStmtPrepare(pctx->curp0, errhp, stmbuf,
            strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));
#else

    /* customer id !=  0, go by c_id */

//   sqlfile("\\tuxedo\\runtime\\sql816\\paynz.sql",stmbuf);
    sqlfile("/home/tuxedo/runtime/sql816/paynz.sql",stmbuf);
    OCIERROR(errhp,OCIStmtPrepare(pctx->curp0, errhp, stmbuf,
            strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));

    /* customer id ==  0, go by last name */

//   sqlfile("\\tuxedo\\runtime\\sql816\\payz.sql",stmbuf);
    sqlfile("/home/tuxedo/runtime/sql816/payz.sql",stmbuf);
    OCIERROR(errhp,OCIStmtPrepare(pctx->curp1, errhp, stmbuf,
            strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));

#endif
  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);
#ifdef PLSQLPAY
    OCIBND(pctx->curp0, pctx->by_lname_bp,
errhp,":by_lname",ADR(bylastname),
           SIZ(int), SQLT_INT);
#endif
    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,
           SIZ(OCIDate), SQLT_ODT, &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(c_discount), SQLT_FLT, &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(OCIDate),SQLT_ODT, &pctx->cr_date_ind, &pctx->cr_date_len,
           &pctx->cr_date_rc);
#ifndef PLSQLPAY

/* ---- 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,
           SIZ(OCIDate), SQLT_ODT, &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(c_discount), SQLT_FLT, &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(OCIDate),SQLT_ODT, &pctx->cr_date_ind, &pctx->cr_date_len,
           &pctx->cr_date_rc);
#endif

   return (0);

}



tkvcp ()

{

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;

#ifdef PLSQLPAY
    execstatus=OCIStmtExecute(tpcsvc,pctx->curp0,errhp,1,0,0,0,OCI_DEFAULT
| OCI_COMMIT_ON_SUCCESS);
#else
  if(bylastname) {
    execstatus=OCIStmtExecute(tpcsvc,pctx->curp1,errhp,1,0,0,0,OCI_DEFAULT
| OCI_COMMIT_ON_SUCCESS);
  } else {
    execstatus=OCIStmtExecute(tpcsvc,pctx->curp0,errhp,1,0,0,0,OCI_DEFAULT
| OCI_COMMIT_ON_SUCCESS);
  }
#endif

   if(execstatus != OCI_SUCCESS)
   {
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      errcode = OCIERROR(errhp,execstatus);
      if (errcode == OCI_NO_DATA)
         return(SVCERR_NOCUSTOMER);
      if(errcode == NOT_SERIALIZABLE)
      {
         retries++;
         goto retry;
      }
      else
      if (errcode == RECOVERR)
      {
         retries++;
         goto retry;
      }
      else
           if (errcode == SNAPSHOT_TOO_OLD)
      {
         retries++;
         goto retry;
      }
      else
      {
         return(SVCERR_OCI);
      };
   };

   return(SVC_NOERROR);

}   // tkvcp

void tkvcpdone ()

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

pldel.c

```
// pldel.c
// Copyright (c) 1998 1999 Unisys Corporation
// Copyright (c) 1995  Oracle Corp, Redwood Shores, CA
```

```
#include "tpcc.h"
#ifdef TUX
#include <userlog.h>
#endif

#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
#define SQLTXT0 "SELECT substr(value,1,5) FROM v$parameter \
   WHERE name = 'instance_number'"
#endif

#ifdef PLSQLDEL
#define SQLTXT "BEGIN delivery.deliver (:w_id, :carrier_id, :order_id,\
 :retry); END;"
#else
# ifdef DMLRETDEL
#define SQLTXT1 "DELETE FROM new_order WHERE no_d_id = :d_id \
                 AND no_w_id = :w_id and rownum <= 1 \
                RETURNING no_o_id into :o_id "
# else
#define SQLTXT1A "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 1, no_o_id,
new_order.rowid, o_c_id, orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 1 AND o_w_id = :w_id AND o_d_id = 1
AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1B "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 2, no_o_id,
new_order.rowid, o_c_id, orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 2 AND o_w_id = :w_id AND o_d_id = 2
AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1C "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 3, no_o_id,
new_order.rowid, o_c_id, orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 3 AND o_w_id = :w_id AND o_d_id = 3
AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1D "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 4, no_o_id,
new_order.rowid, o_c_id, orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 4 AND o_w_id = :w_id AND o_d_id = 4
AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1E "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 5, no_o_id,
new_order.rowid, o_c_id, orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 5 AND o_w_id = :w_id AND o_d_id = 5
AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1F "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 6, no_o_id,
new_order.rowid, o_c_id, orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 6 AND o_w_id = :w_id AND o_d_id = 6
AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1G "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 7, no_o_id,
new_order.rowid, o_c_id, orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 7 AND o_w_id = :w_id AND o_d_id = 7
AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1H "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 8, no_o_id,
new_order.rowid, o_c_id, orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 8 AND o_w_id = :w_id AND o_d_id = 8
AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1I "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 9, no_o_id,
new_order.rowid, o_c_id, orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 9 AND o_w_id = :w_id AND o_d_id = 9
AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1J "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 10, no_o_id,
new_order.rowid, o_c_id, orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 10 AND o_w_id = :w_id AND o_d_id = 10
AND \
   o_id = no_o_id AND rownum <= 1"

#define SQLTXT2 "DELETE FROM new_order WHERE rowid = :no_rowid"
#endif

#ifdef DMLRETDEL
#define SQLTXT3 "UPDATE orders SET o_carrier_id = :carrier_id \
                 WHERE o_id = :o_id and o_d_id = :d_id and o_w_id = :w_id
\
                 returning o_c_id into :o_c_id"
#else
#define SQLTXT3 "UPDATE orders SET o_carrier_id = :carrier_id \
   WHERE rowid = :o_rowid"
#endif
```

```
#ifdef DMLRETDEL
#define SQLTXT4 "UPDATE /*+ buffer */ order_line SET ol_delivery_d =
:cr_date \
   WHERE ol_w_id = :w_id AND ol_d_id = :d_id AND ol_o_id = :o_id \
   RETURNING ol_amount into :ol_amount "
#else
#define SQLTXT4 "UPDATE order_line SET ol_delivery_d = :cr_date \
   WHERE ol_w_id = :w_id AND ol_d_id = :d_id AND ol_o_id = :o_id"

#define SQLTXT5A "\
SELECT :d_id1, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id1 AND ol_o_id = :o_id1 UNION ALL \
SELECT :d_id2, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id2 AND ol_o_id = :o_id2 UNION ALL \
"

#define SQLTXT5B "\
SELECT :d_id3, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id3 AND ol_o_id = :o_id3 UNION ALL \
SELECT :d_id4, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id4 AND ol_o_id = :o_id4 UNION ALL \
"

#define SQLTXT5C "\
SELECT :d_id5, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id5 AND ol_o_id = :o_id5 UNION ALL \
SELECT :d_id6, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id6 AND ol_o_id = :o_id6 UNION ALL \
"

#define SQLTXT5D "\
SELECT :d_id7, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id7 AND ol_o_id = :o_id7 UNION ALL \
SELECT :d_id8, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id8 AND ol_o_id = :o_id8 UNION ALL \
"

#define SQLTXT5E "\
SELECT :d_id9, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id9 AND ol_o_id = :o_id9 UNION ALL \
SELECT :d_id10, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id10 AND ol_o_id = :o_id10"

#endif
#endif /* PLSQLDEL */

#define SQLTXT6 "UPDATE customer SET c_balance = c_balance + :amt, \
   c_delivery_cnt = c_delivery_cnt + 1 WHERE c_w_id = :w_id AND \
   c_d_id = :d_id AND c_id = :c_id"

#define NDISTS 10
#define ROWIDLEN 20


struct delctx {
   sb2 del_o_id_ind[NDISTS];
   sb2 cons_ind[NDISTS];
   sb2 w_id_ind[NDISTS];
   sb2 d_id_ind[NDISTS];
   sb2 c_id_ind[NDISTS];
   sb2 del_date_ind[NDISTS];
   sb2 carrier_id_ind[NDISTS];
   sb2 amt_ind[NDISTS];
   sb2 no_rowid_ind[NDISTS];
   sb2 o_rowid_ind[NDISTS];
#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   sb2 inum_ind;
#endif

#ifdef DMLRETDEL
   ub4 del_o_id_len[NDISTS];
   ub4 c_id_len[NDISTS];
   int oid_ctx;
   int cid_ctx;
   OCIBind *olamt_bp;
#else
   ub2 del_o_id_len[NDISTS];
   ub2 c_id_len[NDISTS];
#endif

   ub2 cons_len[NDISTS];
   ub2 w_id_len[NDISTS];
   ub2 d_id_len[NDISTS];
   ub2 del_date_len[NDISTS];
   ub2 carrier_id_len[NDISTS];
   ub2 amt_len[NDISTS];
   ub2 no_rowid_len[NDISTS];
   ub2 no_rowid_ptr_len[NDISTS];
   ub2 o_rowid_len[NDISTS];
   ub2 o_rowid_ptr_len[NDISTS];
#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   ub2 inum_len;
#endif

   ub2 del_o_id_rcode[NDISTS];
   ub2 cons_rcode[NDISTS];
   ub2 w_id_rcode[NDISTS];
   ub2 d_id_rcode[NDISTS];
   ub2 c_id_rcode[NDISTS];
   ub2 del_date_rcode[NDISTS];
   ub2 carrier_id_rcode[NDISTS];
   ub2 amt_rcode[NDISTS];
   ub2 no_rowid_rcode[NDISTS];
   ub2 o_rowid_rcode[NDISTS];
#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   ub2 inum_rcode;
#endif

   int del_o_id[NDISTS];
   int cons[NDISTS];
   int w_id[NDISTS];
   int d_id[NDISTS];
   int c_id[NDISTS];
   int carrier_id[NDISTS];
   int amt[NDISTS];
   ub4 del_o_id_rcnt;
   int retry;
   OCIRowid *no_rowid_ptr[NDISTS];
   OCIRowid *o_rowid_ptr[NDISTS];
   OCIDate del_date[NDISTS];
#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
```

```
   char inum[10];
#endif
   OCIStmt *curd0;
   OCIStmt *curd1;
   OCIStmt *curd2;
   OCIStmt *curd3;
   OCIStmt *curd4;
   OCIStmt *curd5;
   OCIStmt *curd6;
   OCIStmt *curdtest;

   OCIBind *w_id_bp;
   OCIBind *w_id_bp3;
   OCIBind *w_id_bp4;
   OCIBind *w_id_bp5;
   OCIBind *w_id_bp6;
   OCIBind *d_id_bp;
   OCIBind *d_id_bp3;
   OCIBind *d_id_bp4;
   OCIBind *d_id_bp6;
   OCIBind *o_id_bp;
   OCIBind *cr_date_bp;
   OCIBind *c_id_bp;
   OCIBind *c_id_bp3;
   OCIBind *no_rowid_bp;
   OCIBind *carrier_id_bp;
   OCIBind *o_rowid_bp;
   OCIBind *del_o_id_bp;
   OCIBind *del_o_id_bp3;
   OCIBind *amt_bp;
   OCIBind *bstr1_bp[10];
   OCIBind *bstr2_bp[10];
   OCIBind *retry_bp;
   OCIDefine *inum_dp;
   OCIDefine *d_id_dp;
   OCIDefine *del_o_id_dp;
   OCIDefine *no_rowid_dp;
   OCIDefine *c_id_dp;
   OCIDefine *o_rowid_dp;
   OCIDefine *cons_dp;
   OCIDefine *amt_dp;

   int norow;
};

typedef struct delctx delctx;

delctx *dctx;

#ifdef DMLRETDEL
struct amtctx {
  int ol_amt[NDISTS][NITEMS];
  sb2 ol_amt_ind[NDISTS][NITEMS];
  ub4 ol_amt_len[NDISTS][NITEMS];
  ub2 ol_amt_rcode[NDISTS][NITEMS];
  int ol_cnt[NDISTS];
};
typedef struct amtctx amtctx;
amtctx *actx;

#endif
```

```
#ifdef DMLRETDEL
extern sb4 no_data();

sb4 TPC_oid_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
               dvoid **bufpp, ub4 **alenp, ub1 *piecep,
               dvoid **indpp, ub2 **rcodepp)
{
  *bufpp = &dctx->del_o_id[iter];
  *indpp= &dctx->del_o_id_ind[iter];
  dctx->del_o_id_len[iter]=sizeof(dctx->del_o_id[0]);
  *alenp= &dctx->del_o_id_len[iter];
  *rcodepp = &dctx->del_o_id_rcode[iter];
  *piecep =OCI_ONE_PIECE;
  return (OCI_CONTINUE);
}
sb4 cid_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
               dvoid **bufpp, ub4 **alenp, ub1 *piecep,
               dvoid **indpp, ub2 **rcodepp)
{
  *bufpp = &dctx->c_id[iter];
  *indpp= &dctx->c_id_ind[iter];
  dctx->c_id_len[iter]=sizeof(dctx->c_id[0]);
  *alenp= &dctx->c_id_len[iter];
  *rcodepp = &dctx->c_id_rcode[iter];
  *piecep =OCI_ONE_PIECE;
  return (OCI_CONTINUE);
}


sb4 amt_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
               dvoid **bufpp, ub4 **alenp, ub1 *piecep,
               dvoid **indpp, ub2 **rcodepp)
{
  amtctx *actx;
  actx =(amtctx*)ctxp;
  actx->ol_cnt[iter]=actx->ol_cnt[iter]+1;
  *bufpp = &actx->ol_amt[iter][index];
  *indpp= &actx->ol_amt_ind[iter][index];
  actx->ol_amt_len[iter][index]=sizeof(actx->ol_amt[0][0]);
  *alenp= &actx->ol_amt_len[iter][index];
  *rcodepp = &actx->ol_amt_rcode[iter][index];
  *piecep =OCI_ONE_PIECE;
  return (OCI_CONTINUE);
}

#endif


tkvcdinit ()

{

#ifndef DMLRETDEL
   int i,j;
   char bstr1[10];
   char bstr2[10];
# endif /* !DMLRETDEL */
   text stmbuf[SQL_BUF_SIZE];
```

```
   dctx = (delctx *) malloc (sizeof(delctx));
   memset(dctx,(char)0,sizeof(delctx));
   dctx->norow = 0;
#ifdef DMLRETDEL
   actx = (amtctx *) malloc (sizeof(amtctx));
   memset(actx,(char)0,sizeof(amtctx));
#else
   for(i=0;i<NDISTS;i++) {
    OCIERROR(errhp, OCIDescriptorAlloc(tpcenv,(dvoid**)&dctx-
>o_rowid_ptr[i],
             OCI_DTYPE_ROWID,0,(dvoid**)0));
    OCIERROR(errhp, OCIDescriptorAlloc(tpcenv,(dvoid**)&dctx-
>no_rowid_ptr[i],
             OCI_DTYPE_ROWID,0,(dvoid**)0));
   }
#endif

#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd0), OCI_HTYPE_STMT, 0,
(dvoid**)0);
   sprintf ((char *) stmbuf, SQLTXT0);
   OCIStmtPrepare(dctx->curd0, errhp, stmbuf, strlen((char
*)stmbuf),OCI_NTV_SYNTAX, OCI_DEFAULT);

   OCIDFNRA(dctx->curd0, dctx->inum_dp,errhp,1,dctx->inum,SIZ(dctx-
>inum),SQLT_STR,
          &(dctx->inum_ind),&(dctx->inum_len),&(dctx->inum_rcode));
#endif

/* If PLSQLDEL and ISO? are both defined, then they both try to use
   curd0! This could cause a problem. Will try to fix later - VMM 12/30/97
*/

#ifdef PLSQLDEL
   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd0), OCI_HTYPE_STMT,
                  0, (dvoid**)0);
   sprintf ((char *) stmbuf, SQLTXT);
   OCIStmtPrepare(dctx->curd0, errhp, stmbuf, strlen((char *)stmbuf),
                  OCI_NTV_SYNTAX, OCI_DEFAULT);
   OCIBND(dctx->curd0, dctx->w_id_bp , errhp,":w_id",ADR(w_id),SIZ(int),
          SQLT_INT);
   OCIBND(dctx->curd0, dctx->carrier_id_bp , errhp,":carrier_id",
          ADR(dctx->carrier_id), SIZ(int), SQLT_INT);

   OCIBNDRAA(dctx->curd0, dctx->o_id_bp, errhp, ":order_id",
             dctx->del_o_id,SIZ(int),SQLT_INT, dctx->del_o_id_ind,
             dctx->del_o_id_len,dctx->del_o_id_rcode,NDISTS,
             &dctx->del_o_id_rcnt);
   OCIBND(dctx->curd0, dctx->retry_bp , errhp,":retry",ADR(dctx->retry),
          SIZ(int),SQLT_INT);
#else
# ifdef DMLRETDEL
   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd1), OCI_HTYPE_STMT, 0,
(dvoid**)0);
   sprintf ((char *) stmbuf, "%s",    SQLTXT1);
   OCIStmtPrepare(dctx->curd1, errhp, stmbuf, strlen((char
*)stmbuf),OCI_NTV_SYNTAX, OCI_DEFAULT);

   OCIBND(dctx->curd1, dctx->w_id_bp,errhp,":w_id",dctx->w_id,SIZ(int),
          SQLT_INT);
   OCIBNDRA(dctx->curd1, dctx->d_id_bp,errhp,":d_id",dctx->d_id,SIZ(int),
          SQLT_INT,NULL,NULL,NULL);

   OCIBNDRAD(dctx->curd1, dctx->del_o_id_bp, errhp, ":o_id",
             SIZ(int),SQLT_INT,NULL,
             &dctx->oid_ctx,no_data,TPC_oid_data);
# else

   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd1), OCI_HTYPE_STMT, 0,
(dvoid**)0);
   sprintf ((char *) stmbuf, "%s%s%s%s%s%s%s%s%s%s",    SQLTXT1A,
                                                       SQLTXT1B,
                                                       SQLTXT1C,
                                                       SQLTXT1D,
                                                       SQLTXT1E,
                                                       SQLTXT1F,
                                                       SQLTXT1G,
                                                       SQLTXT1H,
                                                       SQLTXT1I,
                                                       SQLTXT1J
       );
   OCIStmtPrepare(dctx->curd1, errhp, stmbuf, strlen((char
*)stmbuf),OCI_NTV_SYNTAX, OCI_DEFAULT);

   OCIERROR(errhp,
      OCIAttrSet(dctx->curd1,OCI_HTYPE_STMT,(dvoid*)&dctx->norow,0,
                 OCI_ATTR_PREFETCH_ROWS,errhp));



   /* bind variables */

   OCIBND(dctx->curd1, dctx-
>w_id_bp,errhp,":w_id",ADR(w_id),SIZ(int),SQLT_INT);

   OCIDFNRA(dctx->curd1, dctx->d_id_dp,errhp,1,dctx->d_id,SIZ(int),
            SQLT_INT, dctx->d_id_ind,dctx->d_id_len,dctx->d_id_rcode);
   OCIDFNRA(dctx->curd1, dctx->del_o_id_dp,errhp,2,dctx->del_o_id,
            SIZ(int), SQLT_INT,dctx->del_o_id_ind,
            dctx->del_o_id_len, dctx->del_o_id_rcode);
   OCIDFNRA(dctx->curd1, dctx->no_rowid_dp,errhp,3,dctx->no_rowid_ptr,
            SIZ(OCIRowid *), SQLT_RDD,dctx->no_rowid_ind,
            dctx->no_rowid_len, dctx->no_rowid_rcode);
   OCIDFNRA(dctx->curd1, dctx->c_id_dp,errhp,4,dctx->c_id,SIZ(dctx-
>c_id[0]),
            SQLT_INT, dctx->c_id_ind,dctx->c_id_len,dctx->c_id_rcode);
   OCIDFNRA(dctx->curd1, dctx->o_rowid_dp,errhp,5,dctx->o_rowid_ptr,
            SIZ(OCIRowid *), SQLT_RDD,dctx->o_rowid_ind,
            dctx->o_rowid_len, dctx->o_rowid_rcode);

   /* open second cursor */

   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd2), OCI_HTYPE_STMT, 0,
(dvoid**)0);
   sprintf ((char *) stmbuf, SQLTXT2);
   OCIStmtPrepare(dctx->curd2, errhp, stmbuf, strlen((char *)stmbuf),
                  OCI_NTV_SYNTAX, OCI_DEFAULT);

   /* bind variables */
   OCIBNDRA(dctx->curd2, dctx->no_rowid_bp,errhp,":no_rowid",&(dctx-
>no_rowid_ptr[0]),
           SIZ(dctx->no_rowid_ptr[0]),SQLT_RDD,dctx->no_rowid_ind,
```

```
         dctx->no_rowid_len,dctx->no_rowid_rcode);

# endif /*DMLRETDEL*/

   /* open third cursor */

   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd3), OCI_HTYPE_STMT, 0,
(dvoid**)0);
   sprintf ((char *) stmbuf, SQLTXT3);
   OCIStmtPrepare(dctx->curd3, errhp, stmbuf, strlen((char *)stmbuf),
                OCI_NTV_SYNTAX, OCI_DEFAULT);


   /* bind variables */

   OCIBNDRA(dctx->curd3, dctx->carrier_id_bp,errhp,":carrier_id",dctx-
>carrier_id,
          SIZ(dctx->carrier_id[0]),SQLT_INT,dctx->carrier_id_ind,
          dctx->carrier_id_len,dctx->carrier_id_rcode);

# ifdef DMLRETDEL
   OCIBNDRA(dctx->curd3, dctx->w_id_bp3, errhp, ":w_id", dctx-
>w_id,SIZ(int),
            SQLT_INT, NULL, NULL, NULL);
   OCIBNDRA(dctx->curd3, dctx->d_id_bp3, errhp, ":d_id", dctx-
>d_id,SIZ(int),
            SQLT_INT,NULL, NULL, NULL);
   OCIBNDRA(dctx->curd3, dctx->del_o_id_bp3, errhp, ":o_id", dctx-
>del_o_id,
            SIZ(int), SQLT_INT,NULL,NULL,NULL);
   OCIBNDRAD(dctx->curd3, dctx->c_id_bp3, errhp, ":o_c_id", SIZ(int),
            SQLT_INT,NULL,&dctx->cid_ctx,no_data, cid_data);
# else

   OCIBNDRA(dctx->curd3, dctx->o_rowid_bp,errhp,":o_rowid",&(dctx-
>o_rowid_ptr[0]),
           SIZ(dctx->o_rowid_ptr[0]),SQLT_RDD,dctx->o_rowid_ind,
           dctx->o_rowid_ptr_len,dctx->o_rowid_rcode);
#endif

   /* open fourth cursor */

   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd4), OCI_HTYPE_STMT, 0,
(dvoid**)0);
   sprintf ((char *) stmbuf, SQLTXT4);
   OCIStmtPrepare(dctx->curd4, errhp, stmbuf, strlen((char *)stmbuf),
                OCI_NTV_SYNTAX, OCI_DEFAULT);

   /* bind variables */

   OCIBND(dctx->curd4, dctx->w_id_bp4,errhp,":w_id",dctx->w_id,
          SIZ(int), SQLT_INT);
   OCIBND(dctx->curd4, dctx->d_id_bp4,errhp,":d_id",dctx->d_id,
          SIZ(int), SQLT_INT);
   OCIBND(dctx->curd4, dctx->o_id_bp,errhp,":o_id",dctx->del_o_id,
            SIZ(int),SQLT_INT);
   OCIBND(dctx->curd4, dctx->cr_date_bp,errhp,":cr_date", dctx->del_date,
            SIZ(OCIDate), SQLT_ODT);
# ifdef DMLRETDEL
   OCIBNDRAD(dctx->curd4, dctx->olamt_bp, errhp, ":ol_amount",
             SIZ(int), SQLT_INT,NULL, actx,no_data,amt_data);
# else



   /* open fifth cursor */

   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd5), OCI_HTYPE_STMT, 0,
(dvoid**)0);
   sprintf ((char *) stmbuf, "%s%s%s%s%s",   SQLTXT5A,
                                             SQLTXT5B,
                                             SQLTXT5C,
                                             SQLTXT5D,
                                             SQLTXT5E
   );
   OCIStmtPrepare(dctx->curd5, errhp, stmbuf, strlen((char *)stmbuf),
                OCI_NTV_SYNTAX, OCI_DEFAULT);

   OCIERROR(errhp,
      OCIAttrSet(dctx->curd5,OCI_HTYPE_STMT,(dvoid*)&dctx->norow,0,
                OCI_ATTR_PREFETCH_ROWS,errhp));

   /* bind variables */

   OCIBND(dctx->curd5,dctx-
>w_id_bp,errhp,":w_id",ADR(w_id),SIZ(w_id),SQLT_INT);
   for (i = 0; i < NDISTS; i++) {
      sprintf (bstr1, ":d_id%d", i + 1);
      sprintf (bstr2, ":o_id%d", i + 1);
      OCIBNDRA(dctx->curd5,dctx->bstr1_bp[i],errhp,bstr1,ADR(dctx-
>d_id[i]),
               SIZ(dctx->d_id[0]),SQLT_INT, &(dctx->d_id_ind[i]),
               &(dctx->d_id_len[i]),&(dctx->d_id_rcode[i]));
      OCIBNDRA(dctx->curd5,dctx->bstr2_bp[i],errhp,bstr2,ADR(dctx-
>del_o_id[i]),
               SIZ(dctx->del_o_id[0]),SQLT_INT, &(dctx->del_o_id_ind[i]),
               &(dctx->del_o_id_len[i]),&(dctx->del_o_id_rcode[i]));
   }

   OCIDFNRA(dctx->curd5,dctx->cons_dp,errhp,1,dctx->cons,SIZ(dctx-
>cons[0]),SQLT_INT,
          dctx->cons_ind,dctx->cons_len,dctx->cons_rcode);
   OCIDFNRA(dctx->curd5,dctx->amt_dp,errhp,2,dctx->amt,SIZ(dctx-
>amt[0]),SQLT_INT,
          dctx->amt_ind,dctx->amt_len,dctx->amt_rcode);
#endif
   /* open sixth cursor */

   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd6), OCI_HTYPE_STMT, 0,
(dvoid**)0);
   sprintf ((char *) stmbuf, SQLTXT6);
   OCIStmtPrepare(dctx->curd6, errhp, stmbuf, strlen((char *)stmbuf),
                OCI_NTV_SYNTAX, OCI_DEFAULT);

   /* bind variables */


   OCIBND(dctx->curd6,dctx->amt_bp,errhp,":amt",dctx->amt,SIZ(int),
          SQLT_INT);
   OCIBND(dctx->curd6,dctx->w_id_bp6,errhp,":w_id",dctx->w_id,SIZ(int),
            SQLT_INT);
   OCIBND(dctx->curd6,dctx->d_id_bp6,errhp,":d_id",dctx->d_id,SIZ(int),
```

```
            SQLT_INT);
   OCIBND(dctx->curd6,dctx->c_id_bp,errhp,":c_id",dctx->c_id,SIZ(int),
            SQLT_INT);
#endif
   return (0);

}


void shiftdata(from)
int from ;
{
   int i;
   for (i=from;i<NDISTS-1; i++)
   {
     dctx->del_o_id_ind[i] = dctx->del_o_id_ind[i+1];
     dctx->del_o_id[i] = dctx->del_o_id[i+1];
     dctx->w_id[i] = dctx->w_id[i+1];
     dctx->d_id[i] = dctx->d_id[i+1];
     dctx->carrier_id[i] = dctx->carrier_id[i+1];
   }
}


tkvcd ()

{

   int i, j;
   int rpc,rcount,count;
   int invalid;
# ifndef DMLRETDEL
   int tmp_id,v;
   int tmp_amt;
# endif /*  !DMLRETDEL */

#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   int hasno;
   int reread;
   char sdate[30];

   OCIStmtExecute(tpcsvc,dctx->curd0,errhp,1,0,0,0,OCI_DEFAULT);
   sysdate (sdate);
   printf ("Delivery started at %s on node %s\n", sdate, dctx->inum);
#endif
#ifdef PLSQLDEL
   for (i = 0; i < NDISTS; i++)
   {
      dctx->del_o_id_ind[i] = TRUE;
      dctx->del_o_id_len[i] = sizeof(int);
   }

   OCIERROR(errhp,
      OCIStmtExecute(tpcsvc,dctx->curd0,errhp,1,0,0,0,OCI_DEFAULT));

   for (i = 0; i < NDISTS; i++)
   {
      del_o_id[i] = 0;
      if (dctx->del_o_id_ind[i] == 0)
      {
        del_o_id[i] = dctx->del_o_id[i];
      }
   }
#else

retry:

#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   reread = 1;
#endif

#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
iso:
#endif

   invalid = 0;

   /* initialization for array operations */

   for (i = 0; i < NDISTS; i++) {
      dctx->del_o_id_ind[i] = TRUE;
      dctx->cons_ind[i] = TRUE;
      dctx->w_id_ind[i] = TRUE;
      dctx->d_id_ind[i] = TRUE;
      dctx->c_id_ind[i] = TRUE;
      dctx->del_date_ind[i] = TRUE;
      dctx->carrier_id_ind[i] = TRUE;
      dctx->amt_ind[i] = TRUE;
      dctx->no_rowid_ind[i] = TRUE;
      dctx->o_rowid_ind[i] = TRUE;

      dctx->del_o_id_len[i] = SIZ(dctx->del_o_id[0]);
      dctx->cons_len[i] = SIZ(dctx->cons[0]);
      dctx->w_id_len[i] = SIZ(dctx->w_id[0]);
      dctx->d_id_len[i] = SIZ(dctx->d_id[0]);
      dctx->c_id_len[i] = SIZ(dctx->c_id[0]);
      dctx->del_date_len[i] = DEL_DATE_LEN;
      dctx->carrier_id_len[i] = SIZ(dctx->carrier_id[0]);
      dctx->amt_len[i] = SIZ(dctx->amt[0]);
      dctx->no_rowid_len[i] = ROWIDLEN;
      dctx->o_rowid_len[i] = ROWIDLEN;
      dctx->o_rowid_ptr_len[i] = SIZ(dctx->o_rowid_ptr[0]);
      dctx->no_rowid_ptr_len[i] = SIZ(dctx->no_rowid_ptr[0]);

      dctx->w_id[i] = w_id;
      dctx->d_id[i] = i+1;
      dctx->carrier_id[i] = o_carrier_id;
      memcpy(&dctx->del_date[i],&cr_date,sizeof(OCIDate));
   }

#ifdef DMLRETDEL /* VMM 1/13/98 */
   memset(actx,(char)0,sizeof(amtctx));
#endif /* DMLRETDEL */
   /* array select from new_order and orders tables */

   execstatus=OCIStmtExecute(tpcsvc,dctx-
>curd1,errhp,NDISTS,0,0,0,OCI_DEFAULT);
   if((execstatus != OCI_SUCCESS) && (execstatus != OCI_NO_DATA)) {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     errcode = OCIERROR(errhp,execstatus);
     if(errcode == NOT_SERIALIZABLE) {
        retries++;
```

```
         goto retry;
      } else if (errcode == RECOVERR) {
         retries++;
         goto retry;
      } else if (errcode == SNAPSHOT_TOO_OLD) {
         retries++;
         goto retry;
      } else {
         return -1;
      }
   }
   /* mark districts with no new order */
   OCIAttrGet(dctx-
>curd1,OCI_HTYPE_STMT,&rcount,0,OCI_ATTR_ROW_COUNT,errhp);
   rpc = rcount;
#ifdef DMLRETDEL   /* we have to compress the array here */
   if (rcount != NDISTS )
   {
     int j = 0;
     for (i=0;i < NDISTS; i++)
     {
      if (dctx->del_o_id_ind[j] == 0) /* there is data here */
        j++;
      else
        shiftdata(j);
     }
   }
#else
   invalid = NDISTS - rcount;
   for (i = rpc; i < NDISTS; i++) {
      dctx->del_o_id_ind[i] = NA;
      dctx->w_id_ind[i] = NA;
      dctx->d_id_ind[i] = NA;
      dctx->c_id_ind[i] = NA;
      dctx->carrier_id_ind[i] = NA;
      dctx->no_rowid_ind[i] = NA;
      dctx->o_rowid_ind[i] = NA;
   }
#endif

#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   if (invalid) {
      sysdate (sdate);
      for (i = 1; i <= NDISTS; i++) {
         hasno = 0;
         for (j = 0; j < rpc; j++) {
            if (dctx->d_id[j] == i) {
               hasno = 1;
               break;
            }
         }
         if (!hasno)
            printf ("Delivery [dist %d] found no new order at %s\n", i,
sdate);
      }
      if (reread) {
         sleep (60);
         sysdate (sdate);
         printf ("Delivery wake up at %s\n", sdate);
         reread = 0;
         goto iso;
      }
   }
#endif

#ifndef DMLRETDEL
   /* array delete of new_order table */
   execstatus=OCIStmtExecute(tpcsvc,dctx-
>curd2,errhp,rpc,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 if (errcode == SNAPSHOT_TOO_OLD) {
        retries++;
        goto retry;
     } else {
        return -1;
     }
   }

   /* mark districts with no new order */
   OCIAttrGet(dctx-
>curd2,OCI_HTYPE_STMT,&rcount,0,OCI_ATTR_ROW_COUNT,errhp);

   if (rcount != rpc) {
#ifdef TUX
      userlog ("Error in TPC-C server %d: %d rows selected, %d rows
deleted\n",
               proc_no, rpc, rcount);
#else
      fprintf (stderr,
               "Error in TPC-C server %d: %d rows selected, %d rows
deleted\n",
               proc_no, rpc, rcount);
#endif /* TUX */
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      return (DEL_ERROR);
   }
#endif /* DMLRETDEL */

   execstatus=OCIStmtExecute(tpcsvc,dctx-
>curd3,errhp,rpc,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 if (errcode == SNAPSHOT_TOO_OLD) {
        retries++;
        goto retry;
     } else {
        return -1;
```

```
      }
   }

   OCIAttrGet(dctx-
>curd3,OCI_HTYPE_STMT,&rcount,0,OCI_ATTR_ROW_COUNT,errhp);

   if (rcount != rpc) {
#ifdef TUX
      userlog ("Error in TPC-C server %d: %d rows selected, %d ords
updated\n",
               proc_no, rpc, rcount);
#else
      fprintf (stderr,
               "Error in TPC-C server %d: %d rows selected, %d ords
updated\n",
               proc_no, rpc, rcount);
#endif
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      return (SVCERR_OCI);
   }

   /* array update of order_line table */
   execstatus=OCIStmtExecute(tpcsvc,dctx-
>curd4,errhp,rpc,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 if (errcode == SNAPSHOT_TOO_OLD) {
        retries++;
        goto retry;
     } else {
        return -1;
     }
   }
#ifdef DMLRETDEL
  OCIAttrGet(dctx-
>curd4,OCI_HTYPE_STMT,&rcount,NULL,OCI_ATTR_ROW_COUNT,errhp);
/* add up amounts */
   count=0;
   for (i=0;i<rpc;i++)
   {
     dctx->amt[i]=0;
     for (j=0;j<actx->ol_cnt[i];j++)
       if ( actx->ol_amt_rcode[i][j] == 0)
       {
         dctx->amt[i] = dctx->amt[i] + actx->ol_amt[i][j];
         count = count+1;
       }
   }
   if (rcount > rpc*NITEMS) {
      userlog ("Error in TPC-C server %d: %d ordnrs updated, %d ordl
updated\n",
               proc_no, rpc, rcount);
   }
#else
   /* array select from order_line table */
   execstatus=OCIStmtExecute(tpcsvc,dctx-
>curd5,errhp,rpc,0,0,0,OCI_DEFAULT);
   if((execstatus != OCI_SUCCESS) && (execstatus != OCI_NO_DATA)) {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     errcode = OCIERROR(errhp,execstatus);
     if(errcode == NOT_SERIALIZABLE) {
        retries++;
        goto retry;
     } else if (errcode == RECOVERR) {
        retries++;
        goto retry;
     } else if (errcode == SNAPSHOT_TOO_OLD) {
        retries++;
        goto retry;
     } else {
        return -1;
     }
   }

   OCIAttrGet(dctx-
>curd5,OCI_HTYPE_STMT,&rcount,0,OCI_ATTR_ROW_COUNT,errhp);
   if (rcount != rpc) {
#ifdef TUX
      userlog ("Error in TPC-C server %d: %d rows selected, %d ordl
selected\n",
               proc_no, rpc, rcount);
#else
      fprintf (stderr,
               "Error in TPC-C server %d: %d rows selected, %d ordl
selected\n",
               proc_no, rpc, rcount);
#endif
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      return (SVCERR_OCI);
   }

   /* reorder amount selected if necessary */

   for (i = 0; i < rpc; i++) {
      if (dctx->cons[i] != dctx->d_id[i]) {
#ifdef TUX
         userlog ("TPC-C server %d: reordering amount\n", proc_no);
#else
         fprintf (stderr, "TPC-C server %d: reordering amount\n",
proc_no);
#endif
         for (j = i + 1; j < rpc; j++) {
            if (dctx->cons[j] == dctx->d_id[i]) {
               tmp_id = dctx->cons[i];
               dctx->cons[i] = dctx->cons[j];
               dctx->cons[j] = tmp_id;
               tmp_amt = dctx->amt[i];
               dctx->amt[i] = dctx->amt[j];
               dctx->amt[j] = tmp_amt;
               break;
            }
         }
         if (j >= rpc) {
#ifdef TUX
```

```
            userlog ("Error in TPC-C server %d: missing ordl?\n",
proc_no);
#else
            fprintf (stderr,
                     "Error in TPC-C server %d: missing ordl?\n",
proc_no);
#endif
           OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
            return (SVCERR_OCI);
         }
      }
   }
#endif
#if defined(ISO5) || defined(ISO6)
   printf ("d_id:amount\n");
   for (i = 0; i < rpc; i++)
      printf ("%d:%.2f ", dctx->d_id[i], (float)dctx->amt[i]/100);
   printf ("\n");
#endif

   /* array update of customer table */
#if defined(ISO5) || defined (ISO6)
   execstatus=OCIStmtExecute(tpcsvc,dctx->curd6,errhp,rpc,0,0,0,
                             OCI_DEFAULT);
#else
   execstatus=OCIStmtExecute(tpcsvc,dctx->curd6,errhp,rpc,0,0,0,
                                OCI_COMMIT_ON_SUCCESS | OCI_DEFAULT);
#endif

   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 if (errcode == SNAPSHOT_TOO_OLD) {
        retries++;
        goto retry;
     } else {
        return -1;
     }
   }

   OCIAttrGet(dctx-
>curd6,OCI_HTYPE_STMT,&rcount,0,OCI_ATTR_ROW_COUNT,errhp);

   if (rcount != rpc) {
#ifdef TUX
      userlog ("Error in TPC-C server %d: %d rows selected, %d cust
updated\n",
               proc_no, rpc, rcount);
#else
      fprintf (stderr,
               "Error in TPC-C server %d: %d rows selected, %d cust
updated\n",
               proc_no, rpc, rcount);
#endif
      OCITransRollback(tpcsvc, errhp, OCI_DEFAULT);
      return (SVCERR_OCI);
   }

#if defined(ISO5) || defined(ISO6)
   sysdate (sdate);
#ifdef ISO5
   printf ("Delivery sleep before commit at %s\n", sdate);
#else
   printf ("Delivery sleep before abort at %s\n", sdate);
#endif
   sleep (60);
   sysdate (sdate);
   printf ("Delivery wake up at %s\n", sdate);
#endif

#ifdef ISO6
   printf("Delivery ISO6 Rolling back.\n");
   OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
#endif

#ifdef ISO5
   OCITransCommit(tpcsvc, errhp, OCI_DEFAULT);
#endif

#if defined(ISO5) || defined(ISO6)
   sysdate (sdate);
   printf ("Delivery completed at: %s\n", sdate);
#endif

   /* return o_id's in district id order */

   for (i = 0; i < NDISTS; i++)
      del_o_id[i] = 0;
   for (i = 0; i < rpc; i++)
      del_o_id[dctx->d_id[i] - 1] = dctx->del_o_id[i];
#endif

   return (SVC_NOERROR);

}



void tkvcddone ()

{

   if (dctx)
   {
#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
     OCIHandleFree((dvoid *)dctx->curd0,OCI_HTYPE_STMT);
#endif
#ifdef PLSQLDEL
     OCIHandleFree((dvoid *)dctx->curd0,OCI_HTYPE_STMT);
#else
     /* Again the above will cause a problem if both PSQLDEL and ISO are
        defined - VMM 12/30/97 */
     OCIHandleFree((dvoid *)dctx->curd1,OCI_HTYPE_STMT);
     OCIHandleFree((dvoid *)dctx->curd2,OCI_HTYPE_STMT);
     OCIHandleFree((dvoid *)dctx->curd3,OCI_HTYPE_STMT);
     OCIHandleFree((dvoid *)dctx->curd4,OCI_HTYPE_STMT);
```

```
     OCIHandleFree((dvoid *)dctx->curd5,OCI_HTYPE_STMT);
     OCIHandleFree((dvoid *)dctx->curd6,OCI_HTYPE_STMT);
#endif
     free (dctx);
   }

}
```

plord.c

```
// plord.c
// Copyright (c) 1998 1999 Unisys Corporation
// Copyright (c) 1995  Oracle Corp, Redwood Shores, CA

#include "tpcc.h"

#ifdef TUX
#include <userlog.h>
#endif

#ifdef PLSQLORD
#define SQLTXT "BEGIN orderstatus.getstatus (: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;"
#else

#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_last, c_d_id, c_w_id, c_first"

#define SQLCUR1 "SELECT c_id, c_balance, c_first, c_middle, c_last, \
                 o_id, o_entry_d, o_carrier_id, o_ol_cnt \
                 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_c_id, o_d_id, o_w_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,c_id \
                 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_c_id, o_d_id, o_w_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"

#define SQLCUR4 "SELECT count(c_last) FROM customer \
                 WHERE c_d_id = :d_id AND c_w_id = :w_id AND c_last =
:c_last "
#endif



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;

   ub4 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;

   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;

   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;

   OCIStmt *curo0;
   OCIBind *w_id_bp0;
   OCIBind *d_id_bp0;
   OCIBind *c_id_bp;
   OCIBind *c_last_bp;
#ifdef PLSQLORD
```

```
    OCIBind *byln_bp;
    OCIBind *c_first_bp;
    OCIBind *c_middle_bp;
    OCIBind *c_balance_bp;
    OCIBind *o_entry_d_bp;
    OCIBind *o_cr_id_bp;
    OCIBind *o_ol_cnt_bp;
    OCIBind *ol_i_id_bp;
    OCIBind *ol_supply_w_id_bp;
    OCIBind *ol_quantity_bp;
    OCIBind *ol_amount_bp;
    OCIBind *ol_d_base_bp;
    ub4 ol_i_id_cnt;
    ub4 ol_sup_cnt;
    ub4 ol_qty_cnt;
    ub4 ol_amt_cnt;
    ub4 ol_del_d_cnt;
#else
    OCIStmt *curo1;
    OCIStmt *curo2;
    OCIStmt *curo3;
    OCIStmt *curo4;
    OCIBind *w_id_bp2;
    OCIBind *w_id_bp3;
    OCIBind *w_id_bp4;
    OCIBind *d_id_bp2;
    OCIBind *d_id_bp3;
    OCIBind *d_id_bp4;
    OCIBind *c_last_bp4;
    OCIBind *o_id_bp;
    OCIBind *c_rowid_bp;
    OCIDefine *c_rowid_dp;
    OCIDefine *c_last_dp;
    OCIDefine *c_last_dp1;
    OCIDefine *c_id_dp;
    OCIDefine *c_id_dp1;
    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;
    OCIDefine *c_count_dp;
    OCIRowid *c_rowid_ptr[100];
    OCIRowid *middle_cust;
    int cs;
    int cust_idx;
    int norow;
    int rcount;
    int somerows;
#endif
};

typedef struct ordctx ordctx;

struct defctx
{
 boolean reexec;
 ub4 count;
};
typedef struct defctx defctx;

ordctx *octx;

defctx cbctx;

OCIDate ord_null_date;

#ifndef PLSQLORD
sb4 rid_data(dvoid *ctxp, OCIDefine *dp, ub4 iter,
               dvoid **bufpp, ub4 **alenp, ub1 *piecep,
               dvoid **indpp, ub2 **rcodepp)
{
  ub4 i;
  if (((defctx*)ctxp)->reexec)/* if this is the second execute - use entry
0 */
  {
     i = 0;
     ((defctx*)ctxp)->count--;   /* count down */
  }
  else
     i = iter;
  *bufpp = octx->c_rowid_ptr[i];
  *indpp= &octx->c_rowid_ind[i];
  *alenp= &octx->c_rowid_len[i];
  *rcodepp = &octx->c_rowid_rcode[i];
  *piecep =OCI_ONE_PIECE;
  return (OCI_CONTINUE);
}
#endif

tkvcoinit ()

{

   int i;
   text stmbuf[SQL_BUF_SIZE];

   octx = (ordctx *) malloc (sizeof(ordctx));
   memset(octx,(char)0,sizeof(ordctx));
#ifndef PLSQLORD
   octx->cs = 1;
   octx->norow = 0;
   octx->somerows = 10;
/* 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));
  }
#endif


   OCIERROR(errhp,
      OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo0,OCI_HTYPE_STMT,0,(dvoid**)0));
   OCIERROR(errhp,
      OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo0,OCI_HTYPE_STMT,0,(dvoid**)0));
#ifndef PLSQLORD
   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));
   OCIERROR(errhp,
      OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo4,OCI_HTYPE_STMT,0,(dvoid**)0));
#endif

#ifdef PLSQLORD
   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));

   sprintf((char *) stmbuf, SQLCUR4);
   OCIERROR(errhp,
      OCIStmtPrepare(octx->curo4,errhp,stmbuf,strlen((char *)stmbuf),
                  OCI_NTV_SYNTAX,OCI_DEFAULT));
   OCIERROR(errhp,
      OCIAttrSet(octx->curo4,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 PLSQLORD

   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_id_bp , errhp,":c_id",ADR(c_id),
          SIZ(c_id),SQLT_INT);
   OCIBND(octx->curo0, octx->byln_bp , errhp,":byln",ADR(bylastname),
          SIZ(int),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 (float),SQLT_FLT);
   OCIBND(octx->curo0, octx->c_id_bp , errhp,":o_id",ADR(o_id),
          SIZ(int),SQLT_INT);
   OCIBND(octx->curo0, octx->o_entry_d_bp , errhp,":o_entry_d",o_entry_d,
          SIZ(o_entry_d),SQLT_STR);
   OCIBND(octx->curo0, octx->o_cr_id_bp ,
errhp,":o_cr_id",ADR(o_carrier_id),
          SIZ(int), SQLT_INT);
   OCIBND(octx->curo0, octx->o_ol_cnt_bp ,
errhp,":o_ol_cnt",ADR(o_ol_cnt),
          SIZ(int),SQLT_INT);

   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,&octx->ol_i_id_cnt);
   OCIBNDRAA(octx->curo0,octx->ol_supply_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,&octx->ol_sup_cnt);
   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,&octx->ol_qty_cnt);
   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,
             &octx->ol_amt_cnt);
   OCIBNDRAA(octx->curo0,octx->ol_d_base_bp,errhp,":ol_d_d",ol_d_base,
             SIZ(OCIDate),SQLT_ODT,octx->ol_delivery_d_ind,
             octx->ol_delivery_d_len, octx->ol_delivery_d_rcode,NITEMS,
             &octx->ol_del_d_cnt);

#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);
   OCIDFNDYN(octx->curo0,octx->c_rowid_dp,errhp,1,octx->c_rowid_ptr,
          SIZ(OCIRowid*), SQLT_RDD, octx->c_rowid_ind, &cbctx,rid_data);

   OCIBND(octx->curo1,octx->c_rowid_bp,errhp,":cust_rowid",
          &octx->middle_cust,sizeof(octx->middle_cust),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)-1,
           SQLT_CHR);
   OCIDEF(octx->curo1,octx->c_middle_dp1,errhp,4,c_middle,
           SIZ(c_middle)-1,SQLT_AFC);
   OCIDEF(octx->curo1,octx->c_last_dp1,errhp,5,c_last,SIZ(c_last)-1,
           SQLT_CHR);
   OCIDEF(octx->curo1,octx->o_id_dp1,errhp,6,ADR(o_id),SIZ(int),SQLT_INT);
   OCIDEF(octx->curo1,octx->o_entry_d_dp1,errhp,7,
           &o_entry_d_base,SIZ(OCIDate),SQLT_ODT);
   OCIDEF(octx->curo1,octx->o_cr_id_dp1,errhp,8,ADR(o_carrier_id),
           SIZ(int),SQLT_INT);
   OCIDEF(octx->curo1,octx->o_ol_cnt_dp1,errhp,9,ADR(o_ol_cnt),
           SIZ(int),SQLT_INT);

/* 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)-1,
           SQLT_CHR);
   OCIDEF(octx->curo2,octx->c_middle_dp2,errhp,3,c_middle,
           SIZ(c_middle)-1,SQLT_AFC);
   OCIDEF(octx->curo2,octx->c_last_dp,errhp,4,c_last,SIZ(c_last)-1,
SQLT_CHR);
   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(OCIDate),SQLT_ODT);
   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);
   OCIDEF(octx->curo2,octx->c_id_dp1,errhp,9,ADR(c_id),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,SIZ(OCIDate),
            SQLT_ODT, octx->ol_delivery_d_ind,octx->ol_delivery_d_len,
            octx->ol_delivery_d_rcode);

   OCIBND(octx->curo4,octx-
>w_id_bp4,errhp,":w_id",ADR(w_id),SIZ(int),SQLT_INT);
   OCIBND(octx->curo4,octx-
>d_id_bp4,errhp,":d_id",ADR(d_id),SIZ(int),SQLT_INT);
   OCIBND(octx->curo4,octx->c_last_bp4,errhp,":c_last",c_last,SIZ(c_last),
          SQLT_STR);
   OCIDEF(octx->curo4,octx->c_count_dp,errhp,1,ADR(octx->rcount),SIZ(int),
          SQLT_INT);
```

```
#endif

   OCIDateSetDate(&ord_null_date,(sb2)1811,(ub1)1,(ub1)1);
   OCIDateSetTime(&ord_null_date,(ub1)0,(ub1)0,(ub1)0);
   return (0);

};   // tkvcoinit



tkvco ()

{

   int i;
   int rcount;

   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(OCIDate);
   }
   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 PLSQLORD
   octx->ol_i_id_cnt = 0;
   octx->ol_sup_cnt = 0;
   octx->ol_qty_cnt = 0;
   octx->ol_amt_cnt = 0;
   octx->ol_del_d_cnt = 0;
   OCIERROR(errhp,
      OCIStmtExecute(tpcsvc,octx->curo0,errhp,1,0,0,0,OCI_DEFAULT));
#else
retry:
   if(bylastname)
   {
     cbctx.reexec = FALSE;
     execstatus=OCIStmtExecute(tpcsvc,octx-
>curo0,errhp,100,0,0,0,OCI_DEFAULT);
     /* will get OCI_NO_DATA if <100 found */
     if ((execstatus != OCI_NO_DATA) && (execstatus != OCI_SUCCESS))
     {
       errcode=OCIERROR(errhp, execstatus);
       if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR)
                 || (errcode == SNAPSHOT_TOO_OLD))
       {
         OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
         retries++;
         goto retry;
       } else {
         return(SVCERR_OCI);
       }
     }
     if (execstatus == OCI_NO_DATA) /* there are no more rows */
     {
      /* get rowcount, find middle one */
       OCIAttrGet(octx-
>curo0,OCI_HTYPE_STMT,&rcount,NULL,OCI_ATTR_ROW_COUNT,errhp);

       if (rcount < 1)
       {
          OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
          return(SVCERR_NOCUSTOMER);
       };
       octx->cust_idx=(rcount-1)/2 ;
     }
     else
     {
       /* count the number of rows */
       execstatus=OCIStmtExecute(tpcsvc,octx-
>curo4,errhp,1,0,0,0,OCI_DEFAULT);
       if ((execstatus != OCI_NO_DATA) && (execstatus != OCI_SUCCESS))
       {
       if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR)
                 || (errcode == SNAPSHOT_TOO_OLD))
         {
          OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
          retries++;
          goto retry;
         } else {
          return(SVCERR_OCI);
         }
       }
       if (octx->rcount+1 < 200 )
          octx->cust_idx=(octx->rcount-1)/2 ;
       else            /* */
       {
          cbctx.reexec = TRUE;
          cbctx.count = (octx->rcount + 1)/2 ;
          execstatus=OCIStmtExecute(tpcsvc,octx->curo0,errhp,cbctx.count,
                                    0,0,0,OCI_DEFAULT);
          /* will get OCI_NO_DATA if <100 found */
          if (cbctx.count > 0)
          {
             userlog ("did not get all rows ");
             return(SVCERR_OCI);
          }

          if ((execstatus != OCI_NO_DATA) && (execstatus != OCI_SUCCESS))
          {
            errcode=OCIERROR(errhp, execstatus);
       if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR)
                 || (errcode == SNAPSHOT_TOO_OLD))
            {
              OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
              retries++;
              goto retry;
             } else {
             return(SVCERR_OCI);
             }
```

```
          }
          octx->cust_idx=0 ;
        }
     }

     octx->middle_cust = octx->c_rowid_ptr[octx->cust_idx];
     execstatus=OCIStmtExecute(tpcsvc,octx-
>curo1,errhp,1,0,0,0,OCI_DEFAULT);
     if (execstatus != OCI_SUCCESS)
     {
        errcode=OCIERROR(errhp,execstatus);
        OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
        if (errcode == OCI_NO_DATA)
           return(SVCERR_NOCUSTOMER);
       if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR)
                 || (errcode == SNAPSHOT_TOO_OLD))
        {
           retries++;
           goto retry;
           } else {
           return(SVCERR_OCI);
        }
      }
    }
    else
    {
      execstatus=OCIStmtExecute(tpcsvc,octx-
>curo2,errhp,1,0,0,0,OCI_DEFAULT);
      if (execstatus != OCI_SUCCESS)
      {
        errcode=OCIERROR(errhp,execstatus);
        OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
        if (errcode == OCI_NO_DATA)
           return(SVCERR_NOCUSTOMER);
       if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR)
                 || (errcode == SNAPSHOT_TOO_OLD))
         {
          retries++;
          goto retry;
        }
        else
        {
          return(SVCERR_OCI);
        }
      }
    }
    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 )
    {
      errcode=OCIERROR(errhp,execstatus);
      OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
       if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR)
                 || (errcode == SNAPSHOT_TOO_OLD))
      {
        retries++;
        goto retry;
      }
      else
      {
        return(SVCERR_OCI);
      }

    }
#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 */
         memcpy(&ol_delivery_d[i],&ord_null_date,sizeof(OCIDate));
      else
         memcpy(&ol_delivery_d[i],&ol_d_base[i],sizeof(OCIDate));
   };

   return(SVC_NOERROR);

};   // tkvco

void tkvcodone ()

{
   if (octx)
      free (octx);
};   // tpvcodone
```

plsto.c

```
// plsto.c
// Copyright (c) 1998 1999 Unisys Corporation
// Copyright (c) 1995  Oracle Corp, Redwood Shores, CA

#include "tpcc.h"

#ifdef PLSQLSTO
#define SQLTXT "BEGIN stocklevel.getstocklevel (:w_id, :d_id, :threshold,
\
   :low_stock); END;"
#else

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

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

```
#ifdef PLSQLSTO
  OCIBind *low_stock_bp;
#else
  OCIDefine *low_stock_bp;
#endif
  int norow;
};

typedef struct stoctx stoctx;

stoctx *sctx;


tkvcsinit ()

{

   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, SQLTXT);
   OCIERROR(errhp,OCIStmtPrepare(sctx->curs,errhp,stmbuf,strlen((char
*)stmbuf),
                 OCI_NTV_SYNTAX,OCI_DEFAULT));
#ifndef PLSQLSTO
   OCIERROR(errhp,
      OCIAttrSet(sctx->curs,OCI_HTYPE_STMT,(dvoid*)&sctx->norow,0,
                 OCI_ATTR_PREFETCH_ROWS,errhp));
#endif

   /* 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);
#ifdef PLSQLSTO
   OCIBND(sctx->curs,sctx->low_stock_bp,errhp,":low_stock" ,
ADR(low_stock),
      sizeof(int), SQLT_INT);
#else
   OCIDEFINE(sctx->curs,sctx->low_stock_bp,errhp, 1, ADR(low_stock),
      sizeof(int), SQLT_INT);
#endif

   return (0);

}



tkvcs ()
{

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

  return(SVC_NOERROR);
}



void tkvcsdone ()

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

tkvcpnew.sql

```
-- New Order Anonymous block

  DECLARE
      idx                       BINARY_INTEGER;
      dummy_local               BINARY_INTEGER;
      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);
   PROCEDURE u1 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_01,
          DECODE (instr(i_data,'ORIGINAL'), 0, 'G',
                      DECODE(instr(s_data,'ORIGINAL'), 0, 'G', 'B'))
         BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
```

```
:brand_generic;


   END u1;

   PROCEDURE u2 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_02,
          DECODE (instr(i_data,'ORIGINAL'), 0, 'G',
                     DECODE(instr(s_data,'ORIGINAL'), 0, 'G', 'B'))
        BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                          :brand_generic;
   END u2;

   PROCEDURE u3 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_03,
          DECODE (instr(i_data,'ORIGINAL'), 0, 'G',
                     DECODE(instr(s_data,'ORIGINAL'), 0, 'G', 'B'))
        BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                          :brand_generic;
   END u3;

   PROCEDURE u4 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_04,
          DECODE (instr(i_data,'ORIGINAL'), 0, 'G',
                     DECODE(instr(s_data,'ORIGINAL'), 0, 'G', 'B'))
        BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                          :brand_generic;
   END u4;

   PROCEDURE u5 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_05,
          DECODE (instr(i_data,'ORIGINAL'), 0, 'G',
                     DECODE(instr(s_data,'ORIGINAL'), 0, 'G', 'B'))
        BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                          :brand_generic;
   END u5;

   PROCEDURE u6 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_06,
          DECODE (instr(i_data,'ORIGINAL'), 0, 'G',
                     DECODE(instr(s_data,'ORIGINAL'), 0, 'G', 'B'))
        BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                          :brand_generic;
   END u6;

   PROCEDURE u7 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_07,
          DECODE (instr(i_data,'ORIGINAL'), 0, 'G',
                     DECODE(instr(s_data,'ORIGINAL'), 0, 'G', 'B'))
        BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                          :brand_generic;
   END u7;

   PROCEDURE u8 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
```

```
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_08,
          DECODE (instr(i_data,'ORIGINAL'), 0, 'G',
                  DECODE(instr(s_data,'ORIGINAL'), 0, 'G', 'B'))
        BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                          :brand_generic;
   END u8;

   PROCEDURE u9 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_09,
          DECODE (instr(i_data,'ORIGINAL'), 0, 'G',
                  DECODE(instr(s_data,'ORIGINAL'), 0, 'G', 'B'))
        BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                          :brand_generic;
   END u9;

   PROCEDURE u10 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_10,
          DECODE (instr(i_data,'ORIGINAL'), 0, 'G',
                  DECODE(instr(s_data,'ORIGINAL'), 0, 'G', 'B'))
        BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                          :brand_generic;
   END u10;

   PROCEDURE fix_items IS
     rows_lost                    BINARY_INTEGER;
     max_index                    BINARY_INTEGER;
     temp_index                   BINARY_INTEGER;
   BEGIN
-- gotta shift price, name, s_quantity, brand_generic, s_dist, ol_amount
        idx := 1;

-- found 0 bad rows
        rows_lost := 0;
-- so many rows in out array to begin with
       max_index := sql%rowcount;

        WHILE (max_index != :o_ol_cnt) LOOP

-- find item where item ids dont match
         WHILE (idx <= sql%rowcount AND
                       sql%bulk_rowcount(idx + rows_lost) = 1)
          LOOP
            idx := idx + 1;
          END LOOP;

-- shift the items please
          temp_index := max_index;
          WHILE (temp_index >= idx + rows_lost) LOOP
             :i_price(temp_index + 1)       :=  :i_price(temp_index);
             :i_name(temp_index + 1)        :=  :i_name(temp_index);
             :s_quantity(temp_index + 1)    :=  :s_quantity(temp_index);
             initnew.s_dist(temp_index + 1) :=
initnew.s_dist(temp_index);
            :brand_generic(temp_index + 1) :=
:brand_generic(temp_index);
            temp_index := temp_index - 1;
          END LOOP;

--  values for the non-existent items if not at end
           IF (idx + rows_lost <= :o_ol_cnt) THEN
            :i_price(idx + rows_lost)     :=   0;
            :i_name(idx + rows_lost)      :=   NULL;
            :s_quantity(idx + rows_lost)  :=   0;
            initnew.s_dist(idx + rows_lost) := NULL;
            :brand_generic(idx + rows_lost) := NULL;

--  one more bad row
            rows_lost := rows_lost + 1;
            max_index := max_index + 1;
           END IF;

        END LOOP;
   END fix_items;

   BEGIN
      LOOP BEGIN
         UPDATE district SET d_next_o_id = d_next_o_id + 1
          WHERE d_id = :d_id AND  d_w_id = :w_id
          RETURNING d_tax, d_next_o_id-1
          INTO :d_tax, :o_id;

         SELECT c_discount, c_last, c_credit, w_tax
          INTO :c_discount, :c_last, :c_credit , :w_tax
          FROM customer , warehouse
          WHERE c_id = :c_id AND c_d_id = :d_id AND c_w_id = :w_id
          AND  w_id = :w_id;


         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_d_id, o_w_id, o_c_id, o_entry_d,
                               o_carrier_id, o_ol_cnt, o_all_local)
```

```
          VALUES (:o_id, :d_id, :w_id, :c_id,
                  :cr_date, 11, :o_ol_cnt, :o_all_local);

         -- copying :d_id in local variable is important - lots of instr.
         dummy_local :=  :d_id;

         IF (dummy_local = 1) THEN u1; END IF;

         IF (dummy_local = 2) THEN u2; END IF;

         IF (dummy_local = 3) THEN u3; END IF;

         IF (dummy_local = 4) THEN u4; END IF;

         IF (dummy_local = 5) THEN u5; END IF;

         IF (dummy_local = 6) THEN u6; END IF;

         IF (dummy_local = 7) THEN u7; END IF;

         IF (dummy_local = 8) THEN u8; END IF;

         IF (dummy_local = 9) THEN u9; END IF;

         IF (dummy_local = 10) THEN u10; END IF;

-- cache the no of rows processed
         dummy_local := sql%rowcount;


-- fix the rows if necessary
         IF (dummy_local != :o_ol_cnt )  THEN fix_items; END IF;

-- calculate ol_amount

        FOR idx IN 1 ..:o_ol_cnt LOOP
         :ol_amount(idx):=:ol_quantity(idx)*:i_price(idx);
        END LOOP;

        FORALL idx IN 1..:o_ol_cnt
         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)
         VALUES (:o_id, :d_id, :w_id, initnew.idx1arr(idx),
initnew.nulldate,
                 :ol_i_id(idx), :ol_supply_w_id(idx),
                :ol_quantity(idx), :ol_amount(idx), initnew.s_dist(idx));

       IF (dummy_local != :o_ol_cnt) THEN
          :o_ol_cnt := dummy_local;
          ROLLBACK;
        END IF;

        EXIT;

        EXCEPTION
           WHEN not_serializable OR deadlock OR snapshot_too_old THEN
               ROLLBACK;
               :retry := :retry + 1;
         END;
      END LOOP;
   END;
```

paynz.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 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 initpay.ware_name, :w_street_1, :w_street_2, :w_city,
                   :w_state, :w_zip;

           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 initpay.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;
            IF SQL%NOTFOUND THEN
              raise NO_DATA_FOUND;
            END IF;


         :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    c_id = :c_id AND c_d_id = :c_d_id AND
                         c_w_id = :c_w_id
--
--              WHERE rowid = initpay.cust_rowid
--
```

```
            RETURNING substr(c_data,1, 200)
            INTO :c_data;
         IF SQL%NOTFOUND THEN
            raise NO_DATA_FOUND;
         END IF;

      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
initpay.dist_name,:d_street_1,:d_street_2,:d_city,:d_state,
                :d_zip;
       IF SQL%NOTFOUND THEN
         raise NO_DATA_FOUND;
       END IF;


      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, initpay.ware_name || '    ' || initpay.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;
```

payz.sql

```
DECLARE /* payz */
      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 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 initpay.ware_name,
                   :w_street_1, :w_street_2, :w_city, :w_state, :w_zip;


      SELECT rowid
       BULK COLLECT INTO initpay.row_id
       FROM customer
       WHERE c_d_id = :c_d_id AND c_w_id = :c_w_id AND c_last = :c_last
       ORDER BY c_last, c_d_id, c_w_id, c_first;

     initpay.c_num := sql%rowcount;
     initpay.cust_rowid := initpay.row_id((initpay.c_num) / 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 = initpay.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 = initpay.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 initpay.dist_name, :d_street_1, :d_street_2, :d_city,
              :d_state, :d_zip;

         IF  SQL%NOTFOUND
         THEN
            raise NO_DATA_FOUND;
         END IF;

      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, initpay.ware_name || '    ' || initpay.dist_name);

      EXIT;
```

```
      EXCEPTION
         WHEN not_serializable OR deadlock OR snapshot_too_old THEN
            ROLLBACK;
            :retry := :retry + 1;
      END;

   END LOOP;
END;
```

Appendix B - Database Design

Database Creation Scripts

benchsetup.sh

```
#!/bin/ksh

# benchsetup 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                     OPEN SYSTEMS PERFORMANCE GROUP                  |
#                          All Rights Reserved                        |
#=====================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=====================================================================
#

BUILD_HOME=/oracle/8i/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
LOAD_SCRIPTS=${BUILD_HOME}
OUTDIR=${BUILD_HOME}/outdir
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

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

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



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

${LOAD_SCRIPTS}/benchdb.sh > ${OUTDIR}/benchdb.out 2>&1
  echo "Switching Logs ..."
  ${TPCC_UTILS}/switchlog.sh >> ${OUTDIR}/switchlog.out 2>&1


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

${LOAD_SCRIPTS}/create_tmp.sh > ${OUTDIR}/creatmp.out 2>&1

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

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

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

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

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

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

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

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

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

wait

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

echo "Switching Logs ..."
${TPCC_UTILS}/switchlog.sh >> ${OUTDIR}/switchlog.out 2>&1

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

#${LOAD_SCRIPTS}/load_hist.sh > ${OUTDIR}/load_hist.out 2>&1

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

echo "Switching Logs ..."
${TPCC_UTILS}/switchlog.sh >> ${OUTDIR}/switchlog.out 2>&1
```

```
${LOAD_SCRIPTS}/load_ware.sh > ${OUTDIR}/load_ware.out 2>&1

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

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

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

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

wait

echo "Switching Logs ..."
${TPCC_UTILS}/switchlog.sh >> ${OUTDIR}/switchlog.out 2>&1


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

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

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

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

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

${LOAD_SCRIPTS}/create_icust2.sh > ${OUTDIR}/create_icust2.out 2>&1

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

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

${LOAD_SCRIPTS}/create_iordr2.sh > ${OUTDIR}/create_iordr2.out 2>&1

echo "No need to create inord"
#${LOAD_SCRIPTS}/create_inord.sh > ${OUTDIR}/create_inord.out 2>&1

echo "No need to create iordl"
#${LOAD_SCRIPTS}/create_iordl.sh > ${OUTDIR}/create_iordl.out 2>&1

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


sqlplus tpcc/tpcc @$TPCC_SQL/tpcc_ana > ${OUTDIR}/tpcc_ana.out 2>&1

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

${LOAD_SCRIPTS}/tpcc_stored_proc.sh > ${OUTDIR}/tpcc_stored_prod.out 2>&1
$TPCC_UTILS/create_cache_views.sh > ${OUTDIR}/create_cache_views.out 2>&1

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

${LOAD_SCRIPTS}/alter.sh > ${OUTDIR}/alter.out 2>&1
${TPCC_UTILS}/dml.sh > ${OUTDIR}/dml.out 2>&1

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

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

benchdb.sh

```
#
# benchdb.sh 8030100 98/7/7 15:45 vmakhija
# Copyr (c) 1998 Oracle
#

#
#=====================================================================+
#        Copyright (c) 1997  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
#=====================================================================
#


BENCH_HOME=$ORACLE_HOME/bench/tpc
TPCC_ADMIN=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/admin

while [ "$#" != "0" ]
do
   case $1 in
       -n) shift
           NO_CREATE="y"
           ;;
       -c) shift
           NO_CAT="y"
           ;;
        *) echo "Bad arg: $1"
           exit 1;
           ;;
   esac
done


#
# Create database if NO_CREATE unset
#

if [ "$NO_CREATE" = "" ]
```

```
then
svrmgrl <<!
        set echo on
        connect internal
        startup  pfile=$TPCC_ADMIN/p_create.ora  nomount
        create database tpcc controlfile reuse maxdatafiles 960
            datafile '?/dbs/tpcc_disks/sys1'  size 3000M reuse
            logfile  '?/dbs/tpcc_disks/log1' size 15000M reuse,
                     '?/dbs/tpcc_disks/log2' size 15000M reuse;
        exit
!

#
# Create more rollback segments
#

svrmgrl <<!
   connect internal
   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);
   shutdown;
   exit;
!
fi

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

svrmgrl <<!
        connect internal
        startup pfile=$TPCC_ADMIN/p_build.ora;
        exit;
!

#
#       Add tablespaces in parallel
#
addts.sh stock     ?/dbs/tpcc_disks/stok01    2750M 50M &
addts.sh cust      ?/dbs/tpcc_disks/cust01     1850M 50M &
addts.sh ordl      ?/dbs/tpcc_disks/ordl01     2650M 50M &
addts.sh misc      ?/dbs/tpcc_disks/misc01     250M 50M &
addts.sh ordr      ?/dbs/tpcc_disks/ordr01     250M 50M &
addts.sh nord      ?/dbs/tpcc_disks/nord01    1200M 50M &
addts.sh icust2    ?/dbs/tpcc_disks/icust2_01  600M 50M &
addts.sh iord1     ?/dbs/tpcc_disks/iord1_01    250M 50M &
addts.sh iord2     ?/dbs/tpcc_disks/iord2_01    200M 50M &
addts.sh unused    ?/dbs/tpcc_disks/unused01  1200M 50M &
addts.sh hist      ?/dbs/tpcc_disks/hist01    1250M 50M &
addts.sh stats     ?/dbs/tpcc_disks/stat01       1200M  5M &
addroll.sh         ?/dbs/tpcc_disks/roll01       3500M &
#addts.sh stock_min ?/dbs/tpcc_disks/stock_min  100M 50M &

wait

#
#      Add datafiles to tablespaces in parallel
#
```

```
addfile.sh stock  ?/dbs/tpcc_disks/stok02 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok03 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok04 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok05 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok06 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok07 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok08 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok09 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok10 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok11 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok12 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok13 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok14 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok15 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok16 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok17 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok18 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok19 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok20 2750M &
wait
addfile.sh stock  ?/dbs/tpcc_disks/stok21 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok22 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok23 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok24 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok25 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok26 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok27 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok28 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok29 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok30 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok31 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok32 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok33 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok34 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok35 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok36 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok37 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok38 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok39 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok40 2750M &
wait
addfile.sh stock  ?/dbs/tpcc_disks/stok41 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok42 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok43 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok44 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok45 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok46 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok47 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok48 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok49 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok50 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok51 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok52 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok53 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok54 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok55 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok56 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok57 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok58 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok59 2750M &
addfile.sh stock  ?/dbs/tpcc_disks/stok60 2750M &
wait
addfile.sh cust  ?/dbs/tpcc_disks/cust02 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust03 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust04 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust05 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust06 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust07 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust08 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust09 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust10 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust11 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust12 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust13 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust14 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust15 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust16 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust17 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust18 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust19 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust20 1850M &
wait
addfile.sh cust  ?/dbs/tpcc_disks/cust21 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust22 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust23 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust24 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust25 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust26 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust27 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust28 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust29 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust30 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust31 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust32 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust33 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust34 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust35 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust36 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust37 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust38 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust39 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust40 1850M &
wait
addfile.sh cust  ?/dbs/tpcc_disks/cust41 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust42 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust43 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust44 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust45 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust46 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust47 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust48 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust49 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust50 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust51 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust52 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust53 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust54 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust55 1850M &
```

```
addfile.sh cust  ?/dbs/tpcc_disks/cust56 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust57 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust58 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust59 1850M &
addfile.sh cust  ?/dbs/tpcc_disks/cust60 1850M &
wait
addfile.sh ordl  ?/dbs/tpcc_disks/ordl02 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl03 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl04 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl05 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl06 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl07 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl08 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl09 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl10 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl11 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl12 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl13 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl14 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl15 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl16 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl17 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl18 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl19 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl20 2650M &
wait
addfile.sh ordl  ?/dbs/tpcc_disks/ordl21 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl22 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl23 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl24 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl25 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl26 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl27 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl28 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl29 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl30 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl31 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl32 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl33 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl34 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl35 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl36 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl37 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl38 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl39 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl40 2650M &
wait
addfile.sh ordl  ?/dbs/tpcc_disks/ordl41 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl42 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl43 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl44 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl45 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl46 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl47 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl48 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl49 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl50 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl51 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl52 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl53 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl54 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl55 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl56 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl57 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl58 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl59 2650M &
addfile.sh ordl  ?/dbs/tpcc_disks/ordl60 2650M &
wait
addfile.sh misc  ?/dbs/tpcc_disks/misc02 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc03 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc04 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc05 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc06 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc07 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc08 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc09 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc10 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc11 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc12 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc13 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc14 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc15 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc16 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc17 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc18 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc19 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc20 250M &
wait
addfile.sh misc  ?/dbs/tpcc_disks/misc21 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc22 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc23 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc24 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc25 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc26 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc27 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc28 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc29 250M &
addfile.sh misc  ?/dbs/tpcc_disks/misc30 250M &
wait
addfile.sh ordr  ?/dbs/tpcc_disks/ordr02 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr03 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr04 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr05 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr06 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr07 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr08 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr09 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr10 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr11 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr12 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr13 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr14 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr15 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr16 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr17 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr18 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr19 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr20 250M &
```

```
wait
addfile.sh ordr  ?/dbs/tpcc_disks/ordr21 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr22 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr23 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr24 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr25 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr26 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr27 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr28 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr29 250M &
addfile.sh ordr  ?/dbs/tpcc_disks/ordr30 250M &
wait
addfile.sh nord  ?/dbs/tpcc_disks/nord02 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord03 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord04 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord05 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord06 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord07 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord08 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord09 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord10 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord11 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord12 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord13 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord14 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord15 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord16 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord17 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord18 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord19 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord20 1200M &
wait
addfile.sh nord  ?/dbs/tpcc_disks/nord21 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord22 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord23 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord24 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord25 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord26 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord27 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord28 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord29 1200M &
addfile.sh nord  ?/dbs/tpcc_disks/nord30 1200M &
wait
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_02 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_03 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_04 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_05 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_06 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_07 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_08 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_09 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_10 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_11 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_12 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_13 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_14 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_15 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_16 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_17 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_18 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_19 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_20 600M &
wait
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_21 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_22 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_23 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_24 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_25 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_26 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_27 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_28 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_29 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_30 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_31 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_32 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_33 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_34 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_35 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_36 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_37 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_38 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_39 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_40 600M &
wait
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_41 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_42 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_43 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_44 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_45 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_46 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_47 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_48 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_49 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_50 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_51 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_52 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_53 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_54 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_55 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_56 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_57 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_58 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_59 600M &
addfile.sh icust2  ?/dbs/tpcc_disks/icust2_60 600M &
wait
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_02 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_03 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_04 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_05 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_06 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_07 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_08 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_09 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_10 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_11 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_12 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_13 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_14 250M &
```

```
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_15 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_16 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_17 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_18 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_19 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_20 250M &
wait
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_21 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_22 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_23 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_24 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_25 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_26 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_27 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_28 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_29 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_30 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_31 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_32 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_33 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_34 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_35 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_36 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_37 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_38 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_39 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_40 250M &
wait
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_41 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_42 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_43 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_44 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_45 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_46 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_47 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_48 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_49 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_50 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_51 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_52 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_53 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_54 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_55 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_56 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_57 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_58 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_59 250M &
addfile.sh iord1  ?/dbs/tpcc_disks/iord1_60 250M &
wait
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_02 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_03 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_04 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_05 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_06 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_07 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_08 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_09 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_10 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_11 200M &
```

```
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_12 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_13 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_14 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_15 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_16 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_17 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_18 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_19 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_20 200M &
wait
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_21 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_22 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_23 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_24 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_25 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_26 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_27 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_28 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_29 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_30 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_31 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_32 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_33 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_34 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_35 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_36 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_37 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_38 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_39 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_40 200M &
wait
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_41 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_42 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_43 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_44 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_45 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_46 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_47 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_48 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_49 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_50 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_51 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_52 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_53 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_54 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_55 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_56 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_57 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_58 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_59 200M &
addfile.sh iord2  ?/dbs/tpcc_disks/iord2_60 200M &
wait
addfile.sh unused  ?/dbs/tpcc_disks/unused02 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused03 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused04 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused05 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused06 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused07 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused08 1200M &
```

```
addfile.sh unused  ?/dbs/tpcc_disks/unused09 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused10 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused11 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused12 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused13 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused14 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused15 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused16 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused17 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused18 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused19 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused20 1200M &
wait
addfile.sh unused  ?/dbs/tpcc_disks/unused21 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused22 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused23 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused24 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused25 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused26 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused27 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused28 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused29 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused30 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused31 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused32 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused33 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused34 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused35 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused36 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused37 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused38 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused39 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused40 1200M &
wait
addfile.sh unused  ?/dbs/tpcc_disks/unused41 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused42 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused43 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused44 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused45 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused46 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused47 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused48 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused49 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused50 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused51 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused52 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused53 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused54 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused55 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused56 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused57 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused58 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused59 1200M &
addfile.sh unused  ?/dbs/tpcc_disks/unused60 1200M &
wait
addfile.sh hist  ?/dbs/tpcc_disks/hist02 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist03 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist04 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist05 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist06 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist07 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist08 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist09 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist10 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist11 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist12 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist13 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist14 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist15 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist16 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist17 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist18 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist19 1250M &
addfile.sh hist  ?/dbs/tpcc_disks/hist20 1250M &
wait


#
# run catalog if NO_CAT unset
#

if [ "$NO_CAT" = "" ]
then
svrmgrl <<!
        set echo off;
        connect sys/change_on_install;
        @?/rdbms/admin/catalog;
        @?/rdbms/admin/catproc;
        @?/rdbms/admin/catparr;
       @?/sqlplus/admin/pupbld;
        exit;
!
fi
```

addfile.sh

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

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

FILE=`basename $2`
```

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

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

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

addts.sh

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

FILE=`basename $2`

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

svrmgrl <<!
   connect internal
   set echo on
   create tablespace $1 datafile '$2' size $3 reuse extent management
local uniform size $4 nologging ;

   exit;
!

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

addroll.sh

```
#!/bin/ksh
#=====================================================================+
#        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                         All Rights Reserved                         |
#=====================================================================+
# FILENAME
#     addts.sh
# DESCRIPTION
#     Add tablespace to database.
# USAGE
#     addts.sh <tablespace> <data file> <size>
#=====================================================================*/
echo 'ORACLE_HOME=' $ORACLE_HOME
echo 'ORACLE_SID=' $ORACLE_SID

FILE=`basename $1`

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

svrmgrl <<!
   connect internal
   create tablespace roll datafile '$1' size $2 reuse extent management
local uniform size 200K nologging ;

   exit;
!

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

alter_temp.sh

```
#!/bin/ksh

# benchsetup 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                         All Rights Reserved                         |
#=====================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
```

```
#====================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
LOAD_SCRIPTS=${BUILD_HOME}
OUTDIR=${BUILD_HOME}/outdir
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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


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

svrmgrl <<!
   connect internal
   alter tablespace temp
      default storage (initial 100M next 100M pctincrease 0 maxextents
700);
   exit;
!
```

switchlog.sh

```
#
# $Header: switchlog.sh 7030100.1 96/05/02 10:20:11 plai Generic<base> $
Copyr (c) 1995 Oracle
#

#====================================================================+
#        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA         |
#                    OPEN SYSTEMS PERFORMANCE GROUP                  |
#                         All Rights Reserved                        |
#====================================================================+
# FILENAME
#     switchlog.sh
# DESCRIPTION
#     Switch to next log file twice.
# USAGE
#     switchlog.sh
#====================================================================*/

svrmgrl lmode=y <<!
   connect internal;
   alter system switch logfile;
   alter system switch logfile;
   exit;
!
```

tpcc_misc.sh

```
#!/bin/ksh

# benchsetup 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle
#
#
#====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA         |
#                    OPEN SYSTEMS PERFORMANCE GROUP                  |
#                         All Rights Reserved                        |
#====================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#====================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
LOAD_SCRIPTS=${BUILD_HOME}
OUTDIR=${BUILD_HOME}/outdir
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

```
$TPCC_UTILS/ext_all.sh > ${OUTDIR}/ext_all.out 2>&1

$TPCC_UTILS/space_init.sh
$TPCC_UTILS/space_get.sh 14400 3500
$TPCC_UTILS/space_rpt.sh ${OUTDIR}/space.rpt

sqlplus system/manager <<!
   alter user tpcc temporary tablespace system;
   quit;
!

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 @$BUILD_SQL/plsql_mon
sqlplus tpcc/tpcc @$BUILD_SQL/cre_tab
```

tpcc_rol.sh

```
#!/bin/ksh
#
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                   OPEN SYSTEMS PERFORMANCE GROUP                |
#                        All Rights Reserved                      |
#=================================================================+
# FILENAME
#      tpccrol.sh
# DESCRIPTION
#      Script file for creating the roll;back segments.
#=================================================================
#

svrmgrl <<!
   connect internal;
   host date;
   set timing on;


               CREATE ROLLBACK SEGMENT t1 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t2 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t3 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t4 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t5 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t6 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t7 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t8 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t9 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t10 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t11 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t12 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t13 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t14 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t15 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t16 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t17 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t18 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t19 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t20 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t21 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t22 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t23 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t24 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t25 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t26 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t27 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t28 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t29 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t30 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t31 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t32 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t33 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t34 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t35 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t36 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t37 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t38 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t39 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t40 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t41 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t42 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t43 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t44 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t45 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t46 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t47 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t48 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t49 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t50 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t51 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t52 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t53 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t54 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t55 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t56 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t57 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t58 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t59 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t60 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t61 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t62 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t63 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t64 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t65 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t66 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t67 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t68 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t69 TABLESPACE roll;
               CREATE ROLLBACK SEGMENT t70 TABLESPACE roll;
```

```
CREATE ROLLBACK SEGMENT t71 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t72 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t73 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t74 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t75 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t76 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t77 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t78 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t79 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t80 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t81 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t82 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t83 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t84 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t85 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t86 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t87 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t88 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t89 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t90 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t91 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t92 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t93 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t94 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t95 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t96 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t97 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t98 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t99 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t100 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t101 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t102 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t103 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t104 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t105 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t106 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t107 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t108 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t109 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t110 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t111 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t112 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t113 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t114 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t115 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t116 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t117 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t118 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t119 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t120 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t121 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t122 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t123 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t124 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t125 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t126 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t127 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t128 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t129 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t130 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t131 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t132 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t133 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t134 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t135 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t136 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t137 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t138 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t139 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t140 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t141 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t142 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t143 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t144 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t145 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t146 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t147 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t148 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t149 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t150 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t151 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t152 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t153 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t154 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t155 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t156 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t157 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t158 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t159 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t160 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t161 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t162 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t163 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t164 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t165 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t166 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t167 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t168 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t169 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t170 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t171 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t172 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t173 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t174 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t175 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t176 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t177 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t178 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t179 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t180 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t181 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t182 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t183 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t184 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t185 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t186 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t187 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t188 TABLESPACE roll;
```

```
CREATE ROLLBACK SEGMENT t189 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t190 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t191 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t192 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t193 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t194 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t195 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t196 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t197 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t198 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t199 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t200 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t201 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t202 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t203 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t204 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t205 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t206 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t207 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t208 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t209 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t210 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t211 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t212 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t213 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t214 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t215 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t216 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t217 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t218 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t219 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t220 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t221 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t222 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t223 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t224 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t225 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t226 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t227 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t228 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t229 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t230 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t231 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t232 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t233 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t234 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t235 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t236 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t237 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t238 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t239 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t240 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t241 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t242 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t243 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t244 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t245 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t246 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t247 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t248 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t249 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t250 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t251 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t252 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t253 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t254 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t255 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t256 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t257 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t258 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t259 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t260 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t261 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t262 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t263 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t264 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t265 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t266 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t267 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t268 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t269 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t270 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t271 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t272 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t273 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t274 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t275 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t276 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t277 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t278 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t279 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t280 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t281 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t282 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t283 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t284 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t285 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t286 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t287 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t288 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t289 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t290 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t291 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t292 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t293 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t294 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t295 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t296 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t297 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t298 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t299 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t300 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t301 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t302 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t303 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t304 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t305 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t306 TABLESPACE roll;
```

```
CREATE ROLLBACK SEGMENT t307 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t308 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t309 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t310 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t311 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t312 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t313 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t314 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t315 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t316 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t317 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t318 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t319 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t320 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t321 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t322 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t323 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t324 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t325 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t326 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t327 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t328 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t329 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t330 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t331 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t332 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t333 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t334 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t335 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t336 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t337 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t338 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t339 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t340 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t341 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t342 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t343 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t344 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t345 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t346 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t347 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t348 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t349 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t350 TABLESPACE roll;
!
```

tpcc_stored_proc.sh

```
#!/bin/ksh

# benchsetup 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
LOAD_SCRIPTS=${BUILD_HOME}
OUTDIR=${BUILD_HOME}/outdir
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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


sqlplus tpcc/tpcc @$TPCC_BLOCKS/views
#sqlplus tpcc/tpcc @$TPCC_BLOCKS/pay
```

Stored Procedures

views.sql

```
create or replace view wh_cust
(w_id, w_tax, c_id, c_d_id, c_w_id, c_discount, c_last, c_credit)
as select w.w_id, w.w_tax,
          c.c_id, c.c_d_id, c.c_w_id, c.c_discount, c.c_last, c.c_credit
   from customer c, warehouse w
   where w.w_id = c.c_w_id
/

create or replace view wh_dist
(w_id, d_id, d_tax, d_next_o_id, w_tax )
```

```
as select w.w_id, d.d_id, d.d_tax, d.d_next_o_id, w.w_tax
   from district d, warehouse w
   where w.w_id = d.d_w_id
/

create  or replace view stock_item
(i_id, s_w_id, i_price, i_name, i_data, s_data, s_quantity,
 s_order_cnt, s_ytd, s_remote_cnt,
 s_dist_01, s_dist_02, s_dist_03, s_dist_04, s_dist_05,
 s_dist_06, s_dist_07, s_dist_08, s_dist_09, s_dist_10)
as
 select i.i_id, s_w_id, i.i_price, i.i_name, i.i_data, s_data, s_quantity,
 s_order_cnt, s_ytd, s_remote_cnt,
 s_dist_01, s_dist_02, s_dist_03, s_dist_04, s_dist_05,
 s_dist_06, s_dist_07, s_dist_08, s_dist_09, s_dist_10
  from stock s, item i
 where i.i_id = s.s_i_id
/
exit
```

initnew.sql

```
-- The initnew package for storing variables used in the
-- New Order anonymous block

CREATE OR REPLACE PACKAGE initnew
AS
 TYPE intarray IS TABLE OF INTEGER index by binary_integer;
 TYPE distarray IS TABLE OF VARCHAR(24) index by binary_integer;
 nulldate       DATE;
 s_dist                distarray;
 idx1arr               intarray;
 s_remote              intarray;
 PROCEDURE new_init(idxarr intarray);
END initnew;
/
show errors;

CREATE OR REPLACE PACKAGE BODY initnew AS
  PROCEDURE new_init (idxarr  intarray)
  IS
  BEGIN
        -- initialize null date
   nulldate := TO_DATE('09-15-1811', 'MM-DD-YYYY');
    idx1arr := idxarr;
  END new_init;
END initnew;
/
show errors
exit
```

initpay.sql

```
CREATE OR REPLACE PACKAGE initpay
AS
 TYPE rowidarray IS TABLE OF ROWID INDEX BY BINARY_INTEGER;
 row_id                rowidarray;
 cust_rowid            ROWID;
 dist_name             VARCHAR2(11);
 ware_name             VARCHAR2(11);
 c_num                 BINARY_INTEGER;
 PROCEDURE pay_init;
END initpay;
/
CREATE OR REPLACE PACKAGE BODY initpay AS
  PROCEDURE pay_init IS
  BEGIN
    NULL;
  END pay_init;
END initpay;
/
exit
```

p_build.ora

```
#
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                  OPEN SYSTEMS PERFORMANCE GROUP                   |
#                       All Rights Reserved                         |
#===================================================================+
# FILENAME
#      create.ora
# DESCRIPTION
#      Oracle parameter file for creating TPC-C database.
#====================================================================
#

compatible                    = 8.1.3.0.0
control_files                 = (/oracle/8i/dbs/tpcc_disks/ctrl1,
                                 /oracle/8i/dbs/tpcc_disks/ctrl2)
sort_area_size                  = 30000000
parallel_max_servers            = 80
recovery_parallelism            = 40
db_name                         = tpcc
db_files                        = 800
db_block_buffers                = 1000000
dml_locks                       = 700
log_buffer                      = 1048576
max_rollback_segments           = 400
processes                       = 300
sessions                        = 400
transactions                    = 400
transactions_per_rollback_segment = 3
rollback_segments               =
(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,
```

```
s14,s15,s16,s17,s18,s19,s20,s21,s22,s23,s24,s25,s26,s27,s28,s29,s30)
shared_pool_size                  = 15000000
cursor_space_for_time             = TRUE
timed_statistics                 = TRUE
_enable_list_io                  = TRUE
disk_asynch_io                   = TRUE
tape_asynch_io                   = TRUE
_lgwr_async_io                   = FALSE
_dbwr_async_io                   = TRUE
```

p_create.ora

```
#
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                         All Rights Reserved                         |
#=====================================================================+
# FILENAME
#      create.ora
# DESCRIPTION
#      Oracle parameter file for creating TPC-C database.
#=====================================================================
#

compatible                    = 8.1.3.0.0
control_files                 = (/oracle/8i/dbs/tpcc_disks/ctrl1,
                                 /oracle/8i/dbs/tpcc_disks/ctrl2)
db_name                        = tpcc
db_files                       = 200
db_block_buffers               = 2000
dml_locks                      = 500
log_buffer                     = 32768
sessions                       = 200
processes                      = 300
transactions                   = 250
disk_asynch_io=TRUE
tape_asynch_io=TRUE
```

finish.sh

```
sqlplus tpcc/tpcc             @sql/tpcc_ana
sqlplus sys/change_on_install @sql/orst_cre
sqlplus sys/change_on_install @sql/c_stat
sqlplus sys/change_on_install @sql/pst_c
sqlplus tpcc/tpcc             @../../blocks/views
sqlplus tpcc/tpcc             @../../blocks/initpay
sqlplus tpcc/tpcc             @../../blocks/tkvcinin
utils/create_cache_views.sh
altundef.sh
tpcc_rol.sh
```

create_cust.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                         All Rights Reserved                         |
#=====================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=====================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @cust
```

create_dist.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
```

```
# Copyright (c) 1998 Oracle Corp.
#
#==============================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA       |
#                    OPEN SYSTEMS PERFORMANCE GROUP                |
#                         All Rights Reserved                      |
#==============================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @dist
```

create_hist.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#==============================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA       |
#                    OPEN SYSTEMS PERFORMANCE GROUP                |
#                         All Rights Reserved                      |
#==============================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @hist
```

create_icust.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#==============================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA       |
#                    OPEN SYSTEMS PERFORMANCE GROUP                |
#                         All Rights Reserved                      |
#==============================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
```

```
#=================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @icust
```

create_icust2.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @icust2
```

create_idist.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
```

```
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @idist
```

create_iitem.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                         All Rights Reserved                         |
#=====================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=====================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @iitem
```

create_iordr.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                         All Rights Reserved                         |
#=====================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=====================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
```

```
MULT=3500

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

sqlplus tpcc/tpcc @iordr
```

create_iordr2.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @iordr2
```

create_istok.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @istok
```

create_item.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                         All Rights Reserved                         |
#=====================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=====================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @item
```

create_iware.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                         All Rights Reserved                         |
#=====================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=====================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @iware
```

create_nord.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
```

```
#                         All Rights Reserved                          |
#======================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#======================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

#change_defspace.sh nord
sqlplus tpcc/tpcc @nord
#change_defspace.sh system
```

create_ordl.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#======================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA           |
#                    OPEN SYSTEMS PERFORMANCE GROUP                    |
#                         All Rights Reserved                          |
#======================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#======================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @ordl
```

create_ordr.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#======================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA           |
#                    OPEN SYSTEMS PERFORMANCE GROUP                    |
#                         All Rights Reserved                          |
#======================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#======================================================================
#
```

```
BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @ordr
```

create_stok.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                    OPEN SYSTEMS PERFORMANCE GROUP                 |
#                         All Rights Reserved                       |
#===================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#====================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @stok
```

create_tmp.sh

```
#!/bin/ksh

#
# create_tmp.sh 99/1/11 02:03 elford
# Copyright (c) 1999 Intel Corp.
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

```
export PATH

sqlplus system/manager @temp
```

create_user.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                    OPEN SYSTEMS PERFORMANCE GROUP                 |
#                         All Rights Reserved                       |
#===================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#====================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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


svrmgrl <<!
rem
rem ================================================================+
rem         Copyright (c) 1997  Oracle Corp, Redwood Shores, CA      |
rem                    OPEN SYSTEMS PERFORMANCE GROUP                |
rem                         All Rights Reserved                      |
rem ================================================================+
rem  FILENAME
rem       tpcc_user.sql
rem  DESCRIPTION
rem       Create user for TPC-C database.
rem ================================================================
rem

rem
rem Create TPCC userid and connect to it.
rem
        connect internal;
        grant connect,resource,unlimited tablespace to tpcc identified by
tpcc;
        alter user tpcc temporary tablespace temp;
        connect tpcc/tpcc;
        exit;
!
```

create_ware.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                    OPEN SYSTEMS PERFORMANCE GROUP                 |
#                         All Rights Reserved                       |
#===================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#====================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
```

```
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @ware

Table Creation Scripts
```

hist.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       hist.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ====================================================================
rem

rem
rem DROP all first
rem
        drop table history;

set timing on

rem
rem HISTORY table
rem

       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,
        h_data          varchar2(24)
        )
        tablespace hist
        initrans 4
        pctfree  5
        storage (freelist groups 43 freelists 9
                 initial 50M next 50M pctincrease 0 minextents 1 maxextents
unlimited buffer_pool recycle);

rem
rem done
rem

        exit;
```

dist.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       dist.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ====================================================================
rem

rem
rem DROP all first
rem
        drop table district;
        drop cluster dcluster including tables;

set timing on

rem
rem DISTRICT table
rem

        create cluster dcluster (
                d_w_id  number(5,0),
                d_id    number(2,0)
        )
        single          table
        hashkeys        35000
        hash is         (d_w_id) * 10 + d_id
        size            1536
        initrans        3
        pctfree         0
        tablespace      misc;

        create table district (
                d_id            number(2,0),
                d_w_id          number(5,0),
```

```
            d_ytd           number,
            d_tax           number(4,4),
            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)
        )
        cluster dcluster (d_w_id, d_id);

rem
rem done
rem

        exit;
```

cust.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

rem
rem CUSTOMER table
rem

        create cluster ccluster (
            c_id         number(5,0),
            c_d_id       number(2,0),
            c_w_id       number(5,0)
        )
        single         table
        hashkeys        105000000
        hash is         (c_id*(35000)+c_w_id*10 + c_d_id -1 )
        size            850
        initrans        3
        pctfree         0
        tablespace      cust
        storage (initial 350M next 300M pctincrease 0 minextents 1
maxextents unlimited);

        create table customer (
        c_id            number(5,0),
        c_d_id          number(2,0),
        c_w_id          number(5,0),
        c_discount      number(4,4),
        c_credit        char(2),
        c_last          varchar2(16),
        c_first         varchar2(16),
        c_credit_lim    number,
        c_balance       number,
        c_ytd_payment   number,
        c_payment_cnt   number,
        c_delivery_cnt  number,
        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_middle        char(2),
        c_data          varchar2(500)
        )
        cluster ccluster (c_id, c_d_id, c_w_id);

rem
rem done
rem

        exit;
```

ware.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       ware.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ====================================================================
rem

rem
rem DROP all first
rem
        drop table warehouse;
```

```
        drop cluster wcluster including tables;

set timing on

rem
rem WAREHOUSE table
rem

        create cluster wcluster (
               w_id           number (5,0)
        )
        single                 table
        hashkeys               3500
        hash is                w_id
        size                   1536
        initrans               3
        pctfree                0
         tablespace     misc;

         create table warehouse (
                 w_id            number(5,0),
                 w_ytd           number,
                 w_tax           number(4,4),
                 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)
         )
        cluster wcluster (w_id);

rem
rem done
rem

         exit;
```

temp.sql

```
drop tablespace temp;
create temporary tablespace temp tempfile
'/oracle/8i/dbs/tpcc_disks/temp01' size 4400M  reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp02' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp03' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp04' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp05' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp06' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp07' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp08' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp09' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp10' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp11' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp12' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp13' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp14' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp15' size
4400M reuse,
                              '/oracle/8i/dbs/tpcc_disks/temp16' size
4400M reuse
extent management local uniform size 100M;
disconnect;
exit;
```

stok.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

rem
rem STOCK table
rem

        create cluster scluster (
           s_i_id       number(6,0),
           s_w_id       number(5,0)
        )
        single         table
         hashkeys       350000000
```

```
        hash is         (s_i_id * 3500 + s_w_id - 1)
        size            350
        initrans        3
        pctfree         0
        tablespace      stock
       storage (initial 350M next 300M pctincrease 0 minextents 1
maxextents unlimited);
        create table stock (
        s_i_id          number(6,0),
        s_w_id          number(5,0),
        s_quantity      number,
        s_ytd           number,
        s_order_cnt     number,
        s_remote_cnt    number,
        s_data          varchar2(50),
        s_dist_01       char(24),
        s_dist_02       char(24),
        s_dist_03       char(24),
        s_dist_04       char(24),
        s_dist_05       char(24),
        s_dist_06       char(24),
        s_dist_07       char(24),
        s_dist_08       char(24),
        s_dist_09       char(24),
        s_dist_10       char(24)
        )
        cluster scluster (s_i_id, s_w_id);

rem
rem done
rem

        exit;
```

ordr.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       ordr.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ====================================================================
rem

rem
rem DROP all first
rem
        drop table orders;

set timing on

rem
rem ORDERS table
rem

       create table orders (
        o_id            number,
        o_w_id          number,
        o_d_id          number,
        o_c_id          number,
        o_carrier_id    number,
        o_ol_cnt        number,
        o_all_local     number,
        o_entry_d       date
        )
        tablespace ordr
        initrans 3
        pctfree  1
        storage ( initial 50M next 50M pctincrease 0 minextents 1
maxextents unlimited freelist groups 13 freelists 22);


rem
rem done
rem

        exit;
```

ordl.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       ordr.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ====================================================================
rem

rem
rem DROP all first
rem
        drop table order_line;

set timing on

rem
rem ORDER_LINE table
rem

       create table order_line (
```

```
        ol_w_id         number,
        ol_d_id         number,
        ol_o_id         number,
        ol_number       number,
        ol_i_id         number,
        ol_delivery_d   date,
        ol_amount       number,
        ol_supply_w_id  number,
        ol_quantity     number,
        ol_dist_info    char(24),
       constraint iordl primary key (ol_w_id, ol_d_id, ol_o_id,
ol_number)
        )
       organization index
       tablespace ordl
        initrans 4
        pctfree  5
        storage ( initial 50M next 50M pctincrease 0 minextents 1
maxextents unlimited freelist groups 43 freelists 9);

rem
rem done
rem

        exit;
```

nord.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       nord.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ====================================================================
rem

rem
rem DROP all first
rem
        drop table new_order;

set timing on

rem
rem NEW_ORDER table
rem

       create table new_order (
                no_w_id         number,
                no_d_id         number,
                no_o_id         number,
               constraint inord primary key (no_w_id, no_d_id, no_o_id)
        )
        organization index
        tablespace nord
        initrans 4
        pctfree  5
       storage ( freelist groups 13 freelists 22);


rem
rem done
rem

        exit;
```

iware.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       iware.sql
rem  DESCRIPTION
rem       Create warehouse index for TPC-C database.
rem ====================================================================
rem

rem
rem DROP all first
rem
        drop index iwarehouse;

set timing on

rem
rem WAREHOUSE index
rem

     create unique index iwarehouse on warehouse (w_id)
        tablespace unused
        initrans 3
       pctfree 1;

rem
rem done
rem

        exit;
```

item.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       item.sql
rem  DESCRIPTION
rem       Create ITEM table for TPC-C database.
rem ================================================================
rem

rem
rem DROP item cluster and table
rem
        drop cluster icluster including tables;
        drop table item;

        set timing on;

rem
rem ITEM table
rem

        create cluster icluster (
        i_id            number (6,0)
        )
       single         table
        hashkeys        100000
        hash is         i_id
        size            120
        initrans        3
        pctfree         0
        tablespace      misc
       storage (buffer_pool keep);

        create table item (
        i_id            number(6,0),
        i_name          varchar2(24),
        i_price         number,
        i_data          varchar2(50),
        i_im_id         number
        )
        cluster icluster(i_id);


rem
rem done
rem

        exit;
```

istok.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       istok.sql
rem  DESCRIPTION
rem       Create stock index for TPC-C database.
rem ================================================================
rem

rem
rem DROP all first
rem
        drop index istock;

set timing on

rem
rem STOCK index
rem

       create unique index istock on stock(s_i_id, s_w_id)
        tablespace unused
        initrans 3
       parallel 10;

rem
rem done
rem

        exit;
```

iord2.sql

```
drop index iorders2;

create unique index iorders2 on orders(o_c_id,o_d_id,o_w_id,o_id)
tablespace iord2
initrans 4
pctfree 25
parallel 20
storage (initial 50M next 50M pctincrease 0 maxextents unlimited
         freelists 22 freelist groups 43
        );
```

iordr.sql

```
 drop index iorders;

 create unique index iorders on orders(o_w_id,o_d_id,o_id)
 tablespace iord1
 initrans 3
 pctfree 1
 parallel 20
 storage (initial 50M next 50M pctincrease 0 maxextents unlimited
          freelists 22 freelist groups 43
         );
exit;
```

iitem.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       item.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ================================================================
rem

rem
rem DROP all first
rem
        drop index iitem;

set timing on

rem
rem ITEM index
rem

       create unique index iitem on item(i_id)
        tablespace misc
        initrans 3
       parallel 10
        pctfree  1;

rem
rem done
rem

        exit;
```

idist.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       dist.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ================================================================
rem

rem
rem DROP all first
rem
        drop index idistrict;

set timing on

rem
rem DISTRICT index
rem

       create unique index idistrict on district(d_w_id, d_id)
        tablespace unused
        initrans 3
        pctfree  5;

rem
rem done
rem

        exit;
```

icust2.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       icust2.sql
rem  DESCRIPTION
rem       Create customer index 2 for TPC-C database.
rem ================================================================
rem
```

```
rem
rem DROP all first
rem
        drop index icustomer2;

set timing on

rem
rem ICUST2 index
rem

       create unique index icustomer2 on customer(c_last, c_d_id, c_w_id,
c_first)
        tablespace icust2
        initrans 3
       parallel 10
        pctfree  1;

rem
rem done
rem

        exit;
```

icust.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       icust.sql
rem  DESCRIPTION
rem       Create customer index for TPC-C database.
rem +--------------------------------------------------------------------
rem

rem
rem DROP all first
rem
        drop index icustomer;

set timing on

rem
rem ICUST1 index
rem

       create unique index icustomer on customer(c_w_id, c_d_id, c_id)
        tablespace unused
        initrans 3
       parallel 10;

rem
rem done
rem

        exit;
```

Loader Source and Scripts

gettime.c

```
/*+--------------------------------------------------------------------+
  |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
  |                     OPEN SYSTEMS PERFORMANCE GROUP                 |
  |                          All Rights Reserved                       |
  +--------------------------------------------------------------------+
  |
  | FILENAME
  |      gettime.c
  |
  | ROUTINES
  |      gettime
  |      getcpu
  | DESCRIPTION
  |      get wall clock time.
  |      get cpu time.
  | NOTES
  |      Both routines return time in seconds as a double.
  +--------------------------------------------------------------------*/

#include <stdio.h>

#ifdef ORA_NT
  #include <time.h>
  #include <sys/types.h>
  #include <sys/timeb.h>
#else
  #include <sys/time.h>
  #include <sys/resource.h>

  #include <sys/times.h>
  #include <sys/types.h>
  #include <sys/param.h>
#endif /* ORA_NT */


double gettime ()
{
#ifdef ORA_NT
   struct timeb buf;
   ftime(&buf);
   return((double) (buf.time + (1.0e-3 * (double)buf.millitm)));
```

```
#else
   struct timeval tv;
   (void) gettimeofday (&tv, (struct timezone *) 0);
   return ((double) tv.tv_sec + (1.0e-6 * (double) tv.tv_usec));
#endif /* ORA_NT */

}

double getcpu ()
{
#ifdef ORA_NT
   clock_t begin_cpu;
   begin_cpu = clock();
   return (double)begin_cpu / CLOCKS_PER_SEC;
#else
/*
   struct rusage ru;
   double usecs;

   (void) getrusage (0, &ru);
   usecs = 1.0e-6 * (double) (ru.ru_utime.tv_usec + ru.ru_stime.tv_usec);
   return ((double) (ru.ru_utime.tv_sec + ru.ru_stime.tv_sec) + usecs);
*/
   struct tms buf;
   (void)times(&buf);
   return (((double)buf.tms_utime + (double) buf.tms_stime) / HZ);
#endif /* ORA_NT */
}
```

Makefile.unix

```
#====================================================================+
# FILENAME
#      Makefile
# DESCRIPTION
#      Makefile for loader (tpccload)
#====================================================================
#
#  Programs:
#
#      tpccload: Database loader for TPC-C
#      testtime: Tesdts the time calculations for tpccload
#
#====================================================================

SRCHOME=$(ORACLE_HOME)
include prefix.mk
CC=cc -D SVRUNIX

XLDFLAGS=-u _fpgetround -u _fpsetmask -u fpgetmask -u fpsetmask

TPCBIN=../bin
INCLUDE=$(I_SYM). \
 $(I_SYM)../include \
 $(I_SYM)$(ORACLE_HOME)/rdbms/demo \
 $(I_SYM)$(ORACLE_HOME)/rdbms/public \
 $(I_SYM)$(ORACLE_HOME)/rdbms/include \
 $(I_SYM)$(ORACLE_HOME)/plsql/public \
 $(I_SYM)$(ORACLE_HOME)/network/public
ITUX=$(I_SYM)$(ROOTDIR)/include

OBJS=tpccload.o gettime.o

all: compile load
compile: $(OBJS)
load: tpccload testtime
clean:
        $(REMOVE) tpccload $(OBJS) testtime.o

testtime: gettime.o testtime.o
        $(CC) $(CFLAGS) -o $@ testtime.o gettime.o
testtime.o: testtime.c
        $(CC) $(CFLAGS) $(INCLUDE) -c testtime.c
gettime.o: gettime.c
        $(CC) $(CFLAGS) $(INCLUDE) -c gettime.c

tpccload.o: tpccload.c tpcc.h
        echo $(CC)
        $(CC) $(CFLAGS) $(INCLUDE) -c tpccload.c

tpccload: $(OBJS)
        $(LINK) -o $@ $(XLDFLAGS) $(LDFLAGS) \
        $(OBJS) \
        $(SSABED) $(DEF_OPT) $(TTLIBS)
```

prefix.mk

```
#
# $Header: prefix.mk 7030100.2 96/05/30 16:44:10 plai Generic<base> $
Copyr (c) 1995 Oracle
#

#
#====================================================================+
#        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA         |
#                    OPEN SYSTEMS PERFORMANCE GROUP                  |
#                         All Rights Reserved                        |
#====================================================================+
# FILENAME
#      prefix.mk
# DESCRIPTION
#      Makefile prefix for all Makefiles in bench directory.
#====================================================================
#


######################################################################
#  MACRO
#
DOTARGS=\
   echo "DOTARGS"; \
```

```
   echo $(DIRS); \
   TARG=`echo $@ | sed 's/\..*//'`; export TARG; \
   if [ "$(DIRS)" ]; then \
      for dir in $(DIRS); do \
         if [ -d $$dir ]; then \
            echo; echo " cd $$dir; $(MAKE) $$TARG"; \
            BASE=`basename $$dir`; \
            DATE=`date`; \
            echo " --------- Start $$BASE: $$DATE --------- "; \
            (cd $$dir; $(MAKE) $$TARG); \
            DATE=`date`; \
            echo " ---------   End $$BASE: $$DATE --------- "; \
         else \
            echo "WARNING: $$dir doesn't exist under `pwd`"; \
         fi \
      done \
   else \
      echo "WARNING: macro DIRS is NOT set!  Check dir: `pwd`"; \
   fi
#
#  DESCRIPTION:
#     Used by Makefiles for recursive targets.
#
########################################################################


########################################################################
#  MACRO
#
REMOVE=rm -f
#
#  DESCRIPTION:
#     Remove a file.  No error even if file does not exist.
#
########################################################################


########################################################################
#  MACRO
#
CHMOD=chmod
#
#  DESCRIPTION:
#     Change mode of a file or directory.
#
########################################################################


########################################################################
#  MACRO
#
MKDIR=mkdir
#
#  DESCRIPTION:
#     Make a directory.
#
########################################################################


########################################################################
#  MACRO
#
CD=cd
#
#  DESCRIPTION:
#     Change directory.
#
########################################################################


########################################################################
#  MACRO
#
SYM_LINK=ln -s
#
#  DESCRIPTION:
#     Create symbolic/soft link.
#
########################################################################


########################################################################
#  MACRO
#
CP=cp
#
#  DESCRIPTION:
#     Copy a file.
#
########################################################################


########################################################################
#  MACRO
#
#CC=ccccccc
#
#  DESCRIPTION:
#     The C compiler.
#
########################################################################


########################################################################
#  MACRO
#
LD=ld
#
#  DESCRIPTION:
#     The linker.
#
########################################################################


########################################################################
#  MACRO
#
LINK=cc
```

```
#
#  DESCRIPTION:
#     Command for linking executables.
#
########################################################################


########################################################################
#  MACRO
#
I_SYM=-I
#
#  DESCRIPTION:
#     The flag for include path entries.
#
########################################################################


########################################################################
#  MACRO
#
CFLAGS=-O
#
#  DESCRIPTION:
#     C compiler flags.
#
########################################################################


########################################################################
#  MACRO
#
TTLIBS=-lclntsh -ljava \
       -lsocket -lnsl -lgen -ldl -lmw_UXW -lelf -lm \
       -lmw_UXW -lgen -ldshm -Kthread
#
#  DESCRIPTION:
#     Libraries for linking two-task executables.
#
########################################################################


########################################################################
#  If you want to override any default macro definitions set in this
#  file or in env_rdbms.mk, e.g. CFLAGS, STLIBS, etc.,
#  do so after this message.
########################################################################


########################################################################
#  MACRO
#
LLIBPSO=`cat $(ORACLE_HOME)/rdbms/lib/psoliblist`
#  DESCRIPTION:
#     Libraries for Parallel Server Option.
#
########################################################################
########################################################################
#  MACRO
#
SDOLIBS=`if ar tv $(LIBKNLOPT) | grep $(NO_SDOPT) > /dev/null 2>&1;then \
            echo " "; \
         else \
            echo "-lmdknl -lmdhh"; \
         fi`
#  DESCRIPTION:
#     Libraries for Parallel Server Option.
#
########################################################################


########################################################################
#  MACRO
#
#STLIBS=$(ORACLE_HOME)/rdbms/lib/config.o \
#       -lclient -lserver -lcommon -lgeneric -lknlopt \
#       -lapps -lcog -lcox -lidl -lknlde  -lpkg -lpls -lsem -lsyn \
#       -lclient -lserver -lcommon -lgeneric -lknlopt \
#       -lapps -lcog -lcox -lidl -lknlde  -lpkg -lpls -lsem -lsyn \
#       -lclient -lserver -lcommon -lgeneric \
#       -lapps -lcog -lcox -lidl -lknlde  -lpkg -lpls -lsem -lsyn \
#       -lsqlnet -lclient -lserver -lcommon -lgeneric \
#       -lsqlnet -lclient -lserver -lcommon -lgeneric \
#       -lnlsrtl3 -lc3v6 -lcore3 -lnlsrtl3 -lcore3 \
#       `cat $(ORACLE_HOME)/rdbms/lib/psoliblist` \
#       $(ORACLE_HOME)/lib/epcni.o `cat
$(ORACLE_HOME)/rdbms/lib/sysliblist`
#
STLIBS=$(ORACLE_HOME)/lib/osntabst.o \
       $(ORACLE_HOME)/rdbms/lib/config.o \
       $(LLIBSERVER) $(LLIBORA) $(LLIBKNLOPT) $(LLIBSLAX) $(LLIBPLSQL) \
       $(LLIBSERVER) $(LLIBORA) $(LLIBKNLOPT) $(LLIBSLAX) $(LLIBPLSQL) \
       $(LLIBSERVER) $(LLIBORA) $(LLIBSLAX) $(LLIBPLSQL) $(LLIBSERVER) \
       $(LLIBSERVER) $(LLIBORA) $(LLIBSLAX) $(LLIBPLSQL) $(LLIBSERVER) \
       $(CLIBS)
#       $(TTLIBS) $(SDOLIBS) $(LLIBPSO) $(CLIBS) $(LIBLISTRDBMS)
#  DESCRIPTION:
#     Libraries for linking single-task executables.
#
########################################################################


########################################################################
#  MACRO
#
LDFLAGS=-L$(ORACLE_HOME)/lib -L$(ORACLE_HOME)/rdbms/lib
#
#  DESCRIPTION:
#     Flags for linking executables.
#
########################################################################
```

tpcc.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

#ifndef FALSE
# define FALSE 0
#endif

#ifndef TRUE
# define TRUE 1
#endif

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

#ifndef boolean
#define boolean int
#endif

#include <oratypes.h>
#include <oci.h>
#include <ocidfn.h>
/*
#ifdef __STDC__
#include "ociapr.h"
#else
#include "ocikpr.h"
#endif
*/

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 void TPCexit ();
extern int TPCdumpinit ();
extern void TPCdumpnew ();
extern void TPCdumppay ();
extern void TPCdumpord ();
extern void TPCdumpdel ();
extern void TPCdumpsto ();
extern void TPCdumpexit ();
extern void userlog();


/* 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

#define FULLDATE "dd-mon-yy.hh:mi:ss"
#define SHORTDATE "dd-mm-yyyy"



#define DELRT 80.0

extern int tkvcninit ();
extern int tkvcpinit ();
extern int tkvcoinit ();
extern int tkvcdinit ();
extern int tkvcsinit ();

extern int tkvcn ();
extern int tkvcp ();
extern int tkvco ();
extern int tkvcd ();
extern int tkvcs ();

extern void tkvcndone ();
extern void tkvcpdone ();
extern void tkvcodone ();
extern void tkvcddone ();
extern void tkvcsdone ();

extern int tkvcss (); /* for alter session to get memory size and trace */
extern boolean multitranx;
extern int ord_init;


extern void errrpt ();
extern int ocierror(char *fname, int lineno,OCIError *errhp, sword
status);
extern int sqlfile(char *fname, text *linebuf);
```

```
extern FILE *lfp;
extern FILE *fopen ();
extern int proc_no;
extern int doid[];

extern int execstatus;
extern int errcode;

extern OCIEnv *tpcenv;
extern OCIServer *tpcsrv;
extern OCIError *errhp;
extern OCISvcCtx *tpcsvc;
extern OCISession *tpcusr;
extern OCIStmt *curntest;
/* The bind and define handles for each transaction are
   included in their respective header files. */



/* for stock-level transaction */

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

/* for delivery transaction */

extern int del_o_id[10];
extern int carrier_id;
extern int retries;

/* for order-status transaction */

extern int bylastname;
extern char c_last[17];
extern char c_first[17];
extern char c_middle[3];
extern double c_balance;
extern int o_id;
extern text o_entry_d[20];
extern int o_carrier_id;
extern int o_ol_cnt;
extern int ol_supply_w_id[15];
extern int ol_i_id[15];
extern int ol_quantity[15];
extern int ol_amount[15];
ub4    ol_del_len[15];
extern text ol_delivery_d[15][11];

/* for payment transaction */

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

/* for new order transaction */

extern int nol_i_id[15];
extern int nol_supply_w_id[15];
extern int nol_quantity[15];
extern int nol_quanti10[15];
extern int nol_quanti91[15];
extern int nol_ytdqty[15];
extern int nol_amount[15];
extern int o_all_local;
extern int w_tax;
extern int d_tax;
extern float total_amount;
extern char i_name[15][25];
extern int i_name_strlen[15];
extern ub2 i_name_strlen_len[15];
extern ub2 i_name_strlen_rcode[15];
extern ub4 i_name_strlen_csize;
extern int s_quantity[15];
extern char brand_gen[15];
extern ub2 brand_gen_len[15];
extern ub2 brand_gen_rcode[15];
extern ub4 brand_gen_csize;
extern int i_price[15];
extern char brand_generic[15][1];
extern int status;
extern int tracelevel;

/* Miscellaneous */
extern OCIDate cr_date;
extern OCIDate c_since;
extern OCIDate o_entry_d_base;
extern OCIDate ol_d_base[15];

#ifndef DISCARD
# define DISCARD (void)
#endif
```

```
#ifndef sword
# define sword int
#endif

#define VER7            2

#define NA              -1      /* ANSI SQL NULL */
#define NLT             1       /* length for string null terminator */
#define DEADLOCK        60      /* ORA-00060: deadlock */
#define NO_DATA_FOUND   1403    /* ORA-01403: no data found */
#define NOT_SERIALIZABLE  8177  /* ORA-08177: transaction not serializable
*/
#define SNAPSHOT_TOO_OLD  1555  /* ORA-01555: snapshot too old */

#ifndef NULLP
# define NULLP (void *)NULL
#endif /* NULLP */

#define ADR(object) ((ub1 *)&(object))
#define SIZ(object) ((sword)sizeof(object))

typedef char date[24+NLT];
typedef char varchar2;

#define min(x,y)  (((x) < (y)) ? (x) : (y))

#define OCIERROR(errp,function)\
        ocierror(__FILE__,__LINE__,(errp),(function));

#define OCIBND(stmp, bndp, errp, sqlvar, progv, progvl, ftype)\
       ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
       ocierror(__FILE__,__LINE__, (errp), \
             OCIBindByName((stmp), &(bndp), (errp), \
                    (text *)(sqlvar), strlen((sqlvar)),\
                    (progv), (progvl), (ftype),0,0,0,0,0,OCI_DEFAULT));

#define
OCIBNDRA(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode) \
       ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
       ocierror(__FILE__,__LINE__,(errp), \
            OCIBindByName((stmp),&(bndp),(errp),(text
*)(sqlvar),strlen((sqlvar)),\

(progv),(progvl),(ftype),(indp),(alen),(arcode),0,0,OCI_DEFAULT));

#define
OCIBNDRAD(stmp,bndp,errp,sqlvar,progvl,ftype,indp,ctxp,cbf_nodata,cbf_data
) \
       ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
       ocierror(__FILE__,__LINE__,(errp), \
            OCIBindByName((stmp),&(bndp),(errp),(text *)(sqlvar), \
                          strlen((sqlvar)),0,(progvl),(ftype), \
                          indp,0,0,0,0,OCI_DATA_AT_EXEC)); \
       ocierror(__FILE__,__LINE__,(errp), \

OCIBindDynamic((bndp),(errp),(ctxp),(cbf_nodata),(ctxp),(cbf_data)));



#define OCIBNDR(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode)
\
       ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
       ocierror(__FILE__,__LINE__,(errp), \
          OCIBindByName((stmp),&(bndp),(errp),(text
*)(sqlvar),strlen((sqlvar)),\

(progv),(progvl),(ftype),(indp),(alen),(arcode),0,0,OCI_DEFAULT));

#define
OCIBNDRAA(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode,ms,cu)
\
       ocierror(__FILE__,__LINE__, (errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0));\
       ocierror(__FILE__,__LINE__,(errp),\
         OCIBindByName((stmp),&(bndp),(errp),(text
*)(sqlvar),strlen((sqlvar)),\

(progv),(progvl),(ftype),(indp),(alen),(arcode),(ms),(cu),OCI_DEFAULT));

#define OCIDEFINE(stmp,dfnp,errp,pos,progv,progvl,ftype)\

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),(progvl),(ftype),\
                     0,0,0,OCI_DEFAULT);


#define OCIDEF(stmp,dfnp,errp,pos,progv,progvl,ftype) \
       OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTYPE_DEFINE,0,\
                                        (dvoid**)0);\
       OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),(progvl),\
                           (ftype),NULL,NULL,NULL,OCI_DEFAULT); \


#define OCIDFNRA(stmp,dfnp,errp,pos,progv,progvl,ftype,indp,alen,arcode) \
       OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTYPE_DEFINE,0,\
                                        (dvoid**)0);\
       OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),\
                                        (progvl),(ftype),(indp),(alen),\
                                        (arcode),OCI_DEFAULT);

#define
OCIDFNDYN(stmp,dfnp,errp,pos,progv,progvl,ftype,indp,ctxp,cbf_data) \
       ocierror(__FILE__,__LINE__,(errp), \
       OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTYPE_DEFINE,0,\
                                        (dvoid**)0));\
       ocierror(__FILE__,__LINE__,(errp), \
       OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),
(progvl),(ftype),\
                                          (indp),NULL,NULL,
OCI_DYNAMIC_FETCH));\
```

```
        ocierror(__FILE__,__LINE__,(errp), \
        OCIDefineDynamic((dfnp),(errp),(ctxp),(cbf_data)));



/* 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;
};

#endif
```

tpccload.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: tpccload.c 7030100.1 96/05/13 16:20:36 plai Generic<base> $
Copyr (c) 1993 Oracle";
#endif /* RCSID */

/*+=================================================================+
  |        Copyright (c) 1994  Oracle Corp, Redwood Shores, CA      |
  |                    OPEN SYSTEMS PERFORMANCE GROUP               |
  |                         All Rights Reserved                     |
  +=================================================================+
  | FILENAME
  |    tpccload.c
  | DESCRIPTION
  |    Load or generate TPC-C database tables.
  |    Usage: tpccload -M <# of warehouses> [options]
  |               options: -A  load all tables
  |                        -w  load warehouse table
  |                        -d  load district table
  |                        -c  load customer table
  |                        -i  load item table
  |                        -s  load stock table (cluster around s_w_id)
  |                        -S  load stock table (cluster around s_i_id)
  |                        -h  load history table
  |                        -n  load new-order table
  |                        -o <oline file> load order and order-line table
  |                        -b <ware#>  beginning warehouse number
  |                        -e <ware#>  ending warehouse number
  |                        -j <item#>  beginning item number (with -S)
  |                        -k <item#>  ending item number (with -S)
  |                        -g  generate rows to standard output
  +=================================================================*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys/types.h>
#include "tpcc.h"

#ifdef ORA_NT
#undef boolean
#include <process.h>
#define lrand48() ((long)rand() <<15 | rand())
#ifdef __STDC__
#  define PROTO(args)    args
#else
#  define PROTO(args)    ()
#endif
#endif /*ORA_NT*/

extern double gettime();
extern double getcpu();

#define DISTARR 10            /* district insert array size */
#define CUSTARR 100           /* customer insert array size */
#define STOCARR 100           /* stock insert array size    */
#define ITEMARR 100           /* item insert array size     */
#define HISTARR 100             /* history insert array size    */
#define ORDEARR 100             /* order insert array size      */
```

```
#define NEWOARR 100             /* new order insert array size  */

#define DISTFAC 10            /* max. disctrict id   */
#define CUSTFAC 3000          /* max. customer id    */
#define STOCFAC 100000        /* max. stock id       */
#define ITEMFAC 100000        /* max. item id                */
#define HISTFAC 30000           /* history / warehouse  */
#define ORDEFAC 3000            /* order / district     */
#define NEWOFAC 900             /* new order / district */

#define C       0               /* constant in non-uniform dist. eqt. */
#define CNUM1   1               /* first constant in non-uniform dist.
eqt. */
#define CNUM2   2               /* second constant in non-uniform dist.
eqt. */
#define CNUM3   3               /* third constant in non-uniform dist.
eqt. */

#define SEED    2               /* seed for random functions */

#define SQLTXTW "INSERT INTO warehouse (w_id, w_ytd, w_tax, w_name,
w_street_1, w_street_2, w_city, w_state, w_zip) VALUES (:w_id, 30000000,
:w_tax, :w_name, :w_street_1, \
   :w_street_2, :w_city, :w_state, :w_zip)"

#define SQLTXTD "INSERT INTO district (d_id, d_w_id, d_ytd, d_tax,
d_next_o_id, d_name, d_street_1, d_street_2, d_city, d_state, d_zip)
VALUES (:d_id, :d_w_id,3000000, :d_tax, \
     3001, :d_name, :d_street_1, :d_street_2, :d_city, :d_state, :d_zip)"

#define SQLTXTC "INSERT INTO customer (C_ID, C_D_ID, C_W_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_YTD_PAYMENT,
C_PAYMENT_CNT, C_DELIVERY_CNT, C_DATA) VALUES (:c_id, :c_d_id, :c_w_id, \
   :c_first, 'OE', :c_last, :c_street_1, :c_street_2, :c_city, :c_state, \
   :c_zip, :c_phone, SYSDATE, :c_credit, 5000000, :c_discount, -1000,
1000, 1, \
   0, :c_data)"

#define SQLTXTH "INSERT INTO history (h_c_id, h_c_d_id, h_c_w_id, h_d_id,
h_w_id, h_date, h_amount, h_data) VALUES (:h_c_id, :h_c_d_id, :h_c_w_id, \
   :h_d_id, :h_w_id, SYSDATE, 1000, :h_data)"

#define SQLTXTS "INSERT INTO stock (s_i_id, s_w_id, s_quantity,s_dist_01,
s_dist_02, s_dist_03, s_dist_04, s_dist_05 , s_dist_06, s_dist_07,
s_dist_08, s_dist_09, s_dist_10, s_ytd, s_order_cnt, s_remote_cnt, s_data)
\
VALUES (:s_i_id, :s_w_id, :s_quantity, \
   :s_dist_01, :s_dist_02, :s_dist_03, :s_dist_04, :s_dist_05, :s_dist_06,
\
   :s_dist_07, :s_dist_08, :s_dist_09, :s_dist_10, 0, 0, 0, :s_data)" \

#define SQLTXTI "INSERT INTO item (I_ID,I_IM_ID,I_NAME,I_PRICE,I_DATA)
VALUES (:i_id, :i_im_id, :i_name, :i_price, \
   :i_data)"

#define SQLTXTO1 "INSERT INTO orders (O_ID,
O_D_ID,O_W_ID,O_C_ID,O_ENTRY_D,O_CARRIER_ID,O_OL_CNT,O_ALL_LOCAL) \
   VALUES (:o_id, :o_d_id, :o_w_id, :o_c_id, \
   SYSDATE, :o_carrier_id, :o_ol_cnt, 1)"

#define SQLTXTO2 "INSERT INTO orders (O_ID,
O_D_ID,O_W_ID,O_C_ID,O_ENTRY_D,O_CARRIER_ID,O_OL_CNT,O_ALL_LOCAL) \
   VALUES (:o_id, :o_d_id, :o_w_id, :o_c_id, \
   SYSDATE, 11, :o_ol_cnt, 1)"

#define SQLTXTOL1 "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) \
   VALUES (:ol_o_id, :ol_d_id, \
   :ol_w_id, :ol_number, SYSDATE, :ol_i_id, :ol_supply_w_id, 5, 0, \
   :ol_dist_info)"

#define SQLTXTOL2 "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) \
   VALUES (:ol_o_id, :ol_d_id, \
   :ol_w_id, :ol_number, to_date('01-Jan-1811'), :ol_i_id,
:ol_supply_w_id, 5, :ol_amount, \
   :ol_dist_info)"

#define SQLTXTNO "INSERT INTO new_order (no_o_id, no_d_id, no_w_id) VALUES
(:no_o_id, :no_d_id, :no_w_id)"

ldadef tpclda;
csrdef curw, curd, curc, curh, curs, curi, curo1, curo2, curol1, curol2,
curno;
unsigned long tpchda[256];

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

char num9[10];
char num16[17];
char str2[3];
char str24[15][25];
int randperm3000[3000];

void initperm();
void randstr();
void randdatastr();
void randnum();
void randlastname (char*, int);
int NURand();
void sysdate();


void myusage()
```

```
{

   fprintf (stderr, "\n");
   fprintf (stderr, "Usage:\ttpccload -M <multiplier> [options]\n");
   fprintf (stderr, "options:\n");
   fprintf (stderr, "\t-A :\tload all tables\n");
   fprintf (stderr, "\t-w :\tload warehouse table\n");
   fprintf (stderr, "\t-d :\tload district table\n");
   fprintf (stderr, "\t-c :\tload customer table\n");
   fprintf (stderr, "\t-i :\tload item table\n");
   fprintf (stderr, "\t-s :\tload stock table (cluster around s_w_id)\n");
   fprintf (stderr, "\t-S :\tload stock table (cluster around s_i_id)\n");
   fprintf (stderr, "\t-h :\tload history table\n");
   fprintf (stderr, "\t-n :\tload new-order table\n");
   fprintf (stderr, "\t-o <oline file> :\tload order and order-line
table\n");
   fprintf (stderr, "\t-b <ware#> :\tbeginning warehouse number\n");
   fprintf (stderr, "\t-e <ware#> :\tending warehouse number\n");
   fprintf (stderr, "\t-j <item#> :\tbeginning item number (with -S)\n");
   fprintf (stderr, "\t-k <item#> :\tending item number (with -S)\n");
   fprintf (stderr, "\t-g :\tgenerate rows to standard output\n");
   fprintf (stderr, "\n");
   exit(1);

}



void errrpt (lda, cur)

csrdef *lda;
csrdef *cur;

{

   text msg[2048];

   if (cur->rc) {
      oerhms (lda, cur->rc, msg, 2048);
      fprintf (stderr, "TPC-C load error: %s\n", msg);
   }

}



void quit ()

{

   if (oclose (&curw))
      errrpt (&tpclda, &curw);

   if (oclose (&curd))
      errrpt (&tpclda, &curd);

   if (oclose (&curc))
      errrpt (&tpclda, &curc);

   if (oclose (&curh))
      errrpt (&tpclda, &curh);

   if (oclose (&curs))
      errrpt (&tpclda, &curs);

   if (oclose (&curi))
      errrpt (&tpclda, &curi);

   if (oclose (&curo1))
      errrpt (&tpclda, &curo1);

   if (oclose (&curo2))
      errrpt (&tpclda, &curo2);

   if (oclose (&curol1))
      errrpt (&tpclda, &curol1);

   if (oclose (&curol2))
      errrpt (&tpclda, &curol2);

   if (oclose (&curno))
      errrpt (&tpclda, &curno);

   if (ologof (&tpclda))
      fprintf (stderr, "TPC-C load error: Error in logging off\n");

}



void main (argc, argv)

int argc;
char *argv[];

{

   char *uid="tpcc/tpcc";
   text sqlbuf[1024];
   int scale=0;
   int i, j;
   int loop;
   int loopcount;
   int cid;
   int dwid;
   int cdid;
   int cwid;
   int sid;
   int swid;
   int olcnt;
   int nrows;
   int row;

   int w_id;
   char w_name[11];
   char w_street_1[21];
   char w_street_2[21];
```

```
   char w_city[21];
   char w_state[2];
   char w_zip[9];
#ifdef SVRUNIX
   float w_tax;
#else
   int w_tax;
#endif
   int d_id[10];
   int d_w_id[10];
   char d_name[10][11];
   char d_street_1[10][21];
   char d_street_2[10][21];
   char d_city[10][21];
   char d_state[10][2];
   char d_zip[10][9];
#ifdef SVRUNIX
   float d_tax[10];
#else
   int d_tax[10];
#endif

   int c_id[100];
   int c_d_id[100];
   int c_w_id[100];
   char c_first[100][17];
   char c_last[100][17];
   char c_street_1[100][21];
   char c_street_2[100][21];
   char c_city[100][21];
   char c_state[100][2];
   char c_zip[100][9];
   char c_phone[100][16];
   char c_credit[100][2];
   float c_discount[100];
   char c_data[100][501];

   int i_id[100];
   int i_im_id[100];
   int i_price[100];
   char i_name[100][25];
   char i_data[100][51];

   int s_i_id[100];
   int s_w_id[100];
   int s_quantity[100];
   char s_dist_01[100][24];
   char s_dist_02[100][24];
   char s_dist_03[100][24];
   char s_dist_04[100][24];
   char s_dist_05[100][24];
   char s_dist_06[100][24];
   char s_dist_07[100][24];
   char s_dist_08[100][24];
   char s_dist_09[100][24];
   char s_dist_10[100][24];
   char s_data[100][51];

   int h_w_id[100];
   int h_d_id[100];
   int h_c_id[100];
   char h_data[100][25];

   int o_id[100];
   int o_d_id[100];
   int o_w_id[100];
   int o_c_id[100];
   int o_carrier_id[100];
   int o_ol_cnt[100];

   int ol_o_id[15];
   int ol_d_id[15];
   int ol_w_id[15];
   int ol_number[15];
   int ol_i_id[15];
   int ol_supply_w_id[15];
   int ol_amount[15];
   char ol_dist_info[15][24];

   int no_o_id[100];
   int no_d_id[100];
   int no_w_id[100];

   char sdate[30];

       double begin_time, end_time;
   double begin_cpu, end_cpu;
   double gettime(), getcpu();

#ifdef ORA_NT
  char *arg_ptr,  **end_args;
#else
   extern int getopt();
   extern char *optarg;
   extern int optind, opterr;

   int opt;
#endif

   char *argstr="M:AwdcisShno:b:e:j:k:g";
   int do_A=0;
   int do_w=0;
   int do_d=0;
   int do_i=0;
   int do_c=0;
   int do_s=0;
   int do_S=0;
   int do_h=0;
   int do_o=0;
   int do_n=0;
   int gen=0;
   int bware=1;
   int eware=0;
   int bitem=1;
   int eitem=0;

   FILE *olfp=NULL;
   char olfname[100];
```

```
#ifdef ORA_NT
   char fname[100];
   FILE *logfile;
#endif /* ORA_NT */

/*---------------------------------------------------------------+
 | Parse command line -- look for scale factor.                  |
 +---------------------------------------------------------------*/

   if (argc == 1) {
      myusage ();
   }
#ifdef ORA_NT
   end_args = argv + argc;

   for (++argv; argv < end_args; )
   {
      arg_ptr = *argv++;

          if (*arg_ptr != '-')
          {
             myusage ();
          } else
          {
        switch (arg_ptr[1]) {
         case '?': myusage ();
                   break;
         case 'M': scale = atoi (*argv++);
                   break;
         case 'A': do_A = 1;
                   break;
         case 'w': do_w = 1;
                   break;
         case 'd': do_d = 1;
                   break;
         case 'c': do_c = 1;
                   break;
         case 'i': do_i = 1;
                   break;
         case 's': do_s = 1;
                   break;
         case 'S': do_S = 1;
                   break;
         case 'h': do_h = 1;
                   break;
         case 'n': do_n = 1;
                   break;
         case 'o': do_o = 1;
                   strcpy (olfname, *argv++);
                   break;
         case 'b': bware = atoi (*argv++);
                   break;
         case 'e': eware = atoi (*argv++);
                   break;
         case 'j': bitem = atoi (*argv++);
                   break;
         case 'k': eitem = atoi (*argv++);
                   break;
         case 'g': gen = 1;
                   break;
         case 'l': logfile=fopen(*argv++,"w");
                   break;
         default: fprintf (stderr, "THIS SHOULD NEVER HAPPEN!!!\n");
                  fprintf (stderr, "(reached default case in getopt
())\n");
                  myusage ();
        }
          }
   }


#else

   while ((opt = getopt (argc, argv, argstr)) != -1) {
      switch (opt) {
         case '?': myusage ();
                   break;
         case 'M': scale = atoi (optarg);
                   break;
         case 'A': do_A = 1;
                   break;
         case 'w': do_w = 1;
                   break;
         case 'd': do_d = 1;
                   break;
         case 'c': do_c = 1;
                   break;
         case 'i': do_i = 1;
                   break;
         case 's': do_s = 1;
                   break;
         case 'S': do_S = 1;
                   break;
         case 'h': do_h = 1;
                   break;
         case 'n': do_n = 1;
                   break;
         case 'o': do_o = 1;
                   strcpy (olfname, optarg);
                   break;
         case 'b': bware = atoi (optarg);
                   break;
         case 'e': eware = atoi (optarg);
                   break;
         case 'j': bitem = atoi (optarg);
                   break;
         case 'k': eitem = atoi (optarg);
                   break;
         case 'g': gen = 1;
                   break;
         default: fprintf (stderr, "THIS SHOULD NEVER HAPPEN!!!\n");
                  fprintf (stderr, "(reached default case in getopt
())\n");
                  myusage ();
      }
   }
# endif /* ORA_NT */
```

```
/*-------------------------------------------------------------*|
|       Rudimentary error checking                             |
|*-------------------------------------------------------------*/

   if (scale  < 1) {
      fprintf (stderr, "Invalid scale factor: '%d'\n", scale);
      myusage ();
   }

   if (!(do_A || do_w || do_d || do_c || do_i || do_s || do_S || do_h ||
do_o ||
         do_n)) {
      fprintf (stderr, "What should I load???\n");
      myusage ();
   }

   if (gen && (do_A || (do_w + do_d + do_c + do_i + do_s + do_S + do_h +
do_o +
                        do_n > 1))) {
      fprintf (stderr, "Can only generate table one at a time\n");
      myusage ();
   }

   if (do_S && (do_A || do_s)) {
      fprintf (stderr, "Cluster stock table around s_w_id or s_i_id?\n");
      myusage ();
   }

   if (eware <= 0)
      eware = scale;
   if (eitem <= 0)
      eitem = STOCFAC;

   if (do_S) {
      if ((bitem  < 1) || (bitem > STOCFAC)) {
         fprintf (stderr, "Invalid beginning item number: '%d'\n", bitem);
         myusage ();
      }

      if ((eitem  < bitem) || (eitem > STOCFAC)) {
         fprintf (stderr, "Invalid ending item number: '%d'\n", eitem);
         myusage ();
      }
   }

   if ((bware  < 1) || (bware > scale)) {
      fprintf (stderr, "Invalid beginning warehouse number: '%d'\n",
bware);
      myusage ();
   }

   if ((eware  < bware) || (eware > scale)) {
      fprintf (stderr, "Invalid ending warehouse number: '%d'\n", eware);
      myusage ();
   }

   if (gen && do_o) {
      if ((olfp = fopen (olfname, "w")) == NULL) {
         fprintf (stderr, "Can't open '%s' for writing order lines\n",
olfname);
         myusage ();
      }
   }

/*-------------------------------------------------------------+
 | Prepare to insert into database.                            |
 +-------------------------------------------------------------*/

   sysdate (sdate);
   if (!gen) {

      /* log on to Oracle */

      if (orlon (&tpclda, (ub1 *) tpchda, (text *) uid, -1, (text *) 0, -
1, 0)) {
         fprintf (stderr, "TPC-C load error: Error in logging on\n");
         errrpt (&tpclda, &tpclda);
         exit (1);
      }

      fprintf (stderr, "\nConnected to Oracle userid '%s'.\n", uid);

      /* turn off auto-commit */

      if (ocof (&tpclda)) {
         errrpt (&tpclda, &tpclda);
         ologof (&tpclda);
         exit (1);
      }

      /* open cursors */

      if (oopen (&curw, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curw);
         ologof (&tpclda);
         exit (1);
      }

      if (oopen (&curd, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curd);
         oclose (&curw);
         ologof (&tpclda);
         exit (1);
      }

      if (oopen (&curc, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curc);
         oclose (&curw);
         oclose (&curd);
         ologof (&tpclda);
         exit (1);
      }

      if (oopen (&curh, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curh);
         oclose (&curw);
         oclose (&curd);
```

```
      oclose (&curc);
      ologof (&tpclda);
      exit (1);
   }

   if (oopen (&curs, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
      errrpt (&tpclda, &curs);
      oclose (&curw);
      oclose (&curd);
      oclose (&curc);
      oclose (&curh);
      ologof (&tpclda);
      exit (1);
   }

   if (oopen (&curi, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
      errrpt (&tpclda, &curi);
      oclose (&curw);
      oclose (&curd);
      oclose (&curc);
      oclose (&curh);
      oclose (&curs);
      ologof (&tpclda);
      exit (1);
   }

   if (oopen (&curo1, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
      errrpt (&tpclda, &curo1);
      oclose (&curw);
      oclose (&curd);
      oclose (&curc);
      oclose (&curh);
      oclose (&curs);
      oclose (&curi);
      ologof (&tpclda);
      exit (1);
   }

   if (oopen (&curo2, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
      errrpt (&tpclda, &curo2);
      oclose (&curw);
      oclose (&curd);
      oclose (&curc);
      oclose (&curh);
      oclose (&curs);
      oclose (&curi);
      oclose (&curo1);
      ologof (&tpclda);
      exit (1);
   }

   if (oopen (&curol1, &tpclda, (text *) 0, -1, -1, (text *) uid, -1))
{
      errrpt (&tpclda, &curol1);
      oclose (&curw);
      oclose (&curd);
      oclose (&curc);
      oclose (&curh);
      oclose (&curs);
      oclose (&curi);
      oclose (&curo1);
      oclose (&curo2);
      ologof (&tpclda);
      exit (1);
   }

   if (oopen (&curol2, &tpclda, (text *) 0, -1, -1, (text *) uid, -1))
{
      errrpt (&tpclda, &curol2);
      oclose (&curw);
      oclose (&curd);
      oclose (&curc);
      oclose (&curh);
      oclose (&curs);
      oclose (&curi);
      oclose (&curo1);
      oclose (&curo2);
      oclose (&curol1);
      ologof (&tpclda);
      exit (1);
   }

   if (oopen (&curno, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
      errrpt (&tpclda, &curno);
      oclose (&curw);
      oclose (&curd);
      oclose (&curc);
      oclose (&curh);
      oclose (&curs);
      oclose (&curi);
      oclose (&curo1);
      oclose (&curo2);
      oclose (&curol1);
      oclose (&curol2);
      ologof (&tpclda);
      exit (1);
   }

   /* parse statements */

   sprintf ((char *) sqlbuf, SQLTXTW);
   if (oparse (&curw, sqlbuf, -1, 0, 1)) {
      errrpt (&tpclda, &curw);
      quit ();
      exit (1);
   }

   sprintf ((char *) sqlbuf, SQLTXTD);
   if (oparse (&curd, sqlbuf, -1, 0, 1)) {
      errrpt (&tpclda, &curd);
      quit ();
      exit (1);
   }

   sprintf ((char *) sqlbuf, SQLTXTC);
   if (oparse (&curc, sqlbuf, -1, 0, 1)) {
      errrpt (&tpclda, &curc);
      quit ();
```

```
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTH);
if (oparse (&curh, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curh);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTS);
if (oparse (&curs, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTI);
if (oparse (&curi, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curi);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTO1);
if (oparse (&curo1, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curo1);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTO2);
if (oparse (&curo2, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curo2);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTOL1);
if (oparse (&curol1, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curol1);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTOL2);
if (oparse (&curol2, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curol2);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTNO);
if (oparse (&curno, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curno);
   quit ();
   exit (1);
}
```

```
      /* bind variables */

      /* warehouse */

      if (obndrv (&curw, (text *) ":w_id", -1, (ub1 *) &w_id, sizeof
(w_id),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curw);
         quit ();
         exit (1);
      }

      if (obndrv (&curw, (text *) ":w_name", -1, (ub1 *) w_name, 11,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curw);
         quit ();
         exit (1);
      }

      if (obndrv (&curw, (text *) ":w_street_1", -1, (ub1 *) w_street_1,
21,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curw);
         quit ();
         exit (1);
      }

      if (obndrv (&curw, (text *) ":w_street_2", -1, (ub1 *) w_street_2,
21,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curw);
         quit ();
         exit (1);
      }

      if (obndrv (&curw, (text *) ":w_city", -1, (ub1 *) w_city, 21,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curw);
         quit ();
         exit (1);
      }

      if (obndrv (&curw, (text *) ":w_state", -1, (ub1 *) w_state, 2,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curw);
         quit ();
         exit (1);
      }

      if (obndrv (&curw, (text *) ":w_zip", -1, (ub1 *) w_zip, 9,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curw);
         quit ();
         exit (1);
      }

#ifdef SVRUNIX
      if (obndrv (&curw, (text *) ":w_tax", -1, (ub1 *) &w_tax,
sizeof(float),
```

```
                  SQLT_FLT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
#else
      if (obndrv (&curw, (text *) ":w_tax", -1, (ub1 *) &w_tax,
sizeof(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
#endif
         errrpt (&tpclda, &curw);
         quit ();
         exit (1);
      }

      /* district */

      if (obndrv (&curd, (text *) ":d_id", -1, (ub1 *) d_id, sizeof (int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curd);
         quit ();
         exit (1);
      }

      if (obndrv (&curd, (text *) ":d_w_id", -1, (ub1 *) d_w_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curd);
         quit ();
         exit (1);
      }

      if (obndrv (&curd, (text *) ":d_name", -1, (ub1 *) d_name, 11,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curd);
         quit ();
         exit (1);
      }

      if (obndrv (&curd, (text *) ":d_street_1", -1, (ub1 *) d_street_1,
21,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curd);
         quit ();
         exit (1);
      }

      if (obndrv (&curd, (text *) ":d_street_2", -1, (ub1 *) d_street_2,
21,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curd);
         quit ();
         exit (1);
      }

      if (obndrv (&curd, (text *) ":d_city", -1, (ub1 *) d_city, 21,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curd);
         quit ();
         exit (1);
      }

      if (obndrv (&curd, (text *) ":d_state", -1, (ub1 *) d_state, 2,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curd);
         quit ();
         exit (1);
      }

      if (obndrv (&curd, (text *) ":d_zip", -1, (ub1 *) d_zip, 9,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curd);
         quit ();
         exit (1);
      }

#ifdef SVRUNIX
      if (obndrv (&curd, (text *) ":d_tax", -1, (ub1 *) d_tax,
sizeof(float),
                  SQLT_FLT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
#else
      if (obndrv (&curd, (text *) ":d_tax", -1, (ub1 *) d_tax,
sizeof(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
#endif
         errrpt (&tpclda, &curd);
         quit ();
         exit (1);
      }

      /* customer */

      if (obndrv (&curc, (text *) ":c_id", -1, (ub1 *) c_id, sizeof (int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }

      if (obndrv (&curc, (text *) ":c_d_id", -1, (ub1 *) c_d_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }

      if (obndrv (&curc, (text *) ":c_w_id", -1, (ub1 *) c_w_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }

      if (obndrv (&curc, (text *) ":c_first", -1, (ub1 *) c_first, 17,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }
```

```
      if (obndrv (&curc, (text *) ":c_last", -1, (ub1 *) c_last, 17,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }

      if (obndrv (&curc, (text *) ":c_street_1", -1, (ub1 *) c_street_1,
21,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }

      if (obndrv (&curc, (text *) ":c_street_2", -1, (ub1 *) c_street_2,
21,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }

      if (obndrv (&curc, (text *) ":c_city", -1, (ub1 *) c_city, 21,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }

      if (obndrv (&curc, (text *) ":c_state", -1, (ub1 *) c_state, 2,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }

      if (obndrv (&curc, (text *) ":c_zip", -1, (ub1 *) c_zip, 9,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }

      if (obndrv (&curc, (text *) ":c_phone", -1, (ub1 *) c_phone, 16,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }

      if (obndrv (&curc, (text *) ":c_credit", -1, (ub1 *) c_credit, 2,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }

      if (obndrv (&curc, (text *) ":c_discount", -1, (ub1 *) c_discount,
                  sizeof (int), SQLT_FLT, -1, (sb2 *) 0, (text *) 0, -1,
                  -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }

      if (obndrv (&curc, (text *) ":c_data", -1, (ub1 *) c_data, 501,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curc);
         quit ();
         exit (1);
      }

      /* item */

      if (obndrv (&curi, (text *) ":i_id", -1, (ub1 *) i_id, sizeof (int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curi);
         quit ();
         exit (1);
      }

      if (obndrv (&curi, (text *) ":i_im_id", -1, (ub1 *) i_im_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curi);
         quit ();
         exit (1);
      }

      if (obndrv (&curi, (text *) ":i_name", -1, (ub1 *) i_name, 25,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curi);
         quit ();
         exit (1);
      }

      if (obndrv (&curi, (text *) ":i_price", -1, (ub1 *) i_price,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1,
                  -1)) {
         errrpt (&tpclda, &curi);
         quit ();
         exit (1);
      }

      if (obndrv (&curi, (text *) ":i_data", -1, (ub1 *) i_data, 51,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curi);
         quit ();
         exit (1);
      }

      /* stock */

      if (obndrv (&curs, (text *) ":s_i_id", -1, (ub1 *) s_i_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curs);
```

```
         quit ();
         exit (1);
      }

      if (obndrv (&curs, (text *) ":s_w_id", -1, (ub1 *) s_w_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curs);
         quit ();
         exit (1);
      }

      if (obndrv (&curs, (text *) ":s_quantity", -1, (ub1 *) s_quantity,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -
1)) {
         errrpt (&tpclda, &curs);
         quit ();
         exit (1);
      }

      if (obndrv (&curs, (text *) ":s_dist_01", -1, (ub1 *) s_dist_01, 24,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curs);
         quit ();
         exit (1);
      }

      if (obndrv (&curs, (text *) ":s_dist_02", -1, (ub1 *) s_dist_02, 24,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curs);
         quit ();
         exit (1);
      }

      if (obndrv (&curs, (text *) ":s_dist_03", -1, (ub1 *) s_dist_03, 24,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curs);
         quit ();
         exit (1);
      }

      if (obndrv (&curs, (text *) ":s_dist_04", -1, (ub1 *) s_dist_04, 24,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curs);
         quit ();
         exit (1);
      }

      if (obndrv (&curs, (text *) ":s_dist_05", -1, (ub1 *) s_dist_05, 24,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curs);
         quit ();
         exit (1);
      }

      if (obndrv (&curs, (text *) ":s_dist_06", -1, (ub1 *) s_dist_06, 24,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curs);
         quit ();
         exit (1);
      }

      if (obndrv (&curs, (text *) ":s_dist_07", -1, (ub1 *) s_dist_07, 24,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curs);
         quit ();
         exit (1);
      }

      if (obndrv (&curs, (text *) ":s_dist_08", -1, (ub1 *) s_dist_08, 24,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curs);
         quit ();
         exit (1);
      }

      if (obndrv (&curs, (text *) ":s_dist_09", -1, (ub1 *) s_dist_09, 24,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curs);
         quit ();
         exit (1);
      }

      if (obndrv (&curs, (text *) ":s_dist_10", -1, (ub1 *) s_dist_10, 24,
                  SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curs);
         quit ();
         exit (1);
      }

      if (obndrv (&curs, (text *) ":s_data", -1, (ub1 *) s_data, 51,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curs);
         quit ();
         exit (1);
      }

      /* history */

      if (obndrv (&curh, (text *) ":h_c_id", -1, (ub1 *) h_c_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curh);
         quit ();
         exit (1);
      }

      if (obndrv (&curh, (text *) ":h_c_d_id", -1, (ub1 *) h_d_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curh);
         quit ();
         exit (1);
      }

      if (obndrv (&curh, (text *) ":h_c_w_id", -1, (ub1 *) h_w_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
```

```
         errrpt (&tpclda, &curh);
         quit ();
         exit (1);
      }

      if (obndrv (&curh, (text *) ":h_d_id", -1, (ub1 *) h_d_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curh);
         quit ();
         exit (1);
      }

      if (obndrv (&curh, (text *) ":h_w_id", -1, (ub1 *) h_w_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curh);
         quit ();
         exit (1);
      }

      if (obndrv (&curh, (text *) ":h_data", -1, (ub1 *) h_data, 25,
                  SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curh);
         quit ();
         exit (1);
      }

      /* order_line (delivered) */

      if (obndrv (&curol1, (text *) ":ol_o_id", -1, (ub1 *) ol_o_id,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -
1)) {
         errrpt (&tpclda, &curol1);
         quit ();
         exit (1);
      }

      if (obndrv (&curol1, (text *) ":ol_d_id", -1, (ub1 *) ol_d_id,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -
1)) {
         errrpt (&tpclda, &curol1);
         quit ();
         exit (1);
      }

      if (obndrv (&curol1, (text *) ":ol_w_id", -1, (ub1 *) ol_w_id,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -
1)) {
         errrpt (&tpclda, &curol1);
         quit ();
         exit (1);
      }

      if (obndrv (&curol1, (text *) ":ol_number", -1, (ub1 *) ol_number,
          sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curol1);
         quit ();
         exit (1);
      }

      if (obndrv (&curol1, (text *) ":ol_i_id", -1, (ub1 *) ol_i_id,
          sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curol1);
         quit ();
         exit (1);
      }

      if (obndrv (&curol1, (text *) ":ol_supply_w_id", -1,
                  (ub1 *) ol_supply_w_id, sizeof (int), SQLT_INT, -1,
                  (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curol1);
         quit ();
         exit (1);
      }

      if (obndrv (&curol1, (text *) ":ol_dist_info", -1, (ub1 *)
ol_dist_info,
                  24, SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curol1);
         quit ();
         exit (1);
      }

      /* order_line (not delivered) */

      if (obndrv (&curol2, (text *) ":ol_o_id", -1, (ub1 *) ol_o_id,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -
1)) {
         errrpt (&tpclda, &curol2);
         quit ();
         exit (1);
      }

      if (obndrv (&curol2, (text *) ":ol_d_id", -1, (ub1 *) ol_d_id,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -
1)) {
         errrpt (&tpclda, &curol2);
         quit ();
         exit (1);
      }

      if (obndrv (&curol2, (text *) ":ol_w_id", -1, (ub1 *) ol_w_id,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -
1)) {
         errrpt (&tpclda, &curol2);
         quit ();
         exit (1);
      }

      if (obndrv (&curol2, (text *) ":ol_number", -1, (ub1 *) ol_number,
          sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curol2);
         quit ();
         exit (1);
      }

      if (obndrv (&curol2, (text *) ":ol_i_id", -1, (ub1 *) ol_i_id,
```

```
          sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curol2);
         quit ();
         exit (1);
      }

      if (obndrv (&curol2, (text *) ":ol_supply_w_id", -1,
                  (ub1 *) ol_supply_w_id, sizeof (int), SQLT_INT, -1,
                  (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curol2);
         quit ();
         exit (1);
      }

      if (obndrv (&curol2, (text *) ":ol_amount", -1, (ub1 *) ol_amount,
          sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curol2);
         quit ();
         exit (1);
      }

      if (obndrv (&curol2, (text *) ":ol_dist_info", -1, (ub1 *)
ol_dist_info,
                  24, SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curol2);
         quit ();
         exit (1);
      }

      /* orders (delivered) */

      if (obndrv (&curo1, (text *) ":o_id", -1, (ub1 *) o_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo1);
         quit ();
         exit (1);
      }

      if (obndrv (&curo1, (text *) ":o_d_id", -1, (ub1 *) o_d_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo1);
         quit ();
         exit (1);
      }

      if (obndrv (&curo1, (text *) ":o_w_id", -1, (ub1 *) o_w_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo1);
         quit ();
         exit (1);
      }

      if (obndrv (&curo1, (text *) ":o_c_id", -1, (ub1 *) o_c_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo1);
         quit ();
         exit (1);
      }

      if (obndrv (&curo1, (text *) ":o_carrier_id", -1, (ub1 *)
o_carrier_id,
          sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo1);
         quit ();
         exit (1);
      }

      if (obndrv (&curo1, (text *) ":o_ol_cnt", -1, (ub1 *) o_ol_cnt,
          sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo1);
         quit ();
         exit (1);
      }

      /* orders (not delivered) */

      if (obndrv (&curo2, (text *) ":o_id", -1, (ub1 *) o_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo2);
         quit ();
         exit (1);
      }

      if (obndrv (&curo2, (text *) ":o_d_id", -1, (ub1 *) o_d_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo2);
         quit ();
         exit (1);
      }

      if (obndrv (&curo2, (text *) ":o_w_id", -1, (ub1 *) o_w_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo2);
         quit ();
         exit (1);
      }

      if (obndrv (&curo2, (text *) ":o_c_id", -1, (ub1 *) o_c_id, sizeof
(int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo2);
         quit ();
         exit (1);
      }

      if (obndrv (&curo2, (text *) ":o_ol_cnt", -1, (ub1 *) o_ol_cnt,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -
1)) {
         errrpt (&tpclda, &curo2);
         quit ();
         exit (1);
```

```
      }

      /* new order */

      if (obndrv (&curno, (text *) ":no_o_id", -1, (ub1 *) no_o_id,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -
1)) {
         errrpt (&tpclda, &curno);
         quit ();
         exit (1);
      }

      if (obndrv (&curno, (text *) ":no_d_id", -1, (ub1 *) no_d_id,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -
1)) {
         errrpt (&tpclda, &curno);
         quit ();
         exit (1);
      }

      if (obndrv (&curno, (text *) ":no_w_id", -1, (ub1 *) no_w_id,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -
1)) {
         errrpt (&tpclda, &curno);
         quit ();
         exit (1);
      }
   }

/*-------------------------------------------------------------+
 | Initialize random number generator                          |
 +-------------------------------------------------------------*/

   srand (SEED);
#ifndef ORA_NT
   srand48 (SEED);
#endif
   initperm ();


/*-------------------------------------------------------------+
 | Load the WAREHOUSE table.                                   |
 +-------------------------------------------------------------*/

   if (do_A || do_w) {
      nrows = eware - bware + 1;

      fprintf (stderr, "Loading/generating warehouse: w%d - w%d (%d
rows)\n",
               bware, eware, nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      for (loop = bware; loop <= eware; loop++) {

#ifdef SVRUNIX
         w_tax = (lrand48 () % 2001) * 0.0001;
#else
         w_tax = (lrand48 () % 2001);
#endif
         randstr (w_name, 6, 10);
         randstr (w_street_1, 10, 20);
         randstr (w_street_2, 10, 20);
         randstr (w_city, 10, 20);
         randstr (str2, 2, 2);
         randnum (num9, 9);
         num9[4] = num9[5] = num9[6] = num9[7] = num9[8] = '1';

         if (gen) {
#ifdef SVRUNIX
            printf ("%d 30000000 %6.4f %s %s %s %s %s %s\n", loop, w_tax,
#else
            printf ("%d 30000000 %d %s %s %s %s %s %s\n", loop, w_tax,
#endif
                    w_name, w_street_1, w_street_2, w_city, str2, num9);
            fflush (stdout);
         }
         else {
            w_id = loop;
            strncpy (w_state, str2, 2);
            strncpy (w_zip, num9, 9);

            if (oexec (&curw)) {
               errrpt (&tpclda, &curw);
               orol (&tpclda);
               fprintf (stderr, "Aborted at warehouse %d\n", loop);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at warehouse %d\n", loop);
               quit ();
               exit (1);
            }
         }
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec.
(%10.2f cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }

/*-------------------------------------------------------------+
 | Load the DISTRICT table.                                    |
 +-------------------------------------------------------------*/

   if (do_A || do_d) {
      nrows = (eware - bware + 1) * DISTFAC;

      fprintf (stderr, "Loading/generating district: w%d - w%d (%d
rows)\n",
               bware, eware, nrows);
```

```
      begin_time = gettime ();
      begin_cpu = getcpu ();

      dwid = bware - 1;

      for (row = 0; row < nrows; ) {
         dwid++;

         for (i = 0; i < DISTARR; i++, row++) {
#ifdef SVRUNIX
            d_tax[i] = (lrand48 () % 2001) * 0.0001;
#else
            d_tax[i] = (lrand48 () % 2001);
#endif
            randstr (d_name[i], 6, 10);
            randstr (d_street_1[i], 10, 20);
            randstr (d_street_2[i], 10, 20);
            randstr (d_city[i], 10, 20);
            randstr (str2, 2, 2);
            randnum (num9, 9);
            num9[4] = num9[5] = num9[6] = num9[7] = num9[8] = '1';

            if (gen) {
               /* printf ("%d %d %s %s %s %s %s %s %d 30000.0 3001\n",
                       i + 1, dwid, d_name[i], d_street_1[i],
d_street_2[i],
                       d_city[i], str2, num9, d_tax[i]); */
                      /* Reordered columns */
#ifdef SVRUNIX
               printf ("%d %d 3000000 %6.4f 3001 %s %s %s %s %s %s\n",
#else
               printf ("%d %d 3000000 %d 3001 %s %s %s %s %s %s\n",
#endif
                       i + 1, dwid, d_tax[i], d_name[i], d_street_1[i],
                      d_street_2[i], d_city[i], str2, num9 );
            }
            else {
               d_id[i] = i + 1;
               d_w_id[i] = dwid;
               strncpy (d_state[i], str2, 2);
               strncpy (d_zip[i], num9, 9);
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            if (oexn (&curd, DISTARR, 0)) {
               errrpt (&tpclda, &curd);
               orol (&tpclda);
               fprintf (stderr, "Aborted at warehouse %d, district 1\n",
dwid);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at warehouse %d, district 1\n",
dwid);
               quit ();
               exit (1);
            }
         }
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec.
(%10.2f cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }

/*-------------------------------------------------------------+
 | Load the CUSTOMER table.                                    |
 +-------------------------------------------------------------*/

   if (do_A || do_c) {
      nrows = (eware - bware + 1) * CUSTFAC * DISTFAC;

      fprintf (stderr, "Loading/generating customer: w%d - w%d (%d rows)\n
",
               bware, eware, nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      cid = 0;
      cdid = 1;
      cwid = bware;
      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < CUSTARR; i++, row++) {
            cid++;
            if (cid > CUSTFAC) {        /* cycle cust id */
               cid = 1;                 /* cheap mod */
               cdid++;                  /* shift district cycle */
               if (cdid > DISTFAC) {
                  cdid = 1;
                  cwid++;               /* shift warehouse cycle */
               }
            }
            c_id[i] = cid;
            c_d_id[i] = cdid;
            c_w_id[i] = cwid;
            if (cid <= 1000)
               randlastname (c_last[i], cid - 1);
            else
               randlastname (c_last[i], NURand (255, 0, 999, CNUM1));
            c_credit[i][1] = 'C';
            if (lrand48 () % 10)
               c_credit[i][0] = 'G';
            else
               c_credit[i][0] = 'B';
            c_discount[i] = (float)((lrand48 () % 5001) * 0.0001);
            randstr (c_first[i], 8, 16);
```

```
            randstr (c_street_1[i], 10, 20);
            randstr (c_street_2[i], 10, 20);
            randstr (c_city[i], 10, 20);
            randstr (str2, 2, 2);
            randnum (num9, 9);
            num9[4] = num9[5] = num9[6] = num9[7] = num9[8] = '1';
            randnum (num16, 16);
            randstr (c_data[i], 300, 500);

            if (gen) {
               printf ("%d %d %d %s OE %s %s %s %s %s %s %s %s %cC 5000000
%6.4f -1000 1000 1 0 %s\n",
                        cid, cdid, cwid, c_first[i], c_last[i],
                        c_street_1[i], c_street_2[i], c_city[i], str2,
num9,
                        num16, sdate, c_credit[i][0], c_discount[i],
c_data[i]);
            }
            else {
               strncpy (c_state[i], str2, 2);
               strncpy (c_zip[i], num9, 9);
               strncpy (c_phone[i], num16, 16);
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            if (oexn (&curc, CUSTARR, 0)) {
               errrpt (&tpclda, &curc);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, d_id %d, c_id %d\n",
                        c_w_id[0], c_d_id[0], c_id[0]);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, d_id %d, c_id %d\n",
                        c_w_id[0], c_d_id[0], c_id[0]);
               quit ();
               exit (1);
            }
         }

         if ((++loopcount) % 50)
            fprintf (stderr, ".");
         else
            fprintf (stderr, " %d rows committed\n   ", row);
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec.
(%10.2f cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }
```

```
/*-------------------------------------------------------------+
 | Load the ITEM table.                                        |
 +-------------------------------------------------------------*/

   if (do_A || do_i) {
      nrows = ITEMFAC;

      fprintf (stderr, "Loading/generating item: (%d rows)\n   ", nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < ITEMARR; i++, row++) {
            i_im_id[i] = (lrand48 () % 10000) + 1;
            i_price[i] = ((lrand48 () % 9901) + 100);
            randstr (i_name[i], 14, 24);
            randdatastr (i_data[i], 26, 50);

            if (gen) {
               printf ("%d %d %s %d %s\n", row + 1, i_im_id[i], i_name[i],
                        i_price[i], i_data[i]);
            }
            else {
               i_id[i] = row + 1;
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            if (oexn (&curi, ITEMARR, 0)) {
               errrpt (&tpclda, &curi);
               orol (&tpclda);
               fprintf (stderr, "Aborted at i_id %d\n", i_id[0]);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at i_id %d\n", i_id[0]);
               quit ();
               exit (1);
            }
         }

         if ((++loopcount) % 50)
            fprintf (stderr, ".");
         else
            fprintf (stderr, " %d rows committed\n   ", row);
      }

      end_time = gettime ();
```

```
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec.
(%10.2f cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }


/*-------------------------------------------------------------+
 | Load the STOCK table.                                       |
 +-------------------------------------------------------------*/

   if (do_A || do_s) {

      nrows = (eware - bware + 1) * STOCFAC;

      fprintf (stderr, "Loading/generating stock: w%d - w%d (%d rows)\n
",
               bware, eware, nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      sid = 0;
      swid = bware;
      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < STOCARR; i++, row++) {
            if (++sid > STOCFAC) {        /* cheap mod */
               sid = 1;
               swid++;
            }
            s_quantity[i] = (lrand48 () % 91) + 10;
            randstr (str24[0], 24, 24);
            randstr (str24[1], 24, 24);
            randstr (str24[2], 24, 24);
            randstr (str24[3], 24, 24);
            randstr (str24[4], 24, 24);
            randstr (str24[5], 24, 24);
            randstr (str24[6], 24, 24);
            randstr (str24[7], 24, 24);
            randstr (str24[8], 24, 24);
            randstr (str24[9], 24, 24);
            randdatastr (s_data[i], 26, 50);

            if (gen) {
               printf ("%d %d %d %s %s %s %s %s %s %s %s %s %s 0 0 0
%s\n",
                       sid, swid, s_quantity[i], str24[0], str24[1],
str24[2],
                       str24[3], str24[4], str24[5], str24[6], str24[7],
                       str24[8], str24[9], s_data[i]);
            }
            else {
               s_i_id[i] = sid;
               s_w_id[i] = swid;
               strncpy (s_dist_01[i], str24[0], 24);
               strncpy (s_dist_02[i], str24[1], 24);
               strncpy (s_dist_03[i], str24[2], 24);
               strncpy (s_dist_04[i], str24[3], 24);
               strncpy (s_dist_05[i], str24[4], 24);
               strncpy (s_dist_06[i], str24[5], 24);
               strncpy (s_dist_07[i], str24[6], 24);
               strncpy (s_dist_08[i], str24[7], 24);
               strncpy (s_dist_09[i], str24[8], 24);
               strncpy (s_dist_10[i], str24[9], 24);
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            if (oexn (&curs, STOCARR, 0)) {
               errrpt (&tpclda, &curs);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, s_i_id %d\n",
s_w_id[0],
                        s_i_id[0]);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, s_i_id %d\n",
s_w_id[0],
                        s_i_id[0]);
               quit ();
               exit (1);
            }
         }

         if ((++loopcount) % 50)
            fprintf (stderr, ".");
         else
            fprintf (stderr, " %d rows committed\n   ", row);
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec.
(%10.2f cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }

/*-------------------------------------------------------------+
 | Load the STOCK table (cluster around s_i_id).               |
 +-------------------------------------------------------------*/

   if (do_S) {

      nrows = (eitem - bitem + 1) * (eware - bware + 1);

      fprintf (stderr, "Loading/generating stock: i%d - i%d, w%d - w%d (%d
rows)\n   ",
               bitem, eitem, bware, eware, nrows);
```

```
      begin_time = gettime ();
      begin_cpu = getcpu ();

      sid = bitem;
      swid = bware - 1;
      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < STOCARR; i++, row++) {
            if (++swid > eware) {          /* cheap mod */
               swid = bware;
               sid++;
            }
            s_quantity[i] = (lrand48 () % 91) + 10;
            randstr (str24[0], 24, 24);
            randstr (str24[1], 24, 24);
            randstr (str24[2], 24, 24);
            randstr (str24[3], 24, 24);
            randstr (str24[4], 24, 24);
            randstr (str24[5], 24, 24);
            randstr (str24[6], 24, 24);
            randstr (str24[7], 24, 24);
            randstr (str24[8], 24, 24);
            randstr (str24[9], 24, 24);
            randdatastr (s_data[i], 26, 50);

            if (gen) {
               printf ("%d %d %d %s %s %s %s %s %s %s %s %s %s 0 0 0
%s\n",
                       sid, swid, s_quantity[i], str24[0], str24[1],
str24[2],
                       str24[3], str24[4], str24[5], str24[6], str24[7],
                       str24[8], str24[9], s_data[i]);
            }
            else {
               s_i_id[i] = sid;
               s_w_id[i] = swid;
               strncpy (s_dist_01[i], str24[0], 24);
               strncpy (s_dist_02[i], str24[1], 24);
               strncpy (s_dist_03[i], str24[2], 24);
               strncpy (s_dist_04[i], str24[3], 24);
               strncpy (s_dist_05[i], str24[4], 24);
               strncpy (s_dist_06[i], str24[5], 24);
               strncpy (s_dist_07[i], str24[6], 24);
               strncpy (s_dist_08[i], str24[7], 24);
               strncpy (s_dist_09[i], str24[8], 24);
               strncpy (s_dist_10[i], str24[9], 24);
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            if (oexn (&curs, STOCARR, 0)) {
               errrpt (&tpclda, &curs);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, s_i_id %d\n",
s_w_id[0],
                        s_i_id[0]);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, s_i_id %d\n",
s_w_id[0],
                        s_i_id[0]);
               quit ();
               exit (1);
            }
         }

         if ((++loopcount) % 50)
            fprintf (stderr, ".");
         else
            fprintf (stderr, " %d rows committed\n   ", row);
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec.
(%10.2f cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }

/*-------------------------------------------------------------+
 | Load the HISTORY table.                                     |
 +-------------------------------------------------------------*/

   if (do_A || do_h) {
      nrows = (eware - bware + 1) * HISTFAC;

      fprintf (stderr, "Loading/generating history: w%d - w%d (%d rows)\n
",
               bware, eware, nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      cid = 0;
      cdid = 1;
      cwid = bware;
      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < HISTARR; i++, row++) {
            cid++;
            if (cid > CUSTFAC) {        /* cycle cust id */
               cid = 1;                 /* cheap mod */
               cdid++;                  /* shift district cycle */
               if (cdid > DISTFAC) {
                  cdid = 1;
                  cwid++;               /* shift warehouse cycle */
               }
            }
            h_c_id[i] = cid;
```

```
            h_d_id[i] = cdid;
            h_w_id[i] = cwid;
            randstr (h_data[i], 12, 24);
            if (gen) {
               printf ("%d %d %d %d %d %s 1000 %s\n", cid, cdid, cwid,
cdid,
                       cwid, sdate, h_data[i]);
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            if (oexn (&curh, HISTARR, 0)) {
               errrpt (&tpclda, &curh);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, d_id %d, c_id %d\n",
                        h_w_id[0], h_d_id[0], h_c_id[0]);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, d_id %d, c_id %d\n",
                        h_w_id[0], h_d_id[0], h_c_id[0]);
               quit ();
               exit (1);
            }
         }

         if ((++loopcount) % 50)
            fprintf (stderr, ".");
         else
            fprintf (stderr, " %d rows committed\n   ", row);
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec.
(%10.2f cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }

/*-------------------------------------------------------------------+
 | Load the ORDERS and ORDER-LINE table.                             |
 +-------------------------------------------------------------------*/

   if (do_A || do_o) {
      nrows = (eware - bware + 1) * ORDEFAC * DISTFAC;

      fprintf (stderr, "Loading/generating orders and order-line: w%d -
w%d (%d ord, ~%d ordl)\n   ",
               bware, eware, nrows, nrows * 10);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      cid = 0;
      cdid = 1;
      cwid = bware;
      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < ORDEARR; i++, row++) {
            cid++;
            if (cid > ORDEFAC) {        /* cycle cust id */
               cid = 1;                 /* cheap mod */
               cdid++;                  /* shift district cycle */
               if (cdid > DISTFAC) {
                  cdid = 1;
                  cwid++;               /* shift warehouse cycle */
               }
            }
            o_carrier_id[i] = lrand48 () % 10 + 1;
            o_ol_cnt[i] = olcnt = lrand48 () % 11 + 5;

            if (gen) {
               if (cid < 2101) {
                  printf ("%d %d %d %d %s %d %d 1\n", cid, cdid, cwid,
                          randperm3000[cid - 1], sdate,o_carrier_id[i],
                          o_ol_cnt[i]);
               }
               else {
                 /* set carrierid to 11 instead of null */
                  printf ("%d %d %d %d %s 11 %d 1\n", cid, cdid, cwid,
                          randperm3000[cid - 1], sdate, o_ol_cnt[i]);
               }
            }
            else {
               o_id[i] = cid;
               o_d_id[i] = cdid;
               o_w_id[i] = cwid;
               o_c_id[i] = randperm3000[cid - 1];
            }

            for (j = 0; j < o_ol_cnt[i]; j++) {
               ol_i_id[j] = sid = lrand48 () % 100000 + 1;
               if (cid < 2101)
                  ol_amount[j] = 0;
               else
                  ol_amount[j] = (lrand48 () % 999999 + 1) ;
               randstr (str24[j], 24, 24);

               if (gen) {
                  if (cid < 2101) {
                     fprintf (olfp, "%d %d %d %d %s %d %d 5 %ld %s\n",
cid,
                              cdid, cwid, j + 1, sdate, ol_i_id[j], cwid,
                              ol_amount[j], str24[j]);
                  }
                  else {
                    /* Insert a default date instead of null date */
                     fprintf (olfp, "%d %d %d %d 01-Jan-1811 %d %d 5 %ld
%s\n", cid,
                              cdid, cwid, j + 1, ol_i_id[j], cwid,
                              ol_amount[j], str24[j]);
```

```
                }
             }
             else {
                ol_o_id[j] = cid;
                ol_d_id[j] = cdid;
                ol_w_id[j] = cwid;
                ol_number[j] = j + 1;
                ol_supply_w_id[j] = cwid;
                strncpy (ol_dist_info[j], str24[j], 24);
             }
          }

          if (gen) {
             fflush (olfp);
          }
          else {
             if (cid < 2101) {
                if (oexn (&curol1, olcnt, 0)) {
                   errrpt (&tpclda, &curol1);
                   orol (&tpclda);
                   fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id
%d\n",
                            cwid, cdid, cid);
                   quit ();
                   exit (1);
                }
                else if (ocom (&tpclda)) {
                   errrpt (&tpclda, &tpclda);
                   orol (&tpclda);
                   fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id
%d\n",
                           cwid, cdid, cid);
                   quit ();
                   exit (1);
                }
             }
             else {
                if (oexn (&curol2, olcnt, 0)) {
                   errrpt (&tpclda, &curol2);
                   orol (&tpclda);
                   fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id
%d\n",
                            cwid, cdid, cid);
                   quit ();
                   exit (1);
                }
                else if (ocom (&tpclda)) {
                   errrpt (&tpclda, &tpclda);
                   orol (&tpclda);
                   fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id
%d\n",
                           cwid, cdid, cid);
                   quit ();
                   exit (1);
                }
             }
          }
       }
```

```
       if (gen) {
          fflush (stdout);
       }
       else {
          if (cid < 2101) {
             if (oexn (&curo1, ORDEARR, 0)) {
                errrpt (&tpclda, &curo1);
                orol (&tpclda);
                fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n
",
                         cwid, cdid, cid);
                quit ();
                exit (1);
             }
             else if (ocom (&tpclda)) {
                errrpt (&tpclda, &tpclda);
                orol (&tpclda);
                fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n
",
                         cwid, cdid, cid);
                quit ();
                exit (1);
             }
          }
          else {
             if (oexn (&curo2, ORDEARR, 0)) {
                errrpt (&tpclda, &curo2);
                orol (&tpclda);
                fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n
",
                         cwid, cdid, cid);
                quit ();
                exit (1);
             }
             else if (ocom (&tpclda)) {
                errrpt (&tpclda, &tpclda);
                orol (&tpclda);
                fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n
",
                         cwid, cdid, cid);
                quit ();
                exit (1);
             }
          }
       }

       if ((++loopcount) % 50)
          fprintf (stderr, ".");
       else
          fprintf (stderr, " %d orders committed\n   ", row);
    }

    end_time = gettime ();
    end_cpu = getcpu ();
    fprintf (stderr, "Done.  %d orders loaded/generated in %10.2f sec.
(%10.2f cpu)\n\n",
             nrows, end_time - begin_time, end_cpu - begin_cpu);
 }
```

```
/*-------------------------------------------------------------+
 | Load the NEW-ORDER table.                                   |
 +-------------------------------------------------------------*/

   if (do_A || do_n) {
      nrows = (eware - bware + 1) * NEWOFAC * DISTFAC;

      fprintf (stderr, "Loading/generating new-order: w%d - w%d (%d
rows)\n   ",
               bware, eware, nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      cid = 0;
      cdid = 1;
      cwid = bware;
      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < NEWOARR; i++, row++) {
            cid++;
            if (cid > NEWOFAC) {
               cid = 1;
               cdid++;
               if (cdid > DISTFAC) {
                  cdid = 1;
                  cwid++;
               }
            }

            if (gen) {
               printf ("%d %d %d\n", cid + 2100, cdid, cwid);
            }
            else {
               no_o_id[i] = cid + 2100;
               no_d_id[i] = cdid;
               no_w_id[i] = cwid;
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            if (oexn (&curno, NEWOARR, 0)) {
               errrpt (&tpclda, &curno);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n
",
                        cwid, cdid, cid + 2100);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n
",
                        cwid, cdid, cid + 2100);
               quit ();
               exit (1);
            }
         }

         if ((++loopcount) % 45)
            fprintf (stderr, ".");
         else
            fprintf (stderr, " %d rows committed\n   ", row);
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec.
(%10.2f cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }

/*-------------------------------------------------------------+
 | clean up and exit.                                          |
 +-------------------------------------------------------------*/

   if (olfp)
      fclose (olfp);
   if (!gen)
      quit ();
   exit (0);

}



void initperm ()
{

   int i;
   int pos;
   int temp;

   /* init randperm3000 */

   for (i = 0; i < 3000; i++)
      randperm3000[i] = i + 1;
   for (i = 3000; i > 0; i--) {
      pos = lrand48 () % i;
      temp = randperm3000[i - 1];
      randperm3000[i - 1] = randperm3000[pos];
      randperm3000[pos] = temp;
   }

}



void randstr (str, x, y)

char *str;
int x;
```

```
int y;

{

   int i, j;
   int len;

   len = (lrand48 () % (y - x + 1)) + x;
   for (i = 0; i < len; i++) {
      j = lrand48 () % 62;
      if (j < 26)
         str[i] = (char) (j + 'a');
      else if (j < 52)
         str[i] = (char) (j - 26 + 'A');
      else
         str[i] = (char) (j - 52 + '0');
   }
   str[len] = '\0';

}



void randdatastr (str, x, y)

char *str;
int x;
int y;

{

   int i, j;
   int len;
   int pos;

   len = (lrand48 () % (y - x + 1)) + x;
   for (i = 0; i < len; i++) {
      j = lrand48 () % 62;
      if (j < 26)
         str[i] = (char) (j + 'a');
      else if (j < 52)
         str[i] = (char) (j - 26 + 'A');
      else
         str[i] = (char) (j - 52 + '0');
   }
   str[len] = '\0';
   if ((lrand48 () % 10) == 0) {
      pos = (lrand48 () % (len - 8));
      str[pos] = 'O';
      str[pos + 1] = 'R';
      str[pos + 2] = 'I';
      str[pos + 3] = 'G';
      str[pos + 4] = 'I';
      str[pos + 5] = 'N';
      str[pos + 6] = 'A';
      str[pos + 7] = 'L';
   }

}



void randnum (str, len)

char *str;
int len;

{

   int i;

   for (i = 0; i < len; i++)
      str[i] = (char) (lrand48 () % 10 + '0');
   str[len] = '\0';

}



void randlastname (str, id)

char *str;
int id;

{

   id = id % 1000;
   strcpy (str, lastname[id / 100]);
   strcat (str, lastname[(id / 10) % 10]);
   strcat (str, lastname[id % 10]);

}



int NURand (A, x, y, cnum)

int A, x, y, cnum;

{

   int a, b;

   a = lrand48 () % (A + 1);
   b = (lrand48 () % (y - x + 1)) + x;
   return ((((a | b) + cnum) % (y - x + 1)) + x);

}



void sysdate (sdate)

char *sdate;

{

   time_t tp;
```

```
   struct tm *tmptr;

   time (&tp);
   tmptr = localtime (&tp);
   strftime (sdate, 29, "%d-%b-%Y", tmptr);

}
```

load_stok.sh

```
# benchsetup 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                         All Rights Reserved                         |
#=====================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=====================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
LOAD_SCRIPTS=${BUILD_HOME}
OUTDIR=${BUILD_HOME}/outdir
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

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

I=1
SI=1
EI=5555
INC=5555
while [ $I -le 18 ]
do
   tpccload -M $MULT -S -j $SI -k $EI > ${OUTDIR}/stk${I}.out 2>&1 &
   I=`expr $I + 1`
   SI=`expr $SI + $INC`
   EI=`expr $EI + $INC`
done

tpccload -M $MULT -S -j 99991 -k 100000 > ${OUTDIR}/stk19.out 2>&1 &
wait
```

load_ordr.sh

```
#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                         All Rights Reserved                         |
#=====================================================================+
# FILENAME
#      pload.sh
# DESCRIPTION
#      Usage: load_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=====================================================================
#

BUILD_HOME=/oracle/8i/bench/tpc/tpcc/scripts/build3500
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
```

```
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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 "Bad 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 $BUILD_HOME ]
then
   mkdir $BUILD_HOME
fi

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

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

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

#
#  Load order and order-line table
#

#I=1
#while [ $I -le 18 ]
#do
#   mknod ${LDIR}/order${I}.dat p
#   mknod ${LDIR}/ordline${I}.dat p
#   I=`expr $I + 1`
#done

I=1
SW=1
EW=70
INC=70
while [ $I -le 25 ]
do
   tpccload -M $MULT -o ${LDIR}/ordline${I}.dat -b $SW -e $EW > \
      ${LDIR}/order${I}.dat 2> ${OUTDIR}/order${I}.out &
   I=`expr $I + 1`
   SW=`expr $SW + $INC`
   EW=`expr $EW + $INC`
done
wait
I=26
SW=1751
EW=1820
INC=70
while [ $I -le 50 ]
do
   tpccload -M $MULT -o ${LDIR}/ordline${I}.dat -b $SW -e $EW > \
      ${LDIR}/order${I}.dat 2> ${OUTDIR}/order${I}.out &
   I=`expr $I + 1`
   SW=`expr $SW + $INC`
   EW=`expr $EW + $INC`
done

#sleep 30
#
#I=1
#while [ $I -le 18 ]
#do
#
#   J=`expr $I % 18`
#   J=`expr $J + 1`
#   K=`expr $I % 1`
```

```
#   K=`expr $K + 1`
#   sqlldr tpcc/tpcc control=$TPCC_LOADER/order.ctl \
#          log=${OUTDIR}/order${I}.log \
#          bad=${OUTDIR}/order${I}.bad data=${LDIR}/order${I}.dat \
#          file=${ORACLE_HOME}/dbs/tpcc_disks/ordr_${K} \
#          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${I}.dat \
#          file=${ORACLE_HOME}/dbs/tpcc_disks/ordl_${J} \
#          discard=${OUTDIR}/ordline${I}.dsc &
#   I=`expr $I + 1`
#done

wait

#I=1
#while [ $I -le 18 ]
#do
#   rm -f ${LDIR}/order${I}.dat
#   rm -f ${LDIR}/ordline${I}.dat
#   I=`expr $I + 1`
#done
```

load_nord.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#==============================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA   |
#                    OPEN SYSTEMS PERFORMANCE GROUP            |
#                         All Rights Reserved                  |
#==============================================================+
# FILENAME
#      pload.sh
# DESCRIPTION
#      Usage: load_<obj>.sh [options]
#              -mu <multiplier>     (# of warehouses)
#==============================================================
#

BUILD_HOME=/oracle/8i/bench/tpc/tpcc/scripts/build3500
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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 "Bad 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 $BUILD_HOME ]
```

```
then
   mkdir $BUILD_HOME
fi

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

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

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

#
#   Load new-order table
#

#I=1
#while [ $I -le 4 ]
#do
#   mknod ${LDIR}/neword${I}.dat p
#   I=`expr $I + 1`
#done

I=1
SW=1
EW=350
INC=350
while [ $I -le 10 ]
do
  tpccload -M $MULT -n -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
#
#I=1
#while [ $I -le 10 ]
#do
#   J=`expr $I % 1`
#   J=`expr $J + 1`
#   sqlldr tpcc/tpcc control=$TPCC_LOADER/neword.ctl \
#        log=${OUTDIR}/neword${I}.log \
#        bad=${OUTDIR}/neword${I}.bad data=${LDIR}/neword${I}.dat \
#        file=${ORACLE_HOME}/dbs/tpcc_disks/nord_${J} \
#        discard=${OUTDIR}/neword${I}.dsc &
#   I=`expr $I + 1`
#done

wait
```

```
#I=1
#while [ $I -le 4 ]
#do
#   rm -f ${LDIR}/neword${I}.dat
#   I=`expr $I + 1`
#done
```

load_item.sh

```
#!/bin/ksh

# benchsetup 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                         All Rights Reserved                         |
#=====================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=====================================================================
#

BUILD_HOME=/oracle/8i/bench/tpc/tpcc/scripts/build3500
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
LOAD_SCRIPTS=${BUILD_HOME}
OUTDIR=${BUILD_HOME}/outdir
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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


tpccload -M $MULT -i
```

load_hist.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                     OPEN SYSTEMS PERFORMANCE GROUP                |
#                          All Rights Reserved                      |
#===================================================================+
# FILENAME
#      pload.sh
# DESCRIPTION
#      Usage: load_<obj>.sh [options]
#              -mu <multiplier>     (# of warehouses)
#====================================================================
#

BUILD_HOME=/oracle/8i/bench/tpc/tpcc/scripts/build3500
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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 "Bad 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 $BUILD_HOME ]
then
   mkdir $BUILD_HOME
fi

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

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

if [ ! -d $OUTDIR ]
then
   mkdir $OUTDIR
fi
#
#  Load history table
#

I=1
while [ $I -le 5 ]
```

```
do
   mknod ${LDIR}/hist${I}.dat p
   I=`expr $I + 1`
done

I=1
SW=1
EW=175
INC=175
while [ $I -le 5 ]
do
   tpccload -M $MULT -h -g -b $SW -e $EW > ${LDIR}/hist${I}.dat &
   I=`expr $I + 1`
   SW=`expr $SW + $INC`
   EW=`expr $EW + $INC`
done

sleep 30

I=1
while [ $I -le 5 ]
do
   J=`expr $I % 1`
   J=`expr $J + 1`
   sqlldr tpcc/tpcc control=$TPCC_LOADER/hist.ctl \
          log=${OUTDIR}/hist${I}.log \
          bad=${OUTDIR}/hist${I}.bad \
          data=${LDIR}/hist${I}.dat \
          file=$ORACLE_HOME/dbs/tpcc_disks/hist0${I} \
          discard=${OUTDIR}/hist${I}.dsc &
   I=`expr $I + 1`
done


wait

I=1
while [ $I -le 5 ]
do
   rm -f ${LDIR}/hist${I}.dat
   I=`expr $I + 1`
done


I=6
while [ $I -le 10 ]
do
   mknod ${LDIR}/hist${I}.dat p
   I=`expr $I + 1`
done

I=6
SW=876
EW=1050
INC=175
while [ $I -le 10 ]
do
   tpccload -M $MULT -h -g -b $SW -e $EW > ${LDIR}/hist${I}.dat &
   I=`expr $I + 1`
   SW=`expr $SW + $INC`
   EW=`expr $EW + $INC`
done

sleep 30

I=6
while [ $I -le 9 ]
do
   J=`expr $I % 1`
   J=`expr $J + 1`
   sqlldr tpcc/tpcc control=$TPCC_LOADER/hist.ctl \
          log=${OUTDIR}/hist${I}.log \
          bad=${OUTDIR}/hist${I}.bad \
          data=${LDIR}/hist${I}.dat \
          file=$ORACLE_HOME/dbs/tpcc_disks/hist0${I} \
          discard=${OUTDIR}/hist${I}.dsc &
   I=`expr $I + 1`
done
   sqlldr tpcc/tpcc control=$TPCC_LOADER/hist.ctl \
          log=${OUTDIR}/hist10.log \
          bad=${OUTDIR}/hist10.bad \
          data=${LDIR}/hist10.dat \
          file=$ORACLE_HOME/dbs/tpcc_disks/hist10 \
          discard=${OUTDIR}/hist10.dsc &

wait

I=6
while [ $I -le 10 ]
do
   rm -f ${LDIR}/hist${I}.dat
   I=`expr $I + 1`
done



I=11
while [ $I -le 15 ]
do
   mknod ${LDIR}/hist${I}.dat p
   I=`expr $I + 1`
done

I=11
SW=1751
EW=1925
INC=175
while [ $I -le 15 ]
do
   tpccload -M $MULT -h -g -b $SW -e $EW > ${LDIR}/hist${I}.dat &
   I=`expr $I + 1`
   SW=`expr $SW + $INC`
   EW=`expr $EW + $INC`
done

sleep 30

I=11
```

```
while [ $I -le 15 ]
do
   J=`expr $I % 1`
   J=`expr $J + 1`
   sqlldr tpcc/tpcc control=$TPCC_LOADER/hist.ctl \
          log=${OUTDIR}/hist${I}.log \
          bad=${OUTDIR}/hist${I}.bad \
          data=${LDIR}/hist${I}.dat \
          file=$ORACLE_HOME/dbs/tpcc_disks/hist${I} \
          discard=${OUTDIR}/hist${I}.dsc &
   I=`expr $I + 1`
done

wait

I=16
while [ $I -le 20 ]
do
   mknod ${LDIR}/hist${I}.dat p
   I=`expr $I + 1`
done

I=16
SW=2626
EW=2800
INC=175
while [ $I -le 20 ]
do
   tpccload -M $MULT -h -g -b $SW -e $EW > ${LDIR}/hist${I}.dat &
   I=`expr $I + 1`
   SW=`expr $SW + $INC`
   EW=`expr $EW + $INC`
done

sleep 30

I=16
while [ $I -le 20 ]
do
   J=`expr $I % 1`
   J=`expr $J + 1`
   sqlldr tpcc/tpcc control=$TPCC_LOADER/hist.ctl \
          log=${OUTDIR}/hist${I}.log \
          bad=${OUTDIR}/hist${I}.bad \
          data=${LDIR}/hist${I}.dat \
          file=$ORACLE_HOME/dbs/tpcc_disks/hist${I} \
          discard=${OUTDIR}/hist${I}.dsc &
   I=`expr $I + 1`
done
```

load_dist.sh

```
#!/bin/ksh

# benchsetup 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle
#
#
#+=================================================================+
#|        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#|                     OPEN SYSTEMS PERFORMANCE GROUP              |
#|                          All Rights Reserved                    |
#+=================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#==================================================================
#

BUILD_HOME=/oracle/8i/bench/tpc/tpcc/scripts/build3500
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
LOAD_SCRIPTS=${BUILD_HOME}
OUTDIR=${BUILD_HOME}/outdir
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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


tpccload -M $MULT -d
```

load_cust.sh

```
#!/bin/ksh

# benchsetup 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle
#
#
#+=================================================================+
#|        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#|                     OPEN SYSTEMS PERFORMANCE GROUP              |
#|                          All Rights Reserved                    |
#+=================================================================+
```

```
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=================================================================
#

BUILD_HOME=/oracle/8i/bench/tpc/tpcc/scripts/build3500
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
LOAD_SCRIPTS=${BUILD_HOME}
OUTDIR=${BUILD_HOME}/outdir
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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


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

I=1
SW=1
EW=70
INC=70
while [ $I -le 25 ]
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
wait
I=26
SW=1751
EW=1820
INC=70
while [ $I -le 50 ]
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
```

load_ware.sh

```
#!/bin/ksh

# benchsetup 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=================================================================
#

BUILD_HOME=/oracle/8i/bench/tpc/tpcc/scripts/build3500
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
LOAD_SCRIPTS=${BUILD_HOME}
OUTDIR=${BUILD_HOME}/outdir
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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


tpccload -M $MULT -w
```

hist.ctl

```
--
-- $Header: hist.ctl 7030100.1 95/08/07 15:47:23 plai Generic<base> $
Copyr (c) 1994 Oracle
--
-- +---------------------------------------------------------------+
--         Copyright (c) 1994  Oracle Corp, Redwood Shores, CA       |
--                      OPEN SYSTEMS PERFORMANCE GROUP               |
--                           All Rights Reserved                     |
-- +---------------------------------------------------------------+
--  FILENAME
--      hist.ctl
--  DESCRIPTION
--      This is a SQL*Loader control file.  It is used for
--      loading history rows to the tpcc database.
--  USAGE
--      sqlldr[st] <user_name>/<password> <SQL*Loader control file>
-- +---------------------------------------------------------------+*/

OPTIONS (DIRECT = TRUE, PARALLEL = TRUE)

UNRECOVERABLE

LOAD DATA
APPEND

INTO TABLE history
APPEND
FIELDS TERMINATED BY WHITESPACE OPTIONALLY ENCLOSED BY '"'
(
   h_c_id              integer external,
   h_c_d_id            integer external,
   h_c_w_id            integer external,
   h_d_id              integer external,
   h_w_id              integer external,
   h_date              date "DD-Mon-YYYY",
   h_amount            integer external,
   h_data              char(24)
)
```

Symbolic Data File Link Creation Scripts

create_data_link_new.sh

```
./c1_data_new.ln
./c2_data_new.ln
./c3_data_new.ln
./c4_data.ln
./c5_data_new.ln
./log.ln
```

c5_data.ln

```
#####################################################
#            Link                                   #
#####################################################
ln -s /dev/rdsk/c3b0t1d8s1    stok49
ln -s /dev/rdsk/c3b0t1d8s2    cust49
ln -s /dev/rdsk/c3b0t1d8s3    ordl49
ln -s /dev/rdsk/c3b0t1d8s5    ordr25
ln -s /dev/rdsk/c3b0t1d8s9    icust2_49
ln -s /dev/rdsk/c3b0t1d8sa    iord1_49
ln -s /dev/rdsk/c3b0t1d8sb    iord2_49
ln -s /dev/rdsk/c3b0t1d8sc    unused49
ln -s /dev/rdsk/c3b0t1d8se    roll01
#
ln -s /dev/rdsk/c5b0t0d3s1    stok50
ln -s /dev/rdsk/c5b0t0d3s2    cust50
ln -s /dev/rdsk/c5b0t0d3s3    ordl50
ln -s /dev/rdsk/c5b0t0d3s5    ordr26
ln -s /dev/rdsk/c5b0t0d3s9    icust2_50
ln -s /dev/rdsk/c5b0t0d3sa    iord1_50
ln -s /dev/rdsk/c5b0t0d3sb    iord2_50
ln -s /dev/rdsk/c5b0t0d3sc    unused50
#
ln -s /dev/rdsk/c5b0t0d4s1    stok51
ln -s /dev/rdsk/c5b0t0d4s2    cust51
ln -s /dev/rdsk/c5b0t0d4s3    ordl51
ln -s /dev/rdsk/c5b0t0d4s4    misc25
ln -s /dev/rdsk/c5b0t0d4s5    ordr27
ln -s /dev/rdsk/c5b0t0d4s9    icust2_51
ln -s /dev/rdsk/c5b0t0d4sa    iord1_51
ln -s /dev/rdsk/c5b0t0d4sb    iord2_51
ln -s /dev/rdsk/c5b0t0d4sc    unused51
ln -s /dev/rdsk/c5b0t0d4sd    hist17
#
ln -s /dev/rdsk/c5b0t0d5s1    stok52
ln -s /dev/rdsk/c5b0t0d5s2    cust52
ln -s /dev/rdsk/c5b0t0d5s3    ordl52
ln -s /dev/rdsk/c5b0t0d5s6    nord25
ln -s /dev/rdsk/c5b0t0d5s9    icust2_52
ln -s /dev/rdsk/c5b0t0d5sa    iord1_52
ln -s /dev/rdsk/c5b0t0d5sb    iord2_52
ln -s /dev/rdsk/c5b0t0d5sc    unused52
ln -s /dev/rdsk/c5b0t0d5sd    hist18
#
ln -s /dev/rdsk/c5b0t0d6s1    stok53
ln -s /dev/rdsk/c5b0t0d6s2    cust53
ln -s /dev/rdsk/c5b0t0d6s3    ordl53
ln -s /dev/rdsk/c5b0t0d6s4    misc26
ln -s /dev/rdsk/c5b0t0d6s6    nord26
ln -s /dev/rdsk/c5b0t0d6s9    icust2_53
ln -s /dev/rdsk/c5b0t0d6sa    iord1_53
ln -s /dev/rdsk/c5b0t0d6sb    iord2_53
ln -s /dev/rdsk/c5b0t0d6sc    unused53
#
ln -s /dev/rdsk/c5b0t0d7s1    stok54
ln -s /dev/rdsk/c5b0t0d7s2    cust54
```

```
ln -s /dev/rdsk/c5b0t0d7s3    ordl54
ln -s /dev/rdsk/c5b0t0d7s4    misc27
ln -s /dev/rdsk/c5b0t0d7s6    nord27
ln -s /dev/rdsk/c5b0t0d7s9    icust2_54
ln -s /dev/rdsk/c5b0t0d7sa    iord1_54
ln -s /dev/rdsk/c5b0t0d7sb    iord2_54
ln -s /dev/rdsk/c5b0t0d7sc    unused54
#
ln -s /dev/rdsk/c5b0t1d8s1    stok55
ln -s /dev/rdsk/c5b0t1d8s2    cust55
ln -s /dev/rdsk/c5b0t1d8s3    ordl55
ln -s /dev/rdsk/c5b0t1d8s5    ordr28
ln -s /dev/rdsk/c5b0t1d8s9    icust2_55
ln -s /dev/rdsk/c5b0t1d8sa    iord1_55
ln -s /dev/rdsk/c5b0t1d8sb    iord2_55
ln -s /dev/rdsk/c5b0t1d8sc    unused55
#
ln -s /dev/rdsk/c5b0t1d9s1    stok56
ln -s /dev/rdsk/c5b0t1d9s2    cust56
ln -s /dev/rdsk/c5b0t1d9s3    ordl56
ln -s /dev/rdsk/c5b0t1d9s5    ordr29
ln -s /dev/rdsk/c5b0t1d9s9    icust2_56
ln -s /dev/rdsk/c5b0t1d9sa    iord1_56
ln -s /dev/rdsk/c5b0t1d9sb    iord2_56
ln -s /dev/rdsk/c5b0t1d9sc    unused56
#
ln -s /dev/rdsk/c5b0t1d10s1   stok57
ln -s /dev/rdsk/c5b0t1d10s2   cust57
ln -s /dev/rdsk/c5b0t1d10s3   ordl57
ln -s /dev/rdsk/c5b0t1d10s4   misc28
ln -s /dev/rdsk/c5b0t1d10s5   ordr30
ln -s /dev/rdsk/c5b0t1d10s9   icust2_57
ln -s /dev/rdsk/c5b0t1d10sa   iord1_57
ln -s /dev/rdsk/c5b0t1d10sb   iord2_57
ln -s /dev/rdsk/c5b0t1d10sc   unused57
ln -s /dev/rdsk/c5b0t1d10sd   hist19
#
ln -s /dev/rdsk/c5b0t1d11s1   stok58
ln -s /dev/rdsk/c5b0t1d11s2   cust58
ln -s /dev/rdsk/c5b0t1d11s3   ordl58
ln -s /dev/rdsk/c5b0t1d11s6   nord28
ln -s /dev/rdsk/c5b0t1d11s9   icust2_58
ln -s /dev/rdsk/c5b0t1d11sa   iord1_58
ln -s /dev/rdsk/c5b0t1d11sb   iord2_58
ln -s /dev/rdsk/c5b0t1d11sc   unused58
ln -s /dev/rdsk/c5b0t1d11sd   hist20
#
ln -s /dev/rdsk/c5b0t1d12s1   stok59
ln -s /dev/rdsk/c5b0t1d12s2   cust59
ln -s /dev/rdsk/c5b0t1d12s3   ordl59
ln -s /dev/rdsk/c5b0t1d12s4   misc29
ln -s /dev/rdsk/c5b0t1d12s6   nord29
ln -s /dev/rdsk/c5b0t1d12s9   icust2_59
ln -s /dev/rdsk/c5b0t1d12sa   iord1_59
ln -s /dev/rdsk/c5b0t1d12sb   iord2_59
ln -s /dev/rdsk/c5b0t1d12sc   unused59
#
ln -s /dev/rdsk/c5b0t1d13s1   stok60
ln -s /dev/rdsk/c5b0t1d13s2   cust60
ln -s /dev/rdsk/c5b0t1d13s3   ordl60
ln -s /dev/rdsk/c5b0t1d13s4   misc30
ln -s /dev/rdsk/c5b0t1d13s6   nord30
ln -s /dev/rdsk/c5b0t1d13s9   icust2_60
ln -s /dev/rdsk/c5b0t1d13sa   iord1_60
ln -s /dev/rdsk/c5b0t1d13sb   iord2_60
ln -s /dev/rdsk/c5b0t1d13sc   unused60
```

c1_data_new.ln

```
#########################################################
#  Script For Creating Links                            #
#########################################################
ln -s /dev/rdsk/c1b0t0d2s1    stok01
ln -s /dev/rdsk/c1b0t0d2s2    cust01
ln -s /dev/rdsk/c1b0t0d2s3    ordl01
ln -s /dev/rdsk/c1b0t0d2s5    ordr01
ln -s /dev/rdsk/c1b0t0d2s9    icust2_01
ln -s /dev/rdsk/c1b0t0d2sa    iord1_01
ln -s /dev/rdsk/c1b0t0d2sb    iord2_01
ln -s /dev/rdsk/c1b0t0d2sc    unused01
ln -s /dev/rdsk/c1b0t0d2se    temp01
#
ln -s /dev/rdsk/c1b0t0d3s1    stok02
ln -s /dev/rdsk/c1b0t0d3s2    cust02
ln -s /dev/rdsk/c1b0t0d3s3    ordl02
ln -s /dev/rdsk/c1b0t0d3s5    ordr02
ln -s /dev/rdsk/c1b0t0d3s9    icust2_02
ln -s /dev/rdsk/c1b0t0d3sa    iord1_02
ln -s /dev/rdsk/c1b0t0d3sb    iord2_02
ln -s /dev/rdsk/c1b0t0d3sc    unused02
ln -s /dev/rdsk/c1b0t0d3se    temp02

ln -s /dev/rdsk/c1b0t0d4s1    stok03
ln -s /dev/rdsk/c1b0t0d4s2    cust03
ln -s /dev/rdsk/c1b0t0d4s3    ordl03
ln -s /dev/rdsk/c1b0t0d4s4    misc01
ln -s /dev/rdsk/c1b0t0d4s5    ordr03
ln -s /dev/rdsk/c1b0t0d4s9    icust2_03
ln -s /dev/rdsk/c1b0t0d4sa    iord1_03
ln -s /dev/rdsk/c1b0t0d4sb    iord2_03
ln -s /dev/rdsk/c1b0t0d4sc    unused03
ln -s /dev/rdsk/c1b0t0d4sd    hist01

ln -s /dev/rdsk/c1b0t0d5s1    stok04
ln -s /dev/rdsk/c1b0t0d5s2    cust04
ln -s /dev/rdsk/c1b0t0d5s3    ordl04
ln -s /dev/rdsk/c1b0t0d5s5    ctrl1
ln -s /dev/rdsk/c1b0t0d5s6    nord01
ln -s /dev/rdsk/c1b0t0d5s9    icust2_04
ln -s /dev/rdsk/c1b0t0d5sa    iord1_04
ln -s /dev/rdsk/c1b0t0d5sb    iord2_04
ln -s /dev/rdsk/c1b0t0d5sc    unused04
ln -s /dev/rdsk/c1b0t0d5sd    hist02

ln -s /dev/rdsk/c2b0t1d10s1   stok05
```

```
ln -s /dev/rdsk/c2b0t1d10s2   cust05
ln -s /dev/rdsk/c2b0t1d10s3   ordl05
ln -s /dev/rdsk/c2b0t1d10s4   misc02
ln -s /dev/rdsk/c2b0t1d10s5   ctrl2
ln -s /dev/rdsk/c2b0t1d10s6   nord02
ln -s /dev/rdsk/c2b0t1d10s9   icust2_05
ln -s /dev/rdsk/c2b0t1d10sa   iord1_05
ln -s /dev/rdsk/c2b0t1d10sb   iord2_05
ln -s /dev/rdsk/c2b0t1d10sc   unused05

ln -s /dev/rdsk/c2b0t1d11s1   stok06
ln -s /dev/rdsk/c2b0t1d11s2   cust06
ln -s /dev/rdsk/c2b0t1d11s3   ordl06
ln -s /dev/rdsk/c2b0t1d11s4   misc03
ln -s /dev/rdsk/c2b0t1d11s6   nord03
ln -s /dev/rdsk/c2b0t1d11s9   icust2_06
ln -s /dev/rdsk/c2b0t1d11sa   iord1_06
ln -s /dev/rdsk/c2b0t1d11sb   iord2_06
ln -s /dev/rdsk/c2b0t1d11sc   unused06

ln -s /dev/rdsk/c2b0t0d4s1    stok07
ln -s /dev/rdsk/c2b0t0d4s2    cust07
ln -s /dev/rdsk/c2b0t0d4s3    ordl07
ln -s /dev/rdsk/c2b0t0d4s5    ordr04
ln -s /dev/rdsk/c2b0t0d4s9    icust2_07
ln -s /dev/rdsk/c2b0t0d4sa    iord1_07
ln -s /dev/rdsk/c2b0t0d4sb    iord2_07
ln -s /dev/rdsk/c2b0t0d4sc    unused07
ln -s /dev/rdsk/c2b0t0d4se    temp03

ln -s /dev/rdsk/c2b0t0d5s1    stok08
ln -s /dev/rdsk/c2b0t0d5s2    cust08
ln -s /dev/rdsk/c2b0t0d5s3    ordl08
ln -s /dev/rdsk/c2b0t0d5s5    ordr05
ln -s /dev/rdsk/c2b0t0d5s9    icust2_08
ln -s /dev/rdsk/c2b0t0d5sa    iord1_08
ln -s /dev/rdsk/c2b0t0d5sb    iord2_08
ln -s /dev/rdsk/c2b0t0d5sc    unused08
ln -s /dev/rdsk/c2b0t0d5se    temp04

ln -s /dev/rdsk/c1b0t1d10s1   stok09
ln -s /dev/rdsk/c1b0t1d10s2   cust09
ln -s /dev/rdsk/c1b0t1d10s3   ordl09
ln -s /dev/rdsk/c1b0t1d10s4   misc04
ln -s /dev/rdsk/c1b0t1d10s5   ordr06
ln -s /dev/rdsk/c1b0t1d10s9   icust2_09
ln -s /dev/rdsk/c1b0t1d10sa   iord1_09
ln -s /dev/rdsk/c1b0t1d10sb   iord2_09
ln -s /dev/rdsk/c1b0t1d10sc   unused09
ln -s /dev/rdsk/c1b0t1d10sd   hist03

ln -s /dev/rdsk/c1b0t1d11s1   stok10
ln -s /dev/rdsk/c1b0t1d11s2   cust10
ln -s /dev/rdsk/c1b0t1d11s3   ordl10
ln -s /dev/rdsk/c1b0t1d11s6   nord04
ln -s /dev/rdsk/c1b0t1d11s9   icust2_10
ln -s /dev/rdsk/c1b0t1d11sa   iord1_10
ln -s /dev/rdsk/c1b0t1d11sb   iord2_10
ln -s /dev/rdsk/c1b0t1d11sc   unused10
ln -s /dev/rdsk/c1b0t1d11sd   hist04

ln -s /dev/rdsk/c1b0t1d12s1   stok11
ln -s /dev/rdsk/c1b0t1d12s2   cust11
ln -s /dev/rdsk/c1b0t1d12s3   ordl11
ln -s /dev/rdsk/c1b0t1d12s4   misc05
ln -s /dev/rdsk/c1b0t1d12s6   nord05
ln -s /dev/rdsk/c1b0t1d12s9   icust2_11
ln -s /dev/rdsk/c1b0t1d12sa   iord1_11
ln -s /dev/rdsk/c1b0t1d12sb   iord2_11
ln -s /dev/rdsk/c1b0t1d12sc   unused11

ln -s /dev/rdsk/c1b0t1d13s1   stok12
ln -s /dev/rdsk/c1b0t1d13s2   cust12
ln -s /dev/rdsk/c1b0t1d13s3   ordl12
ln -s /dev/rdsk/c1b0t1d13s4   misc06
ln -s /dev/rdsk/c1b0t1d13s6   nord06
ln -s /dev/rdsk/c1b0t1d13s9   icust2_12
ln -s /dev/rdsk/c1b0t1d13sa   iord1_12
ln -s /dev/rdsk/c1b0t1d13sb   iord2_12
ln -s /dev/rdsk/c1b0t1d13sc   unused12
```

c2_data_new.ln

```
################################################
#       Link                                   #
################################################
ln -s /dev/rdsk/c2b0t0d2s1    stok13
ln -s /dev/rdsk/c2b0t0d2s2    cust13
ln -s /dev/rdsk/c2b0t0d2s3    ordl13
ln -s /dev/rdsk/c2b0t0d2s5    ordr07
ln -s /dev/rdsk/c2b0t0d2s9    icust2_13
ln -s /dev/rdsk/c2b0t0d2sa    iord1_13
ln -s /dev/rdsk/c2b0t0d2sb    iord2_13
ln -s /dev/rdsk/c2b0t0d2sc    unused13
ln -s /dev/rdsk/c2b0t0d2se    temp05
#
ln -s /dev/rdsk/c2b0t0d3s1    stok14
ln -s /dev/rdsk/c2b0t0d3s2    cust14
ln -s /dev/rdsk/c2b0t0d3s3    ordl14
ln -s /dev/rdsk/c2b0t0d3s5    ordr08
ln -s /dev/rdsk/c2b0t0d3s9    icust2_14
ln -s /dev/rdsk/c2b0t0d3sa    iord1_14
ln -s /dev/rdsk/c2b0t0d3sb    iord2_14
ln -s /dev/rdsk/c2b0t0d3sc    unused14
ln -s /dev/rdsk/c2b0t0d3se    temp06
#
ln -s /dev/rdsk/c1b0t1d8s1    stok15
ln -s /dev/rdsk/c1b0t1d8s2    cust15
ln -s /dev/rdsk/c1b0t1d8s3    ordl15
ln -s /dev/rdsk/c1b0t1d8s4    misc07
ln -s /dev/rdsk/c1b0t1d8s5    ordr09
ln -s /dev/rdsk/c1b0t1d8s9    icust2_15
ln -s /dev/rdsk/c1b0t1d8sa    iord1_15
ln -s /dev/rdsk/c1b0t1d8sb    iord2_15
```

```
ln -s /dev/rdsk/c1b0t1d8sc    unused15
ln -s /dev/rdsk/c1b0t1d8sd    hist05
#
ln -s /dev/rdsk/c1b0t1d9s1    stok16
ln -s /dev/rdsk/c1b0t1d9s2    cust16
ln -s /dev/rdsk/c1b0t1d9s3    ordl16
ln -s /dev/rdsk/c1b0t1d9s6    nord07
ln -s /dev/rdsk/c1b0t1d9s9    icust2_16
ln -s /dev/rdsk/c1b0t1d9sa    iord1_16
ln -s /dev/rdsk/c1b0t1d9sb    iord2_16
ln -s /dev/rdsk/c1b0t1d9sc    unused16
ln -s /dev/rdsk/c1b0t1d9sd    hist06
#
ln -s /dev/rdsk/c2b0t0d6s1    stok17
ln -s /dev/rdsk/c2b0t0d6s2    cust17
ln -s /dev/rdsk/c2b0t0d6s3    ordl17
ln -s /dev/rdsk/c2b0t0d6s4    misc08
ln -s /dev/rdsk/c2b0t0d6s6    nord08
ln -s /dev/rdsk/c2b0t0d6s9    icust2_17
ln -s /dev/rdsk/c2b0t0d6sa    iord1_17
ln -s /dev/rdsk/c2b0t0d6sb    iord2_17
ln -s /dev/rdsk/c2b0t0d6sc    unused17
#
ln -s /dev/rdsk/c2b0t0d7s1    stok18
ln -s /dev/rdsk/c2b0t0d7s2    cust18
ln -s /dev/rdsk/c2b0t0d7s3    ordl18
ln -s /dev/rdsk/c2b0t0d7s4    misc09
ln -s /dev/rdsk/c2b0t0d7s6    nord09
ln -s /dev/rdsk/c2b0t0d7s9    icust2_18
ln -s /dev/rdsk/c2b0t0d7sa    iord1_18
ln -s /dev/rdsk/c2b0t0d7sb    iord2_18
ln -s /dev/rdsk/c2b0t0d7sc    unused18
#
ln -s /dev/rdsk/c2b0t1d8s1    stok19
ln -s /dev/rdsk/c2b0t1d8s2    cust19
ln -s /dev/rdsk/c2b0t1d8s3    ordl19
ln -s /dev/rdsk/c2b0t1d8s5    ordr10
ln -s /dev/rdsk/c2b0t1d8s9    icust2_19
ln -s /dev/rdsk/c2b0t1d8sa    iord1_19
ln -s /dev/rdsk/c2b0t1d8sb    iord2_19
ln -s /dev/rdsk/c2b0t1d8sc    unused19
ln -s /dev/rdsk/c2b0t1d8se    temp07
#
ln -s /dev/rdsk/c2b0t1d9s1    stok20
ln -s /dev/rdsk/c2b0t1d9s2    cust20
ln -s /dev/rdsk/c2b0t1d9s3    ordl20
ln -s /dev/rdsk/c2b0t1d9s5    ordr11
ln -s /dev/rdsk/c2b0t1d9s9    icust2_20
ln -s /dev/rdsk/c2b0t1d9sa    iord1_20
ln -s /dev/rdsk/c2b0t1d9sb    iord2_20
ln -s /dev/rdsk/c2b0t1d9sc    unused20
ln -s /dev/rdsk/c2b0t1d9se    temp08
#
ln -s /dev/rdsk/c1b0t0d6s1    stok21
ln -s /dev/rdsk/c1b0t0d6s2    cust21
ln -s /dev/rdsk/c1b0t0d6s3    ordl21
ln -s /dev/rdsk/c1b0t0d6s4    misc10
ln -s /dev/rdsk/c1b0t0d6s5    ordr12
ln -s /dev/rdsk/c1b0t0d6s9    icust2_21
ln -s /dev/rdsk/c1b0t0d6sa    iord1_21
ln -s /dev/rdsk/c1b0t0d6sb    iord2_21
ln -s /dev/rdsk/c1b0t0d6sc    unused21
ln -s /dev/rdsk/c1b0t0d6sd    hist07
#
ln -s /dev/rdsk/c1b0t0d7s1    stok22
ln -s /dev/rdsk/c1b0t0d7s2    cust22
ln -s /dev/rdsk/c1b0t0d7s3    ordl22
ln -s /dev/rdsk/c1b0t0d7s6    nord10
ln -s /dev/rdsk/c1b0t0d7s9    icust2_22
ln -s /dev/rdsk/c1b0t0d7sa    iord1_22
ln -s /dev/rdsk/c1b0t0d7sb    iord2_22
ln -s /dev/rdsk/c1b0t0d7sc    unused22
ln -s /dev/rdsk/c1b0t0d7sd    hist08
#
ln -s /dev/rdsk/c2b0t1d12s1   stok23
ln -s /dev/rdsk/c2b0t1d12s2   cust23
ln -s /dev/rdsk/c2b0t1d12s3   ordl23
ln -s /dev/rdsk/c2b0t1d12s4   misc11
ln -s /dev/rdsk/c2b0t1d12s6   nord11
ln -s /dev/rdsk/c2b0t1d12s9   icust2_23
ln -s /dev/rdsk/c2b0t1d12sa   iord1_23
ln -s /dev/rdsk/c2b0t1d12sb   iord2_23
ln -s /dev/rdsk/c2b0t1d12sc   unused23
#
ln -s /dev/rdsk/c2b0t1d13s1   stok24
ln -s /dev/rdsk/c2b0t1d13s2   cust24
ln -s /dev/rdsk/c2b0t1d13s3   ordl24
ln -s /dev/rdsk/c2b0t1d13s4   misc12
ln -s /dev/rdsk/c2b0t1d13s6   nord12
ln -s /dev/rdsk/c2b0t1d13s9   icust2_24
ln -s /dev/rdsk/c2b0t1d13sa   iord1_24
ln -s /dev/rdsk/c2b0t1d13sb   iord2_24
ln -s /dev/rdsk/c2b0t1d13sc   unused24
```

c3_data_new.ln

```
################################################
#       Link                                   #
################################################
ln -s /dev/rdsk/c3b0t0d2s1    stok25
ln -s /dev/rdsk/c3b0t0d2s2    cust25
ln -s /dev/rdsk/c3b0t0d2s3    ordl25
ln -s /dev/rdsk/c3b0t0d2s5    ordr13
ln -s /dev/rdsk/c3b0t0d2s9    icust2_25
ln -s /dev/rdsk/c3b0t0d2sa    iord1_25
ln -s /dev/rdsk/c3b0t0d2sb    iord2_25
ln -s /dev/rdsk/c3b0t0d2sc    unused25
ln -s /dev/rdsk/c3b0t0d2se    temp09
#
ln -s /dev/rdsk/c3b0t0d3s1    stok26
ln -s /dev/rdsk/c3b0t0d3s2    cust26
ln -s /dev/rdsk/c3b0t0d3s3    ordl26
ln -s /dev/rdsk/c3b0t0d3s5    ordr14
ln -s /dev/rdsk/c3b0t0d3s9    icust2_26
```

```
ln -s /dev/rdsk/c3b0t0d3sa    iord1_26
ln -s /dev/rdsk/c3b0t0d3sb    iord2_26
ln -s /dev/rdsk/c3b0t0d3sc    unused26
ln -s /dev/rdsk/c3b0t0d3se    temp10
#
ln -s /dev/rdsk/c3b0t0d4s1    stok27
ln -s /dev/rdsk/c3b0t0d4s2    cust27
ln -s /dev/rdsk/c3b0t0d4s3    ordl27
ln -s /dev/rdsk/c3b0t0d4s4    misc13
ln -s /dev/rdsk/c3b0t0d4s5    ordr15
ln -s /dev/rdsk/c3b0t0d4s9    icust2_27
ln -s /dev/rdsk/c3b0t0d4sa    iord1_27
ln -s /dev/rdsk/c3b0t0d4sb    iord2_27
ln -s /dev/rdsk/c3b0t0d4sc    unused27
ln -s /dev/rdsk/c3b0t0d4sd    hist09
#
ln -s /dev/rdsk/c3b0t0d5s1    stok28
ln -s /dev/rdsk/c3b0t0d5s2    cust28
ln -s /dev/rdsk/c3b0t0d5s3    ordl28
ln -s /dev/rdsk/c3b0t0d5s6    nord13
ln -s /dev/rdsk/c3b0t0d5s9    icust2_28
ln -s /dev/rdsk/c3b0t0d5sa    iord1_28
ln -s /dev/rdsk/c3b0t0d5sb    iord2_28
ln -s /dev/rdsk/c3b0t0d5sc    unused28
ln -s /dev/rdsk/c3b0t0d5sd    hist10
#
ln -s /dev/rdsk/c3b0t0d6s1    stok29
ln -s /dev/rdsk/c3b0t0d6s2    cust29
ln -s /dev/rdsk/c3b0t0d6s3    ordl29
ln -s /dev/rdsk/c3b0t0d6s4    misc14
ln -s /dev/rdsk/c3b0t0d6s6    nord14
ln -s /dev/rdsk/c3b0t0d6s9    icust2_29
ln -s /dev/rdsk/c3b0t0d6sa    iord1_29
ln -s /dev/rdsk/c3b0t0d6sb    iord2_29
ln -s /dev/rdsk/c3b0t0d6sc    unused29
#
ln -s /dev/rdsk/c3b0t0d7s1    stok30
ln -s /dev/rdsk/c3b0t0d7s2    cust30
ln -s /dev/rdsk/c3b0t0d7s3    ordl30
ln -s /dev/rdsk/c3b0t0d7s4    misc15
ln -s /dev/rdsk/c3b0t0d7s6    nord15
ln -s /dev/rdsk/c3b0t0d7s9    icust2_30
ln -s /dev/rdsk/c3b0t0d7sa    iord1_30
ln -s /dev/rdsk/c3b0t0d7sb    iord2_30
ln -s /dev/rdsk/c3b0t0d7sc    unused30
#
ln -s /dev/rdsk/c5b0t0d2s1    stok31
ln -s /dev/rdsk/c5b0t0d2s2    cust31
ln -s /dev/rdsk/c5b0t0d2s3    ordl31
ln -s /dev/rdsk/c5b0t0d2s5    ordr16
ln -s /dev/rdsk/c5b0t0d2s9    icust2_31
ln -s /dev/rdsk/c5b0t0d2sa    iord1_31
ln -s /dev/rdsk/c5b0t0d2sb    iord2_31
ln -s /dev/rdsk/c5b0t0d2sc    unused31
ln -s /dev/rdsk/c5b0t0d2se    temp11
#
ln -s /dev/rdsk/c3b0t1d9s1    stok32
ln -s /dev/rdsk/c3b0t1d9s2    cust32
ln -s /dev/rdsk/c3b0t1d9s3    ordl32
ln -s /dev/rdsk/c3b0t1d9s5    ordr17
ln -s /dev/rdsk/c3b0t1d9s9    icust2_32
ln -s /dev/rdsk/c3b0t1d9sa    iord1_32
ln -s /dev/rdsk/c3b0t1d9sb    iord2_32
ln -s /dev/rdsk/c3b0t1d9sc    unused32
ln -s /dev/rdsk/c3b0t1d9se    temp12
#
ln -s /dev/rdsk/c3b0t1d10s1   stok33
ln -s /dev/rdsk/c3b0t1d10s2   cust33
ln -s /dev/rdsk/c3b0t1d10s3   ordl33
ln -s /dev/rdsk/c3b0t1d10s4   misc16
ln -s /dev/rdsk/c3b0t1d10s5   ordr18
ln -s /dev/rdsk/c3b0t1d10s9   icust2_33
ln -s /dev/rdsk/c3b0t1d10sa   iord1_33
ln -s /dev/rdsk/c3b0t1d10sb   iord2_33
ln -s /dev/rdsk/c3b0t1d10sc   unused33
ln -s /dev/rdsk/c3b0t1d10sd   hist11
ln -s /dev/rdsk/c3b0t1d10se   sys1
#
ln -s /dev/rdsk/c3b0t1d11s1   stok34
ln -s /dev/rdsk/c3b0t1d11s2   cust34
ln -s /dev/rdsk/c3b0t1d11s3   ordl34
ln -s /dev/rdsk/c3b0t1d11s6   nord16
ln -s /dev/rdsk/c3b0t1d11s9   icust2_34
ln -s /dev/rdsk/c3b0t1d11sa   iord1_34
ln -s /dev/rdsk/c3b0t1d11sb   iord2_34
ln -s /dev/rdsk/c3b0t1d11sc   unused34
ln -s /dev/rdsk/c3b0t1d11sd   hist12
#
ln -s /dev/rdsk/c3b0t1d12s1   stok35
ln -s /dev/rdsk/c3b0t1d12s2   cust35
ln -s /dev/rdsk/c3b0t1d12s3   ordl35
ln -s /dev/rdsk/c3b0t1d12s4   misc17
ln -s /dev/rdsk/c3b0t1d12s6   nord17
ln -s /dev/rdsk/c3b0t1d12s9   icust2_35
ln -s /dev/rdsk/c3b0t1d12sa   iord1_35
ln -s /dev/rdsk/c3b0t1d12sb   iord2_35
ln -s /dev/rdsk/c3b0t1d12sc   unused35
#
ln -s /dev/rdsk/c3b0t1d13s1   stok36
ln -s /dev/rdsk/c3b0t1d13s2   cust36
ln -s /dev/rdsk/c3b0t1d13s3   ordl36
ln -s /dev/rdsk/c3b0t1d13s4   misc18
ln -s /dev/rdsk/c3b0t1d13s6   nord18
ln -s /dev/rdsk/c3b0t1d13s9   icust2_36
ln -s /dev/rdsk/c3b0t1d13sa   iord1_36
ln -s /dev/rdsk/c3b0t1d13sb   iord2_36
ln -s /dev/rdsk/c3b0t1d13sc   unused36
```

c4_data.ln

```
#####################################################
#         Link                                      #
#####################################################
ln -s /dev/rdsk/c4b0t0d2s1    stok37
```

```
ln -s /dev/rdsk/c4b0t0d2s2    cust37
ln -s /dev/rdsk/c4b0t0d2s3    ordl37
ln -s /dev/rdsk/c4b0t0d2s5    ordr19
ln -s /dev/rdsk/c4b0t0d2s9    icust2_37
ln -s /dev/rdsk/c4b0t0d2sa    iord1_37
ln -s /dev/rdsk/c4b0t0d2sb    iord2_37
ln -s /dev/rdsk/c4b0t0d2sc    unused37
ln -s /dev/rdsk/c4b0t0d2se    temp13
#
ln -s /dev/rdsk/c4b0t0d3s1    stok38
ln -s /dev/rdsk/c4b0t0d3s2    cust38
ln -s /dev/rdsk/c4b0t0d3s3    ordl38
ln -s /dev/rdsk/c4b0t0d3s5    ordr20
ln -s /dev/rdsk/c4b0t0d3s9    icust2_38
ln -s /dev/rdsk/c4b0t0d3sa    iord1_38
ln -s /dev/rdsk/c4b0t0d3sb    iord2_38
ln -s /dev/rdsk/c4b0t0d3sc    unused38
ln -s /dev/rdsk/c4b0t0d3se    temp14
#
ln -s /dev/rdsk/c4b0t0d4s1    stok39
ln -s /dev/rdsk/c4b0t0d4s2    cust39
ln -s /dev/rdsk/c4b0t0d4s3    ordl39
ln -s /dev/rdsk/c4b0t0d4s4    misc19
ln -s /dev/rdsk/c4b0t0d4s5    ordr21
ln -s /dev/rdsk/c4b0t0d4s6    stat01
ln -s /dev/rdsk/c4b0t0d4s9    icust2_39
ln -s /dev/rdsk/c4b0t0d4sa    iord1_39
ln -s /dev/rdsk/c4b0t0d4sb    iord2_39
ln -s /dev/rdsk/c4b0t0d4sc    unused39
ln -s /dev/rdsk/c4b0t0d4sd    hist13
#
ln -s /dev/rdsk/c4b0t0d5s1    stok40
ln -s /dev/rdsk/c4b0t0d5s2    cust40
ln -s /dev/rdsk/c4b0t0d5s3    ordl40
ln -s /dev/rdsk/c4b0t0d5s6    nord19
ln -s /dev/rdsk/c4b0t0d5s9    icust2_40
ln -s /dev/rdsk/c4b0t0d5sa    iord1_40
ln -s /dev/rdsk/c4b0t0d5sb    iord2_40
ln -s /dev/rdsk/c4b0t0d5sc    unused40
ln -s /dev/rdsk/c4b0t0d5sd    hist14
#
ln -s /dev/rdsk/c4b0t0d6s1    stok41
ln -s /dev/rdsk/c4b0t0d6s2    cust41
ln -s /dev/rdsk/c4b0t0d6s3    ordl41
ln -s /dev/rdsk/c4b0t0d6s4    misc20
ln -s /dev/rdsk/c4b0t0d6s6    nord20
ln -s /dev/rdsk/c4b0t0d6s9    icust2_41
ln -s /dev/rdsk/c4b0t0d6sa    iord1_41
ln -s /dev/rdsk/c4b0t0d6sb    iord2_41
ln -s /dev/rdsk/c4b0t0d6sc    unused41
#
ln -s /dev/rdsk/c4b0t0d7s1    stok42
ln -s /dev/rdsk/c4b0t0d7s2    cust42
ln -s /dev/rdsk/c4b0t0d7s3    ordl42
ln -s /dev/rdsk/c4b0t0d7s4    misc21
ln -s /dev/rdsk/c4b0t0d7s6    nord21
ln -s /dev/rdsk/c4b0t0d7s9    icust2_42
ln -s /dev/rdsk/c4b0t0d7sa    iord1_42
ln -s /dev/rdsk/c4b0t0d7sb    iord2_42
ln -s /dev/rdsk/c4b0t0d7sc    unused42
#
ln -s /dev/rdsk/c4b0t1d8s1    stok43
ln -s /dev/rdsk/c4b0t1d8s2    cust43
ln -s /dev/rdsk/c4b0t1d8s3    ordl43
ln -s /dev/rdsk/c4b0t1d8s5    ordr22
ln -s /dev/rdsk/c4b0t1d8s9    icust2_43
ln -s /dev/rdsk/c4b0t1d8sa    iord1_43
ln -s /dev/rdsk/c4b0t1d8sb    iord2_43
ln -s /dev/rdsk/c4b0t1d8sc    unused43
ln -s /dev/rdsk/c4b0t1d8se    temp15

ln -s /dev/rdsk/c4b0t1d9s1    stok44
ln -s /dev/rdsk/c4b0t1d9s2    cust44
ln -s /dev/rdsk/c4b0t1d9s3    ordl44
ln -s /dev/rdsk/c4b0t1d9s5    ordr23
ln -s /dev/rdsk/c4b0t1d9s9    icust2_44
ln -s /dev/rdsk/c4b0t1d9sa    iord1_44
ln -s /dev/rdsk/c4b0t1d9sb    iord2_44
ln -s /dev/rdsk/c4b0t1d9sc    unused44
ln -s /dev/rdsk/c4b0t1d9se    temp16
#
ln -s /dev/rdsk/c4b0t1d10s1   stok45
ln -s /dev/rdsk/c4b0t1d10s2   cust45
ln -s /dev/rdsk/c4b0t1d10s3   ordl45
ln -s /dev/rdsk/c4b0t1d10s4   misc22
ln -s /dev/rdsk/c4b0t1d10s5   ordr24
ln -s /dev/rdsk/c4b0t1d10s9   icust2_45
ln -s /dev/rdsk/c4b0t1d10sa   iord1_45
ln -s /dev/rdsk/c4b0t1d10sb   iord2_45
ln -s /dev/rdsk/c4b0t1d10sc   unused45
ln -s /dev/rdsk/c4b0t1d10sd   hist15
ln -s /dev/rdsk/c4b0t1d10se   sys2
#
ln -s /dev/rdsk/c4b0t1d11s1   stok46
ln -s /dev/rdsk/c4b0t1d11s2   cust46
ln -s /dev/rdsk/c4b0t1d11s3   ordl46
ln -s /dev/rdsk/c4b0t1d11s6   nord22
ln -s /dev/rdsk/c4b0t1d11s9   icust2_46
ln -s /dev/rdsk/c4b0t1d11sa   iord1_46
ln -s /dev/rdsk/c4b0t1d11sb   iord2_46
ln -s /dev/rdsk/c4b0t1d11sc   unused46
ln -s /dev/rdsk/c4b0t1d11sd   hist16
#
ln -s /dev/rdsk/c4b0t1d12s1   stok47
ln -s /dev/rdsk/c4b0t1d12s2   cust47
ln -s /dev/rdsk/c4b0t1d12s3   ordl47
ln -s /dev/rdsk/c4b0t1d12s4   misc23
ln -s /dev/rdsk/c4b0t1d12s6   nord23
ln -s /dev/rdsk/c4b0t1d12s9   icust2_47
ln -s /dev/rdsk/c4b0t1d12sa   iord1_47
ln -s /dev/rdsk/c4b0t1d12sb   iord2_47
ln -s /dev/rdsk/c4b0t1d12sc   unused47
#
ln -s /dev/rdsk/c4b0t1d13s1   stok48
ln -s /dev/rdsk/c4b0t1d13s2   cust48
ln -s /dev/rdsk/c4b0t1d13s3   ordl48
ln -s /dev/rdsk/c4b0t1d13s4   misc24
ln -s /dev/rdsk/c4b0t1d13s6   nord24
```

```
ln -s /dev/rdsk/c4b0t1d13s9   icust2_48
ln -s /dev/rdsk/c4b0t1d13sa   iord1_48
ln -s /dev/rdsk/c4b0t1d13sb   iord2_48
ln -s /dev/rdsk/c4b0t1d13sc   unused48
```

log.ln

```
ln -s /dev/rdsk/c6b0t0d0s1 log1
ln -s /dev/rdsk/c6b0t0d0s2 log2
```

Backup Scripts

take_backup.sh

```
ODIR=/oracle/8i/dbs/tpcc_backup/outdir


backup_iord1.sh > ${ODIR}/backup_iord1.out 2>&1 &
wait
backup_ordl.sh > ${ODIR}/backup_ordl.out 2>&1 &
wait
backup_temp.sh > ${ODIR}/backup_temp.out 2>&1 &
wait
backup_iord2.sh > ${ODIR}/backup_iord2.out 2>&1 &
wait
backup_ordr.sh > ${ODIR}/backup_ordr.out 2>&1 &
wait
backup_unused.sh > ${ODIR}/backup_unused.out 2>&1 &
wait
backup_cust.sh > ${ODIR}/backup_cust.out 2>&1 &
wait
backup_log.sh > ${ODIR}/backup_log.out 2>&1 &
wait
backup_roll.sh > ${ODIR}/backup_roll.out 2>&1 &
wait
backup_hist.sh > ${ODIR}/backup_hist.out 2>&1 &
wait
backup_misc.sh > ${ODIR}/backup_misc.out 2>&1 &
wait
backup_stat.sh > ${ODIR}/backup_stat.out 2>&1 &
wait
backup_icust2.sh > ${ODIR}/backup_icust2.out 2>&1 &
wait
backup_nord.sh > ${ODIR}/backup_nord.out 2>&1 &
wait
backup_stok.sh > ${ODIR}/backup_stok.out 2>&1 &
```

create_link.sh

```
### Script for creating backup links

./backup1.ln
./backup2.ln
./backup3.ln
./backup4.ln
./backup5_new.ln
./backup6_new.ln
./backup7.ln
./backup8.ln
```

backup1.ln

```
################################################################
# Backup of Last 4 Drives of Cabinet 4 & Log                   #
################################################################
ln -s /dev/rdsk/c1b0t0d0s1    stok45
ln -s /dev/rdsk/c1b0t0d0s2    cust45
ln -s /dev/rdsk/c1b0t0d0s3    ordl45
ln -s /dev/rdsk/c1b0t0d0s4    misc22
ln -s /dev/rdsk/c1b0t0d0s5    ordr24
ln -s /dev/rdsk/c1b0t0d0s9    icust2_45
ln -s /dev/rdsk/c1b0t0d0sa    iord1_45
ln -s /dev/rdsk/c1b0t0d0sb    iord2_45
ln -s /dev/rdsk/c1b0t0d0sc    unused45
ln -s /dev/rdsk/c1b0t0d0sd    hist15
ln -s /dev/rdsk/c1b0t0d0se    sys2

ln -s /dev/rdsk/c1b0t0d0sf    stok46
ln -s /dev/rdsk/c1b0t0d0s10   cust46
ln -s /dev/rdsk/c1b0t0d0s11   ordl46
ln -s /dev/rdsk/c1b0t0d0s14   nord22
ln -s /dev/rdsk/c1b0t0d0s15   icust2_46
ln -s /dev/rdsk/c1b0t0d0s16   iord1_46
ln -s /dev/rdsk/c1b0t0d0s17   iord2_46
ln -s /dev/rdsk/c1b0t0d0s18   unused46
ln -s /dev/rdsk/c1b0t0d0s19   hist16

ln -s /dev/rdsk/c1b0t0d0s1b   stok47
ln -s /dev/rdsk/c1b0t0d0s1c   cust47
ln -s /dev/rdsk/c1b0t0d0s1d   ordl47
ln -s /dev/rdsk/c1b0t0d0s1e   misc23
ln -s /dev/rdsk/c1b0t0d0s20   nord23
ln -s /dev/rdsk/c1b0t0d0s21   icust2_47
ln -s /dev/rdsk/c1b0t0d0s22   iord1_47
ln -s /dev/rdsk/c1b0t0d0s23   iord2_47
ln -s /dev/rdsk/c1b0t0d0s24   unused47

ln -s /dev/rdsk/c1b0t0d0s27   stok48
ln -s /dev/rdsk/c1b0t0d0s28   cust48
ln -s /dev/rdsk/c1b0t0d0s29   ordl48
ln -s /dev/rdsk/c1b0t0d0s2a   misc24
```

```
ln -s /dev/rdsk/c1b0t0d0s2c   nord24
ln -s /dev/rdsk/c1b0t0d0s2d   icust2_48
ln -s /dev/rdsk/c1b0t0d0s2e   iord1_48
ln -s /dev/rdsk/c1b0t0d0s2f   iord2_48
ln -s /dev/rdsk/c1b0t0d0s30   unused48

####### Backup of Log ##########

ln -s /dev/rdsk/c1b0t0d0s33   log1
ln -s /dev/rdsk/c1b0t0d0s34   log2
ln -s /dev/rdsk/c1b0t0d0s35   log3
ln -s /dev/rdsk/c1b0t0d0s36   log4
```

backup8.ln

```
#############################################################
#   Backup of Last 4 Drives of Cabinet 3 and Cabinet 5      #
#############################################################
ln -s /dev/rdsk/c4b0t1d1s1    stok33
ln -s /dev/rdsk/c4b0t1d1s2    cust33
ln -s /dev/rdsk/c4b0t1d1s3    ordl33
ln -s /dev/rdsk/c4b0t1d1s4    misc16
ln -s /dev/rdsk/c4b0t1d1s5    ordr18
ln -s /dev/rdsk/c4b0t1d1s9    icust2_33
ln -s /dev/rdsk/c4b0t1d1sa    iord1_33
ln -s /dev/rdsk/c4b0t1d1sb    iord2_33
ln -s /dev/rdsk/c4b0t1d1sc    unused33
ln -s /dev/rdsk/c4b0t1d1sd    hist11
ln -s /dev/rdsk/c4b0t1d1se    sys1

ln -s /dev/rdsk/c4b0t1d1sf    stok34
ln -s /dev/rdsk/c4b0t1d1s10   cust34
ln -s /dev/rdsk/c4b0t1d1s11   ordl34
ln -s /dev/rdsk/c4b0t1d1s14   nord16
ln -s /dev/rdsk/c4b0t1d1s15   icust2_34
ln -s /dev/rdsk/c4b0t1d1s16   iord1_34
ln -s /dev/rdsk/c4b0t1d1s17   iord2_34
ln -s /dev/rdsk/c4b0t1d1s18   unused34
ln -s /dev/rdsk/c4b0t1d1s19   hist12

ln -s /dev/rdsk/c4b0t1d1s1b   stok35
ln -s /dev/rdsk/c4b0t1d1s1c   cust35
ln -s /dev/rdsk/c4b0t1d1s1d   ordl35
ln -s /dev/rdsk/c4b0t1d1s1e   misc17
ln -s /dev/rdsk/c4b0t1d1s20   nord17
ln -s /dev/rdsk/c4b0t1d1s21   icust2_35
ln -s /dev/rdsk/c4b0t1d1s22   iord1_35
ln -s /dev/rdsk/c4b0t1d1s23   iord2_35
ln -s /dev/rdsk/c4b0t1d1s24   unused35

ln -s /dev/rdsk/c4b0t1d1s27   stok36
ln -s /dev/rdsk/c4b0t1d1s28   cust36
ln -s /dev/rdsk/c4b0t1d1s29   ordl36
ln -s /dev/rdsk/c4b0t1d1s2a   misc18
ln -s /dev/rdsk/c4b0t1d1s2c   nord18
ln -s /dev/rdsk/c4b0t1d1s2d   icust2_36
ln -s /dev/rdsk/c4b0t1d1s2e   iord1_36
ln -s /dev/rdsk/c4b0t1d1s2f   iord2_36
ln -s /dev/rdsk/c4b0t1d1s30   unused36

#### C5.data

ln -s /dev/rdsk/c4b0t1d1s33   stok57
ln -s /dev/rdsk/c4b0t1d1s34   cust57
ln -s /dev/rdsk/c4b0t1d1s35   ordl57
ln -s /dev/rdsk/c4b0t1d1s36   misc28
ln -s /dev/rdsk/c4b0t1d1s37   ordr30
ln -s /dev/rdsk/c4b0t1d1s39   icust2_57
ln -s /dev/rdsk/c4b0t1d1s3a   iord1_57
ln -s /dev/rdsk/c4b0t1d1s3b   iord2_57
ln -s /dev/rdsk/c4b0t1d1s3c   unused57
ln -s /dev/rdsk/c4b0t1d1s3d   hist19

ln -s /dev/rdsk/c4b0t1d1s3f   stok58
ln -s /dev/rdsk/c4b0t1d1s40   cust58
ln -s /dev/rdsk/c4b0t1d1s41   ordl58
ln -s /dev/rdsk/c4b0t1d1s44   nord28
ln -s /dev/rdsk/c4b0t1d1s45   icust2_58
ln -s /dev/rdsk/c4b0t1d1s46   iord1_58
ln -s /dev/rdsk/c4b0t1d1s47   iord2_58
ln -s /dev/rdsk/c4b0t1d1s48   unused58
ln -s /dev/rdsk/c4b0t1d1s49   hist20

ln -s /dev/rdsk/c4b0t1d1s4b   stok59
ln -s /dev/rdsk/c4b0t1d1s4c   cust59
ln -s /dev/rdsk/c4b0t1d1s4d   ordl59
ln -s /dev/rdsk/c4b0t1d1s4e   misc29
ln -s /dev/rdsk/c4b0t1d1s50   nord29
ln -s /dev/rdsk/c4b0t1d1s51   icust2_59
ln -s /dev/rdsk/c4b0t1d1s52   iord1_59
ln -s /dev/rdsk/c4b0t1d1s53   iord2_59
ln -s /dev/rdsk/c4b0t1d1s54   unused59

ln -s /dev/rdsk/c4b0t1d1s57   stok60
#ln -s /dev/rdsk/c4b0t1d1s58  cust60
ln -s /dev/rdsk/c4b0t1d1s59   cust60
ln -s /dev/rdsk/c4b0t1d1s5a   ordl60
ln -s /dev/rdsk/c4b0t1d1s5b   misc30
ln -s /dev/rdsk/c4b0t1d1s5d   nord30
ln -s /dev/rdsk/c4b0t1d1s5e   icust2_60
ln -s /dev/rdsk/c4b0t1d1s5f   iord1_60
ln -s /dev/rdsk/c4b0t1d1s60   iord2_60
ln -s /dev/rdsk/c4b0t1d1s61   unused60
```

backup7.ln

```
#############################################################
#   Backup of Cabinet 3 for first 8 data drives             #
#############################################################
```

```
ln -s /dev/rdsk/c4b0t0d0s1    stok25
ln -s /dev/rdsk/c4b0t0d0s2    cust25
ln -s /dev/rdsk/c4b0t0d0s3    ordl25
ln -s /dev/rdsk/c4b0t0d0s5    ordr13
ln -s /dev/rdsk/c4b0t0d0s9    icust2_25
ln -s /dev/rdsk/c4b0t0d0sa    iord1_25
ln -s /dev/rdsk/c4b0t0d0sb    iord2_25
ln -s /dev/rdsk/c4b0t0d0sc    unused25
ln -s /dev/rdsk/c4b0t0d0se    temp09

ln -s /dev/rdsk/c4b0t0d0sf    stok26
ln -s /dev/rdsk/c4b0t0d0s10   cust26
ln -s /dev/rdsk/c4b0t0d0s11   ordl26
ln -s /dev/rdsk/c4b0t0d0s13   ordr14
ln -s /dev/rdsk/c4b0t0d0s15   icust2_26
ln -s /dev/rdsk/c4b0t0d0s16   iord1_26
ln -s /dev/rdsk/c4b0t0d0s17   iord2_26
ln -s /dev/rdsk/c4b0t0d0s18   unused26
ln -s /dev/rdsk/c4b0t0d0s1a   temp10

ln -s /dev/rdsk/c4b0t0d0s1b   stok27
ln -s /dev/rdsk/c4b0t0d0s1c   cust27
ln -s /dev/rdsk/c4b0t0d0s1d   ordl27
ln -s /dev/rdsk/c4b0t0d0s1e   misc13
ln -s /dev/rdsk/c4b0t0d0s1f   ordr15
ln -s /dev/rdsk/c4b0t0d0s21   icust2_27
ln -s /dev/rdsk/c4b0t0d0s22   iord1_27
ln -s /dev/rdsk/c4b0t0d0s23   iord2_27
ln -s /dev/rdsk/c4b0t0d0s24   unused27
ln -s /dev/rdsk/c4b0t0d0s25   hist09

ln -s /dev/rdsk/c4b0t0d0s27   stok28
ln -s /dev/rdsk/c4b0t0d0s28   cust28
ln -s /dev/rdsk/c4b0t0d0s29   ordl28
ln -s /dev/rdsk/c4b0t0d0s2c   nord13
ln -s /dev/rdsk/c4b0t0d0s2d   icust2_28
ln -s /dev/rdsk/c4b0t0d0s2e   iord1_28
ln -s /dev/rdsk/c4b0t0d0s2f   iord2_28
ln -s /dev/rdsk/c4b0t0d0s30   unused28
ln -s /dev/rdsk/c4b0t0d0s31   hist10

ln -s /dev/rdsk/c4b0t0d0s33   stok29
ln -s /dev/rdsk/c4b0t0d0s34   cust29
ln -s /dev/rdsk/c4b0t0d0s35   ordl29
ln -s /dev/rdsk/c4b0t0d0s36   misc14
ln -s /dev/rdsk/c4b0t0d0s38   nord14
ln -s /dev/rdsk/c4b0t0d0s39   icust2_29
ln -s /dev/rdsk/c4b0t0d0s3a   iord1_29
ln -s /dev/rdsk/c4b0t0d0s3b   iord2_29
ln -s /dev/rdsk/c4b0t0d0s3c   unused29

ln -s /dev/rdsk/c4b0t0d0s3f   stok30
ln -s /dev/rdsk/c4b0t0d0s40   cust30
ln -s /dev/rdsk/c4b0t0d0s41   ordl30
ln -s /dev/rdsk/c4b0t0d0s42   misc15
ln -s /dev/rdsk/c4b0t0d0s44   nord15
ln -s /dev/rdsk/c4b0t0d0s45   icust2_30
ln -s /dev/rdsk/c4b0t0d0s46   iord1_30
ln -s /dev/rdsk/c4b0t0d0s47   iord2_30
ln -s /dev/rdsk/c4b0t0d0s48   unused30

ln -s /dev/rdsk/c4b0t0d0s4b   stok31
ln -s /dev/rdsk/c4b0t0d0s4c   cust31
ln -s /dev/rdsk/c4b0t0d0s4d   ordl31
ln -s /dev/rdsk/c4b0t0d0s4f   ordr16
ln -s /dev/rdsk/c4b0t0d0s51   icust2_31
ln -s /dev/rdsk/c4b0t0d0s52   iord1_31
ln -s /dev/rdsk/c4b0t0d0s53   iord2_31
ln -s /dev/rdsk/c4b0t0d0s54   unused31
ln -s /dev/rdsk/c4b0t0d0s56   temp11

ln -s /dev/rdsk/c4b0t0d0s57   stok32
#ln -s /dev/rdsk/c4b0t0d0s58  cust32
ln -s /dev/rdsk/c4b0t0d0s59   cust32
ln -s /dev/rdsk/c4b0t0d0s5a   ordl32
ln -s /dev/rdsk/c4b0t0d0s5c   ordr17
ln -s /dev/rdsk/c4b0t0d0s5e   icust2_32
ln -s /dev/rdsk/c4b0t0d0s5f   iord1_32
ln -s /dev/rdsk/c4b0t0d0s60   iord2_32
ln -s /dev/rdsk/c4b0t0d0s61   unused32
ln -s /dev/rdsk/c4b0t0d0s63   temp12
```

backup6.ln

```
#############################################################
#  Backup Last 4 Drives of Cabinet 1 & Cabinet 2            #
#############################################################
ln -s /dev/rdsk/c5b0t1d1s1    stok09
ln -s /dev/rdsk/c5b0t1d1s2    cust09
ln -s /dev/rdsk/c5b0t1d1s3    ordl09
ln -s /dev/rdsk/c5b0t1d1s4    misc04
ln -s /dev/rdsk/c5b0t1d1s5    ordr06
ln -s /dev/rdsk/c5b0t1d1s9    icust2_09
ln -s /dev/rdsk/c5b0t1d1sa    iord1_09
ln -s /dev/rdsk/c5b0t1d1sb    iord2_09
ln -s /dev/rdsk/c5b0t1d1sc    unused09
ln -s /dev/rdsk/c5b0t1d1sd    hist03

ln -s /dev/rdsk/c5b0t1d1sf    stok10
ln -s /dev/rdsk/c5b0t1d1s10   cust10
ln -s /dev/rdsk/c5b0t1d1s11   ordl10
ln -s /dev/rdsk/c5b0t1d1s14   nord04
ln -s /dev/rdsk/c5b0t1d1s15   icust2_10
ln -s /dev/rdsk/c5b0t1d1s16   iord1_10
ln -s /dev/rdsk/c5b0t1d1s17   iord2_10
ln -s /dev/rdsk/c5b0t1d1s18   unused10
ln -s /dev/rdsk/c5b0t1d1s19   hist04
ln -s /dev/rdsk/c5b0t1d1s1b   stok11
ln -s /dev/rdsk/c5b0t1d1s1c   cust11
ln -s /dev/rdsk/c5b0t1d1s1d   ordl11
ln -s /dev/rdsk/c5b0t1d1s1e   misc05
ln -s /dev/rdsk/c5b0t1d1s20   nord05
ln -s /dev/rdsk/c5b0t1d1s21   icust2_11
```

```
ln -s /dev/rdsk/c5b0t1d1s22   iord1_11
ln -s /dev/rdsk/c5b0t1d1s23   iord2_11
ln -s /dev/rdsk/c5b0t1d1s24   unused11

ln -s /dev/rdsk/c5b0t1d1s27   stok12
ln -s /dev/rdsk/c5b0t1d1s28   cust12
ln -s /dev/rdsk/c5b0t1d1s29   ordl12
ln -s /dev/rdsk/c5b0t1d1s2a   misc06
ln -s /dev/rdsk/c5b0t1d1s2c   nord06
ln -s /dev/rdsk/c5b0t1d1s2d   icust2_12
ln -s /dev/rdsk/c5b0t1d1s2e   iord1_12
ln -s /dev/rdsk/c5b0t1d1s2f   iord2_12
ln -s /dev/rdsk/c5b0t1d1s30   unused12

##### c2_data

ln -s /dev/rdsk/c5b0t1d1s33   stok21
ln -s /dev/rdsk/c5b0t1d1s34   cust21
ln -s /dev/rdsk/c5b0t1d1s35   ordl21
ln -s /dev/rdsk/c5b0t1d1s36   misc10
ln -s /dev/rdsk/c5b0t1d1s37   ordr12
ln -s /dev/rdsk/c5b0t1d1s39   icust2_21
ln -s /dev/rdsk/c5b0t1d1s3a   iord1_21
ln -s /dev/rdsk/c5b0t1d1s3b   iord2_21
ln -s /dev/rdsk/c5b0t1d1s3c   unused21
ln -s /dev/rdsk/c5b0t1d1s3d   hist07

ln -s /dev/rdsk/c5b0t1d1s3f   stok22
ln -s /dev/rdsk/c5b0t1d1s40   cust22
ln -s /dev/rdsk/c5b0t1d1s41   ordl22
ln -s /dev/rdsk/c5b0t1d1s44   nord10
ln -s /dev/rdsk/c5b0t1d1s45   icust2_22
ln -s /dev/rdsk/c5b0t1d1s46   iord1_22
ln -s /dev/rdsk/c5b0t1d1s47   iord2_22
ln -s /dev/rdsk/c5b0t1d1s48   unused22
ln -s /dev/rdsk/c5b0t1d1s49   hist08

ln -s /dev/rdsk/c5b0t1d1s4b   stok23
ln -s /dev/rdsk/c5b0t1d1s4c   cust23
ln -s /dev/rdsk/c5b0t1d1s4d   ordl23
ln -s /dev/rdsk/c5b0t1d1s4e   misc11
ln -s /dev/rdsk/c5b0t1d1s50   nord11
ln -s /dev/rdsk/c5b0t1d1s51   icust2_23
ln -s /dev/rdsk/c5b0t1d1s52   iord1_23
ln -s /dev/rdsk/c5b0t1d1s53   iord2_23
ln -s /dev/rdsk/c5b0t1d1s54   unused23

ln -s /dev/rdsk/c5b0t1d1s57   stok24
#ln -s /dev/rdsk/c5b0t1d1s58  cust24
ln -s /dev/rdsk/c5b0t1d1s59   cust24
ln -s /dev/rdsk/c5b0t1d1s5a   ordl24
ln -s /dev/rdsk/c5b0t1d1s5b   misc12
ln -s /dev/rdsk/c5b0t1d1s5d   nord12
ln -s /dev/rdsk/c5b0t1d1s5e   icust2_24
ln -s /dev/rdsk/c5b0t1d1s5f   iord1_24
ln -s /dev/rdsk/c5b0t1d1s60   iord2_24
ln -s /dev/rdsk/c5b0t1d1s61   unused24
```

backup5.ln

```
############################################################
#    Backup of First 8 Drives of Cabinet 2                 #
############################################################
ln -s /dev/rdsk/c5b0t0d0s1    stok13
ln -s /dev/rdsk/c5b0t0d0s2    cust13
ln -s /dev/rdsk/c5b0t0d0s3    ordl13
ln -s /dev/rdsk/c5b0t0d0s5    ordr07
ln -s /dev/rdsk/c5b0t0d0s9    icust2_13
ln -s /dev/rdsk/c5b0t0d0sa    iord1_13
ln -s /dev/rdsk/c5b0t0d0sb    iord2_13
ln -s /dev/rdsk/c5b0t0d0sc    unused13
ln -s /dev/rdsk/c5b0t0d0se    temp05

ln -s /dev/rdsk/c5b0t0d0sf    stok14
ln -s /dev/rdsk/c5b0t0d0s10   cust14
ln -s /dev/rdsk/c5b0t0d0s11   ordl14
ln -s /dev/rdsk/c5b0t0d0s13   ordr08
ln -s /dev/rdsk/c5b0t0d0s15   icust2_14
ln -s /dev/rdsk/c5b0t0d0s16   iord1_14
ln -s /dev/rdsk/c5b0t0d0s17   iord2_14
ln -s /dev/rdsk/c5b0t0d0s18   unused14
ln -s /dev/rdsk/c5b0t0d0s1a   temp06

ln -s /dev/rdsk/c5b0t0d0s1b   stok15
ln -s /dev/rdsk/c5b0t0d0s1c   cust15
ln -s /dev/rdsk/c5b0t0d0s1d   ordl15
ln -s /dev/rdsk/c5b0t0d0s1e   misc07
ln -s /dev/rdsk/c5b0t0d0s1f   ordr09
ln -s /dev/rdsk/c5b0t0d0s21   icust2_15
ln -s /dev/rdsk/c5b0t0d0s22   iord1_15
ln -s /dev/rdsk/c5b0t0d0s23   iord2_15
ln -s /dev/rdsk/c5b0t0d0s24   unused15
ln -s /dev/rdsk/c5b0t0d0s25   hist05

ln -s /dev/rdsk/c5b0t0d0s27   stok16
ln -s /dev/rdsk/c5b0t0d0s28   cust16
ln -s /dev/rdsk/c5b0t0d0s29   ordl16
ln -s /dev/rdsk/c5b0t0d0s2c   nord07
ln -s /dev/rdsk/c5b0t0d0s2d   icust2_16
ln -s /dev/rdsk/c5b0t0d0s2e   iord1_16
ln -s /dev/rdsk/c5b0t0d0s2f   iord2_16
ln -s /dev/rdsk/c5b0t0d0s30   unused16
ln -s /dev/rdsk/c5b0t0d0s31   hist06

ln -s /dev/rdsk/c5b0t0d0s33   stok17
ln -s /dev/rdsk/c5b0t0d0s34   cust17
ln -s /dev/rdsk/c5b0t0d0s35   ordl17
ln -s /dev/rdsk/c5b0t0d0s36   misc08
ln -s /dev/rdsk/c5b0t0d0s38   nord08
ln -s /dev/rdsk/c5b0t0d0s39   icust2_17
ln -s /dev/rdsk/c5b0t0d0s3a   iord1_17
ln -s /dev/rdsk/c5b0t0d0s3b   iord2_17
ln -s /dev/rdsk/c5b0t0d0s3c   unused17

ln -s /dev/rdsk/c5b0t0d0s3f   stok18
```

```
ln -s /dev/rdsk/c5b0t0d0s40   cust18
ln -s /dev/rdsk/c5b0t0d0s41   ordl18
ln -s /dev/rdsk/c5b0t0d0s42   misc09
ln -s /dev/rdsk/c5b0t0d0s44   nord09
ln -s /dev/rdsk/c5b0t0d0s45   icust2_18
ln -s /dev/rdsk/c5b0t0d0s46   iord1_18
ln -s /dev/rdsk/c5b0t0d0s47   iord2_18
ln -s /dev/rdsk/c5b0t0d0s48   unused18

ln -s /dev/rdsk/c5b0t0d0s4b   stok19
ln -s /dev/rdsk/c5b0t0d0s4c   cust19
ln -s /dev/rdsk/c5b0t0d0s4d   ordl19
ln -s /dev/rdsk/c5b0t0d0s4f   ordr10
ln -s /dev/rdsk/c5b0t0d0s51   icust2_19
ln -s /dev/rdsk/c5b0t0d0s52   iord1_19
ln -s /dev/rdsk/c5b0t0d0s53   iord2_19
ln -s /dev/rdsk/c5b0t0d0s54   unused19
ln -s /dev/rdsk/c5b0t0d0s56   temp07

ln -s /dev/rdsk/c5b0t0d0s57   stok20
#ln -s /dev/rdsk/c5b0t0d0s58  cust20
ln -s /dev/rdsk/c5b0t0d0s59   cust20
ln -s /dev/rdsk/c5b0t0d0s5a   ordl20
ln -s /dev/rdsk/c5b0t0d0s5c   ordr11
ln -s /dev/rdsk/c5b0t0d0s5e   icust2_20
ln -s /dev/rdsk/c5b0t0d0s5f   iord1_20
ln -s /dev/rdsk/c5b0t0d0s60   iord2_20
ln -s /dev/rdsk/c5b0t0d0s61   unused20
ln -s /dev/rdsk/c5b0t0d0s63   temp08
```

backup4.ln

```
#############################################################
#  Backup of First 8 data drives of cabinet 5               #
#############################################################
ln -s /dev/rdsk/c2b0t1d1s1    stok49
ln -s /dev/rdsk/c2b0t1d1s2    cust49
ln -s /dev/rdsk/c2b0t1d1s3    ordl49
ln -s /dev/rdsk/c2b0t1d1s5    ordr25
ln -s /dev/rdsk/c2b0t1d1s9    icust2_49
ln -s /dev/rdsk/c2b0t1d1sa    iord1_49
ln -s /dev/rdsk/c2b0t1d1sb    iord2_49
ln -s /dev/rdsk/c2b0t1d1sc    unused49
ln -s /dev/rdsk/c2b0t1d1se    roll01

ln -s /dev/rdsk/c2b0t1d1sf    stok50
ln -s /dev/rdsk/c2b0t1d1s10   cust50
ln -s /dev/rdsk/c2b0t1d1s11   ordl50
ln -s /dev/rdsk/c2b0t1d1s13   ordr26
ln -s /dev/rdsk/c2b0t1d1s15   icust2_50
ln -s /dev/rdsk/c2b0t1d1s16   iord1_50
ln -s /dev/rdsk/c2b0t1d1s17   iord2_50
ln -s /dev/rdsk/c2b0t1d1s18   unused50

ln -s /dev/rdsk/c2b0t1d1s1b   stok51
ln -s /dev/rdsk/c2b0t1d1s1c   cust51
ln -s /dev/rdsk/c2b0t1d1s1d   ordl51
ln -s /dev/rdsk/c2b0t1d1s1e   misc25
ln -s /dev/rdsk/c2b0t1d1s1f   ordr27
ln -s /dev/rdsk/c2b0t1d1s21   icust2_51
ln -s /dev/rdsk/c2b0t1d1s22   iord1_51
ln -s /dev/rdsk/c2b0t1d1s23   iord2_51
ln -s /dev/rdsk/c2b0t1d1s24   unused51
ln -s /dev/rdsk/c2b0t1d1s25   hist17

ln -s /dev/rdsk/c2b0t1d1s27   stok52
ln -s /dev/rdsk/c2b0t1d1s28   cust52
ln -s /dev/rdsk/c2b0t1d1s29   ordl52
ln -s /dev/rdsk/c2b0t1d1s2c   nord25
ln -s /dev/rdsk/c2b0t1d1s2d   icust2_52
ln -s /dev/rdsk/c2b0t1d1s2e   iord1_52
ln -s /dev/rdsk/c2b0t1d1s2f   iord2_52
ln -s /dev/rdsk/c2b0t1d1s30   unused52
ln -s /dev/rdsk/c2b0t1d1s31   hist18

ln -s /dev/rdsk/c2b0t1d1s33   stok53
ln -s /dev/rdsk/c2b0t1d1s34   cust53
ln -s /dev/rdsk/c2b0t1d1s35   ordl53
ln -s /dev/rdsk/c2b0t1d1s36   misc26
ln -s /dev/rdsk/c2b0t1d1s38   nord26
ln -s /dev/rdsk/c2b0t1d1s39   icust2_53
ln -s /dev/rdsk/c2b0t1d1s3a   iord1_53
ln -s /dev/rdsk/c2b0t1d1s3b   iord2_53
ln -s /dev/rdsk/c2b0t1d1s3c   unused53

ln -s /dev/rdsk/c2b0t1d1s3f   stok54
ln -s /dev/rdsk/c2b0t1d1s40   cust54
ln -s /dev/rdsk/c2b0t1d1s41   ordl54
ln -s /dev/rdsk/c2b0t1d1s42   misc27
ln -s /dev/rdsk/c2b0t1d1s44   nord27
ln -s /dev/rdsk/c2b0t1d1s45   icust2_54
ln -s /dev/rdsk/c2b0t1d1s46   iord1_54
ln -s /dev/rdsk/c2b0t1d1s47   iord2_54
ln -s /dev/rdsk/c2b0t1d1s48   unused54

ln -s /dev/rdsk/c2b0t1d1s4b   stok55
ln -s /dev/rdsk/c2b0t1d1s4c   cust55
ln -s /dev/rdsk/c2b0t1d1s4d   ordl55
ln -s /dev/rdsk/c2b0t1d1s4f   ordr28
ln -s /dev/rdsk/c2b0t1d1s51   icust2_55
ln -s /dev/rdsk/c2b0t1d1s52   iord1_55
ln -s /dev/rdsk/c2b0t1d1s53   iord2_55
ln -s /dev/rdsk/c2b0t1d1s54   unused55

ln -s /dev/rdsk/c2b0t1d1s57   stok56
#ln -s /dev/rdsk/c2b0t1d1s58  cust56
ln -s /dev/rdsk/c2b0t1d1s59   cust56
ln -s /dev/rdsk/c2b0t1d1s5a   ordl56
ln -s /dev/rdsk/c2b0t1d1s5c   ordr29
ln -s /dev/rdsk/c2b0t1d1s5e   icust2_56
ln -s /dev/rdsk/c2b0t1d1s5f   iord1_56
ln -s /dev/rdsk/c2b0t1d1s60   iord2_56
ln -s /dev/rdsk/c2b0t1d1s61   unused56
```

backup3.ln

```
#########################################################
#   Backup of Datafiles from Cabinet1 first 8 drives    #
#########################################################
ln -s /dev/rdsk/c2b0t0d0s1    stok01
ln -s /dev/rdsk/c2b0t0d0s2    cust01
ln -s /dev/rdsk/c2b0t0d0s3    ordl01
ln -s /dev/rdsk/c2b0t0d0s5    ordr01
ln -s /dev/rdsk/c2b0t0d0s9    icust2_01
ln -s /dev/rdsk/c2b0t0d0sa    iord1_01
ln -s /dev/rdsk/c2b0t0d0sb    iord2_01
ln -s /dev/rdsk/c2b0t0d0sc    unused01
ln -s /dev/rdsk/c2b0t0d0se    temp01

ln -s /dev/rdsk/c2b0t0d0sf    stok02
ln -s /dev/rdsk/c2b0t0d0s10   cust02
ln -s /dev/rdsk/c2b0t0d0s11   ordl02
ln -s /dev/rdsk/c2b0t0d0s13   ordr02
ln -s /dev/rdsk/c2b0t0d0s15   icust2_02
ln -s /dev/rdsk/c2b0t0d0s16   iord1_02
ln -s /dev/rdsk/c2b0t0d0s17   iord2_02
ln -s /dev/rdsk/c2b0t0d0s18   unused02
ln -s /dev/rdsk/c2b0t0d0s1a   temp02

ln -s /dev/rdsk/c2b0t0d0s1b   stok03
ln -s /dev/rdsk/c2b0t0d0s1c   cust03
ln -s /dev/rdsk/c2b0t0d0s1d   ordl03
ln -s /dev/rdsk/c2b0t0d0s1e   misc01
ln -s /dev/rdsk/c2b0t0d0s1f   ordr03
ln -s /dev/rdsk/c2b0t0d0s21   icust2_03
ln -s /dev/rdsk/c2b0t0d0s22   iord1_03
ln -s /dev/rdsk/c2b0t0d0s23   iord2_03
ln -s /dev/rdsk/c2b0t0d0s24   unused03
ln -s /dev/rdsk/c2b0t0d0s25   hist01

ln -s /dev/rdsk/c2b0t0d0s27   stok04
ln -s /dev/rdsk/c2b0t0d0s28   cust04
ln -s /dev/rdsk/c2b0t0d0s29   ordl04
ln -s /dev/rdsk/c2b0t0d0s2b   ctrl1
ln -s /dev/rdsk/c2b0t0d0s2c   nord01
ln -s /dev/rdsk/c2b0t0d0s2d   icust2_04
ln -s /dev/rdsk/c2b0t0d0s2e   iord1_04
ln -s /dev/rdsk/c2b0t0d0s2f   iord2_04
ln -s /dev/rdsk/c2b0t0d0s30   unused04
ln -s /dev/rdsk/c2b0t0d0s31   hist02

ln -s /dev/rdsk/c2b0t0d0s33   stok05
ln -s /dev/rdsk/c2b0t0d0s34   cust05
ln -s /dev/rdsk/c2b0t0d0s35   ordl05
ln -s /dev/rdsk/c2b0t0d0s36   misc02
ln -s /dev/rdsk/c2b0t0d0s37   ctrl2
ln -s /dev/rdsk/c2b0t0d0s38   nord02
ln -s /dev/rdsk/c2b0t0d0s39   icust2_05
ln -s /dev/rdsk/c2b0t0d0s3a   iord1_05
ln -s /dev/rdsk/c2b0t0d0s3b   iord2_05
ln -s /dev/rdsk/c2b0t0d0s3c   unused05

ln -s /dev/rdsk/c2b0t0d0s3f   stok06
ln -s /dev/rdsk/c2b0t0d0s40   cust06
ln -s /dev/rdsk/c2b0t0d0s41   ordl06
ln -s /dev/rdsk/c2b0t0d0s42   misc03
ln -s /dev/rdsk/c2b0t0d0s44   nord03
ln -s /dev/rdsk/c2b0t0d0s45   icust2_06
ln -s /dev/rdsk/c2b0t0d0s46   iord1_06
ln -s /dev/rdsk/c2b0t0d0s47   iord2_06
ln -s /dev/rdsk/c2b0t0d0s48   unused06

ln -s /dev/rdsk/c2b0t0d0s4b   stok07
ln -s /dev/rdsk/c2b0t0d0s4c   cust07
ln -s /dev/rdsk/c2b0t0d0s4d   ordl07
ln -s /dev/rdsk/c2b0t0d0s4f   ordr04
ln -s /dev/rdsk/c2b0t0d0s51   icust2_07
ln -s /dev/rdsk/c2b0t0d0s52   iord1_07
ln -s /dev/rdsk/c2b0t0d0s53   iord2_07
ln -s /dev/rdsk/c2b0t0d0s54   unused07
ln -s /dev/rdsk/c2b0t0d0s56   temp03

ln -s /dev/rdsk/c2b0t0d0s57   stok08
#ln -s /dev/rdsk/c2b0t0d0s58  cust08
ln -s /dev/rdsk/c2b0t0d0s59   cust08
ln -s /dev/rdsk/c2b0t0d0s5a   ordl08
ln -s /dev/rdsk/c2b0t0d0s5c   ordr05
ln -s /dev/rdsk/c2b0t0d0s5e   icust2_08
ln -s /dev/rdsk/c2b0t0d0s5f   iord1_08
ln -s /dev/rdsk/c2b0t0d0s60   iord2_08
ln -s /dev/rdsk/c2b0t0d0s61   unused08
ln -s /dev/rdsk/c2b0t0d0s63   temp04
```

backup2.ln

```
#########################################################
#   Backup of cabinet 4 First 8 Data drives             #
#########################################################
ln -s /dev/rdsk/c1b0t1d1s1    stok37
ln -s /dev/rdsk/c1b0t1d1s2    cust37
ln -s /dev/rdsk/c1b0t1d1s3    ordl37
ln -s /dev/rdsk/c1b0t1d1s5    ordr19
ln -s /dev/rdsk/c1b0t1d1s9    icust2_37
ln -s /dev/rdsk/c1b0t1d1sa    iord1_37
ln -s /dev/rdsk/c1b0t1d1sb    iord2_37
ln -s /dev/rdsk/c1b0t1d1sc    unused37
ln -s /dev/rdsk/c1b0t1d1se    temp13

ln -s /dev/rdsk/c1b0t1d1sf    stok38
ln -s /dev/rdsk/c1b0t1d1s10   cust38
ln -s /dev/rdsk/c1b0t1d1s11   ordl38
ln -s /dev/rdsk/c1b0t1d1s13   ordr20
```

```
ln -s /dev/rdsk/c1b0t1d1s15   icust2_38
ln -s /dev/rdsk/c1b0t1d1s16   iord1_38
ln -s /dev/rdsk/c1b0t1d1s17   iord2_38
ln -s /dev/rdsk/c1b0t1d1s18   unused38
ln -s /dev/rdsk/c1b0t1d1s1a   temp14

ln -s /dev/rdsk/c1b0t1d1s1b   stok39
ln -s /dev/rdsk/c1b0t1d1s1c   cust39
ln -s /dev/rdsk/c1b0t1d1s1d   ordl39
ln -s /dev/rdsk/c1b0t1d1s1e   misc19
ln -s /dev/rdsk/c1b0t1d1s1f   ordr21
ln -s /dev/rdsk/c1b0t1d1s20   stat01
ln -s /dev/rdsk/c1b0t1d1s21   icust2_39
ln -s /dev/rdsk/c1b0t1d1s22   iord1_39
ln -s /dev/rdsk/c1b0t1d1s23   iord2_39
ln -s /dev/rdsk/c1b0t1d1s24   unused39
ln -s /dev/rdsk/c1b0t1d1s25   hist13

ln -s /dev/rdsk/c1b0t1d1s27   stok40
ln -s /dev/rdsk/c1b0t1d1s28   cust40
ln -s /dev/rdsk/c1b0t1d1s29   ordl40
ln -s /dev/rdsk/c1b0t1d1s2c   nord19
ln -s /dev/rdsk/c1b0t1d1s2d   icust2_40
ln -s /dev/rdsk/c1b0t1d1s2e   iord1_40
ln -s /dev/rdsk/c1b0t1d1s2f   iord2_40
ln -s /dev/rdsk/c1b0t1d1s30   unused40
ln -s /dev/rdsk/c1b0t1d1s31   hist14

ln -s /dev/rdsk/c1b0t1d1s33   stok41
ln -s /dev/rdsk/c1b0t1d1s34   cust41
ln -s /dev/rdsk/c1b0t1d1s35   ordl41
ln -s /dev/rdsk/c1b0t1d1s36   misc20
ln -s /dev/rdsk/c1b0t1d1s38   nord20
ln -s /dev/rdsk/c1b0t1d1s39   icust2_41
ln -s /dev/rdsk/c1b0t1d1s3a   iord1_41
ln -s /dev/rdsk/c1b0t1d1s3b   iord2_41
ln -s /dev/rdsk/c1b0t1d1s3c   unused41

ln -s /dev/rdsk/c1b0t1d1s3f   stok42
ln -s /dev/rdsk/c1b0t1d1s40   cust42
ln -s /dev/rdsk/c1b0t1d1s41   ordl42
ln -s /dev/rdsk/c1b0t1d1s42   misc21
ln -s /dev/rdsk/c1b0t1d1s44   nord21
ln -s /dev/rdsk/c1b0t1d1s45   icust2_42
ln -s /dev/rdsk/c1b0t1d1s46   iord1_42
ln -s /dev/rdsk/c1b0t1d1s47   iord2_42
ln -s /dev/rdsk/c1b0t1d1s48   unused42

ln -s /dev/rdsk/c1b0t1d1s4b   stok43
ln -s /dev/rdsk/c1b0t1d1s4c   cust43
ln -s /dev/rdsk/c1b0t1d1s4d   ordl43
ln -s /dev/rdsk/c1b0t1d1s4f   ordr22
ln -s /dev/rdsk/c1b0t1d1s51   icust2_43
ln -s /dev/rdsk/c1b0t1d1s52   iord1_43
ln -s /dev/rdsk/c1b0t1d1s53   iord2_43
ln -s /dev/rdsk/c1b0t1d1s54   unused43
ln -s /dev/rdsk/c1b0t1d1s56   temp15

ln -s /dev/rdsk/c1b0t1d1s57   stok44
#ln -s /dev/rdsk/c1b0t1d1s58  cust44
ln -s /dev/rdsk/c1b0t1d1s59   cust44
ln -s /dev/rdsk/c1b0t1d1s5a   ordl44
ln -s /dev/rdsk/c1b0t1d1s5c   ordr23
ln -s /dev/rdsk/c1b0t1d1s5e   icust2_44
ln -s /dev/rdsk/c1b0t1d1s5f   iord1_44
ln -s /dev/rdsk/c1b0t1d1s60   iord2_44
ln -s /dev/rdsk/c1b0t1d1s61   unused44
ln -s /dev/rdsk/c1b0t1d1s63   temp16
```

backup_hist.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/hist0${I} of=$DES/hist0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 20 ]
do
   dd if=$SRC/hist${I} of=$DES/hist${I} bs=128k &
   I=`expr $I + 1`
done
```

backup_cust.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/cust0${I} of=$DES/cust0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
do
   dd if=$SRC/cust${I} of=$DES/cust${I} bs=128k &
   I=`expr $I + 1`
```

```
done
```

backup_unused.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/unused0${I} of=$DES/unused0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
do
   dd if=$SRC/unused${I} of=$DES/unused${I} bs=128k &
   I=`expr $I + 1`
done
```

backup_temp.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/temp0${I} of=$DES/temp0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 16 ]
do
   dd if=$SRC/temp${I} of=$DES/temp${I} bs=128k &
   I=`expr $I + 1`
done
```

backup_stok.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/stok0${I} of=$DES/stok0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
do
   dd if=$SRC/stok${I} of=$DES/stok${I} bs=128k &
   I=`expr $I + 1`
done
```

backup_stat.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

   dd if=$SRC/stat01 of=$DES/stat01 bs=128k &
```

backup_roll.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

   dd if=$SRC/roll01 of=$DES/roll01 bs=128k &
   dd if=$SRC/sys1 of=$DES/sys1 bs=128k &
   dd if=$SRC/ctrl1 of=$DES/ctrl1 bs=128k &
   dd if=$SRC/ctrl2 of=$DES/ctrl2 bs=128k &
```

backup_ordr.sh

```
#!/bin/ksh
```

```
SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/ordr0${I} of=$DES/ordr0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 30 ]
do
   dd if=$SRC/ordr${I} of=$DES/ordr${I} bs=128k &
   I=`expr $I + 1`
done
```

backup_ordl.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/ordl0${I} of=$DES/ordl0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
do
   dd if=$SRC/ordl${I} of=$DES/ordl${I} bs=128k &
   I=`expr $I + 1`
done
```

backup_nord.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/nord0${I} of=$DES/nord0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 30 ]
do
   dd if=$SRC/nord${I} of=$DES/nord${I} bs=128k &
   I=`expr $I + 1`
done
```

backup_misc.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/misc0${I} of=$DES/misc0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 30 ]
do
   dd if=$SRC/misc${I} of=$DES/misc${I} bs=128k &
   I=`expr $I + 1`
done
```

backup_log.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

I=1
while [ $I -le 2 ]
do
   dd if=$SRC/log${I} of=$DES/log${I} bs=128k &
   I=`expr $I + 1`
done
```

backup_iord2.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/iord2_0${I} of=$DES/iord2_0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
do
   dd if=$SRC/iord2_${I} of=$DES/iord2_${I} bs=128k &
   I=`expr $I + 1`
done
```

backup_iord1.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/iord1_0${I} of=$DES/iord1_0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
do
   dd if=$SRC/iord1_${I} of=$DES/iord1_${I} bs=128k &
   I=`expr $I + 1`
done
```

backup_icust2.sh

```
#!/bin/ksh

SRC=/oracle/8i/dbs/tpcc_disks
DES=/oracle/8i/dbs/tpcc_backup

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/icust2_0${I} of=$DES/icust2_0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
do
   dd if=$SRC/icust2_${I} of=$DES/icust2_${I} bs=128k &
   I=`expr $I + 1`
done
```

Restore Script

restore.sh

```
#!/bin/ksh

ODIR=/oracle/8i/dbs/tpcc_backup/outdir
DES=/oracle/8i/dbs/tpcc_disks
SRC=/oracle/8i/dbs/tpcc_backup


#backup_iord1.sh > ${ODIR}/backup_iord1.out 2>&1 &


I=1
while [ $I -le 9 ]
do
   dd if=$SRC/iord1_0${I} of=$DES/iord1_0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
do
   dd if=$SRC/iord1_${I} of=$DES/iord1_${I} bs=128k &
   I=`expr $I + 1`
done

#backup_ordl.sh > ${ODIR}/backup_ordl.out 2>&1 &

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/ordl0${I} of=$DES/ordl0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
```

```
do
   dd if=$SRC/ordl${I} of=$DES/ordl${I} bs=128k &
   I=`expr $I + 1`
done

wait
#backup_temp.sh > ${ODIR}/backup_temp.out 2>&1 &

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/temp0${I} of=$DES/temp0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 16 ]
do
   dd if=$SRC/temp${I} of=$DES/temp${I} bs=128k &
   I=`expr $I + 1`
done

#backup_iord2.sh > ${ODIR}/backup_iord2.out 2>&1 &

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/iord2_0${I} of=$DES/iord2_0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
do
   dd if=$SRC/iord2_${I} of=$DES/iord2_${I} bs=128k &
   I=`expr $I + 1`
done

#backup_ordr.sh > ${ODIR}/backup_ordr.out 2>&1 &

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/ordr0${I} of=$DES/ordr0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 30 ]
do
   dd if=$SRC/ordr${I} of=$DES/ordr${I} bs=128k &
   I=`expr $I + 1`
done

wait
#backup_unused.sh > ${ODIR}/backup_unused.out 2>&1 &

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/unused0${I} of=$DES/unused0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
do
   dd if=$SRC/unused${I} of=$DES/unused${I} bs=128k &
   I=`expr $I + 1`
done

#backup_cust.sh > ${ODIR}/backup_cust.out 2>&1 &

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/cust0${I} of=$DES/cust0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
do
   dd if=$SRC/cust${I} of=$DES/cust${I} bs=128k &
   I=`expr $I + 1`
done

wait
#backup_log.sh > ${ODIR}/backup_log.out 2>&1 &

I=1
while [ $I -le 2 ]
do
   dd if=$SRC/log${I} of=$DES/log${I} bs=128k &
   I=`expr $I + 1`
done

#backup_roll.sh > ${ODIR}/backup_roll.out 2>&1 &

   dd if=$SRC/roll01 of=$DES/roll01 bs=128k &
   dd if=$SRC/sys1 of=$DES/sys1 bs=128k &
   dd if=$SRC/ctrl1 of=$DES/ctrl1 bs=128k &
   dd if=$SRC/ctrl2 of=$DES/ctrl2 bs=128k &

#backup_hist.sh > ${ODIR}/backup_hist.out 2>&1 &

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/hist0${I} of=$DES/hist0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 20 ]
do
   dd if=$SRC/hist${I} of=$DES/hist${I} bs=128k &
   I=`expr $I + 1`
```

```
done

#backup_misc.sh > ${ODIR}/backup_misc.out 2>&1 &

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/misc0${I} of=$DES/misc0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 30 ]
do
   dd if=$SRC/misc${I} of=$DES/misc${I} bs=128k &
   I=`expr $I + 1`
done

#backup_stat.sh > ${ODIR}/backup_stat.out 2>&1 &

   dd if=$SRC/stat01 of=$DES/stat01 bs=128k &
wait
#backup_icust2.sh > ${ODIR}/backup_icust2.out 2>&1 &

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/icust2_0${I} of=$DES/icust2_0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
do
   dd if=$SRC/icust2_${I} of=$DES/icust2_${I} bs=128k &
   I=`expr $I + 1`
done

#backup_nord.sh > ${ODIR}/backup_nord.out 2>&1 &

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/nord0${I} of=$DES/nord0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 30 ]
do
   dd if=$SRC/nord${I} of=$DES/nord${I} bs=128k &
   I=`expr $I + 1`
done

#backup_stok.sh > ${ODIR}/backup_stok.out 2>&1 &

I=1
while [ $I -le 9 ]
do
   dd if=$SRC/stok0${I} of=$DES/stok0${I} bs=128k &
   I=`expr $I + 1`
done

I=10
while [ $I -le 60 ]
do
   dd if=$SRC/stok${I} of=$DES/stok${I} bs=128k &
   I=`expr $I + 1`
done
wait
date
```

Utilities

altdef.sh

```
#!/bin/ksh

svrmgrl <<!
 connect tpcc/tpcc;
 alter table customer enable table lock;
 alter table stock enable table lock;
 alter table item enable table lock;
 alter table history enable table lock;

 alter cluster ccluster storage(buffer_pool default);
 alter cluster scluster storage(buffer_pool default);
 alter cluster icluster storage(buffer_pool default);
 alter table history storage(buffer_pool default);

 alter table customer disable table lock;
 alter table stock disable table lock;
 alter table item disable table lock;
 alter table history disable table lock;
!
```

altundef.sh

```
#!/bin/ksh

svrmgrl <<!
 connect tpcc/tpcc;
 alter table customer enable table lock;
 alter table stock enable table lock;
 alter table item enable table lock;
 alter table history enable table lock;

 alter cluster ccluster storage(buffer_pool recycle);
```

```
 alter cluster scluster storage(buffer_pool keep);
 alter cluster icluster storage(buffer_pool keep);
 alter table history storage(buffer_pool recycle);

 alter table customer disable table lock;
 alter table stock disable table lock;
 alter table item disable table lock;
 alter table history disable table lock;
!
```

alterroll.sql

```
alter rollback segment T51 online;
alter rollback segment T52 online;
alter rollback segment T53 online;
alter rollback segment T54 online;
alter rollback segment T55 online;
alter rollback segment T56 online;
alter rollback segment T57 online;
alter rollback segment T58 online;
alter rollback segment T59 online;
alter rollback segment T60 online;
alter rollback segment T61 online;
alter rollback segment T62 online;
alter rollback segment T63 online;
alter rollback segment T64 online;
alter rollback segment T65 online;
alter rollback segment T66 online;
alter rollback segment T67 online;
alter rollback segment T68 online;
alter rollback segment T69 online;
alter rollback segment T70 online;
alter rollback segment T71 online;
alter rollback segment T72 online;
alter rollback segment T73 online;
alter rollback segment T74 online;
alter rollback segment T75 online;
alter rollback segment T76 online;
alter rollback segment T77 online;
alter rollback segment T78 online;
alter rollback segment T79 online;
alter rollback segment T80 online;
alter rollback segment T81 online;
alter rollback segment T82 online;
alter rollback segment T83 online;
alter rollback segment T84 online;
alter rollback segment T85 online;
alter rollback segment T86 online;
alter rollback segment T87 online;
alter rollback segment T88 online;
alter rollback segment T89 online;
alter rollback segment T90 online;
alter rollback segment T91 online;
alter rollback segment T92 online;
alter rollback segment T93 online;
alter rollback segment T94 online;
alter rollback segment T95 online;
alter rollback segment T96 online;
alter rollback segment T97 online;
alter rollback segment T98 online;
alter rollback segment T99 online;
alter rollback segment T100 online;

alter rollback segment T40 offline;
alter rollback segment T41 offline;
alter rollback segment T42 offline;
alter rollback segment T43 offline;
alter rollback segment T44 offline;
alter rollback segment T45 offline;
alter rollback segment T46 offline;
alter rollback segment T47 offline;
alter rollback segment T48 offline;
alter rollback segment T49 offline;
alter rollback segment T30 offline;
alter rollback segment T31 offline;
alter rollback segment T32 offline;
alter rollback segment T33 offline;
alter rollback segment T34 offline;
alter rollback segment T35 offline;
alter rollback segment T36 offline;
alter rollback segment T37 offline;
alter rollback segment T38 offline;
alter rollback segment T39 offline;
alter rollback segment T20 offline;
alter rollback segment T21 offline;
alter rollback segment T22 offline;
alter rollback segment T23 offline;
alter rollback segment T24 offline;
alter rollback segment T25 offline;
alter rollback segment T26 offline;
alter rollback segment T27 offline;
alter rollback segment T28 offline;
alter rollback segment T29 offline;
alter rollback segment T10 offline;
alter rollback segment T11 offline;
alter rollback segment T12 offline;
alter rollback segment T13 offline;
alter rollback segment T14 offline;
alter rollback segment T15 offline;
alter rollback segment T16 offline;
alter rollback segment T17 offline;
alter rollback segment T18 offline;
alter rollback segment T19 offline;
alter rollback segment T50 offline;
alter rollback segment T1 offline;
alter rollback segment T2 offline;
alter rollback segment T3 offline;
alter rollback segment T4 offline;
alter rollback segment T5 offline;
alter rollback segment T6 offline;
alter rollback segment T7 offline;
alter rollback segment T8 offline;
alter rollback segment T9 offline;
```

realter_temp.sh

```
#!/bin/ksh

# benchsetup 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                         All Rights Reserved                         |
#=====================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=====================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
LOAD_SCRIPTS=${BUILD_HOME}
OUTDIR=${BUILD_HOME}/outdir
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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


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

alter_tab.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/7/7 15:45 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                         All Rights Reserved                         |
#=====================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=====================================================================
#

BUILD_HOME=$ORACLE_HOME/bench/tpc/tpcc/scripts/build3500/
BENCH_HOME=$ORACLE_HOME/bench/tpc
BENCH_GEN=$ORACLE_HOME/bench/gen
GEN_SQL=$BUILD_HOME/sql
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_SQL=$BUILD_HOME/sql
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
TPCC_UTILS=$BUILD_HOME/utils
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql
BUILD_SQL=sql
TPCC_LOADER=$BUILD_HOME/loader
LOAD_SCRIPTS=${BUILD_HOME}/scripts
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data
STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0
MULT=3500

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

sqlplus tpcc/tpcc @alter
```

alter.sql

```
alter table customer storage (maxextents unlimited);
alter table stock storage (maxextents unlimited);
alter tablespace misc default storage (maxextents unlimited);
alter tablespace icust2 default storage (maxextents unlimited);
alter tablespace unused default storage (maxextents unlimited);
alter tablespace iord1 default storage (maxextents unlimited);
alter tablespace iord2 default storage (maxextents unlimited);
```

```
exit;
```

undml.sh

```
#
# $Header: undml.sh 7030100.2 96/05/02 10:29:30 plai Generic<base> $ Copyr
(c) 1995 Oracle
#

#+--------------------------------------------------------------------+
#|        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA         |
#|                      OPEN SYSTEMS PERFORMANCE GROUP                |
#|                           All Rights Reserved                      |
#+--------------------------------------------------------------------+
# FILENAME
#     undml.sh
# DESCRIPTION
#     Enable table locks for TPC-C tables.
# USAGE
#     undml.sh
#====================================================================*/

sqlplus tpcc/tpcc <<!
   alter table warehouse  enable table lock;
   alter table district   enable table lock;
   alter table customer   enable table lock;
   alter table history    enable table lock;
   alter table item       enable table lock;
   alter table stock      enable table lock;
   alter table orders     enable table lock;
   alter table new_order  enable table lock;
   alter table order_line enable table lock;
   quit;
!
```

dml.sh

```
#
# $Header: dml.sh 7030100.1 96/05/02 10:22:52 plai Generic<base> $ Copyr
(c) 1995 Oracle
#

#+--------------------------------------------------------------------+
#|        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA         |
#|                      OPEN SYSTEMS PERFORMANCE GROUP                |
#|                           All Rights Reserved                      |
#+--------------------------------------------------------------------+
# FILENAME
#     dml.sh
# DESCRIPTION
#     Disable table locks for TPC-C tables.
# USAGE
#     dml.sh
#====================================================================*/

sqlplus tpcc/tpcc <<!
   alter table warehouse  disable table lock;
   alter table district   disable table lock;
   alter table customer   disable table lock;
   alter table history    disable table lock;
   alter table item       disable table lock;
   alter table stock      disable table lock;
   alter table orders     disable table lock;
   alter table new_order  disable table lock;
   alter table order_line disable table lock;
   quit;
!
```

tpcc_ana.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1995  Oracle Corp, Redwood Shores, CA        |
rem |                      OPEN SYSTEMS PERFORMANCE GROUP                |
rem |                           All Rights Reserved                      |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       tpcc_ana.sql
rem  DESCRIPTION
rem       Analyze all tables and indexes of TPC-C database.
rem ====================================================================
rem

set timing on;
analyze table warehouse compute statistics;
analyze table district compute statistics;
analyze table item estimate statistics;
analyze table history estimate statistics;
analyze table customer estimate statistics;
analyze table stock estimate statistics;
analyze table orders estimate statistics;
analyze table new_order estimate statistics;
analyze table order_line estimate statistics;
analyze cluster icluster estimate statistics;
analyze cluster scluster estimate statistics;
analyze cluster ccluster estimate statistics;
analyze index iwarehouse compute statistics;
analyze index idistrict compute statistics;
analyze index icustomer estimate statistics;
analyze index icustomer2 estimate statistics;
analyze index istock estimate statistics;
analyze index iitem estimate statistics;
analyze index iorders estimate statistics;
analyze index iorders2 estimate statistics;
analyze index inord estimate statistics;
```

```
analyze index iordl estimate statistics;
quit;
```

count8i.sql

```
set echo on
set timing on
spool count8i.ver
select /*+ FULL (t) PARALLEL (t,30) */ count(*) from WAREHOUSE t;
select /*+ FULL (t) PARALLEL (t,30) */ count(*) from DISTRICT t;
select /*+ FULL (t) PARALLEL (t,60) */ count(*) from CUSTOMER t;
select /*+ FULL (t) PARALLEL (t,60) */ count(*) from STOCK t;
select /*+ FULL (t) PARALLEL (t,30) */ count(*) from ITEM t;
select /*+ FULL (t) PARALLEL (t,30) */ count(*) from NEW_ORDER t;
select /*+ FULL (t) PARALLEL (t,30) */ count(*) from ORDERS t;
select /*+ FULL (t) PARALLEL (t,60) */ count(*) from ORDER_LINE t;
select /*+ FULL (t) PARALLEL (t,20) */ count(*) from HISTORY t;
spool off
```

dbdn

```
#   Shutdown database
svrmgrl <<!
   connect internal;
   shutdown;
   exit;
!
lsnrctl <<!
stop tcp_listener
exit
!
```

dbup

```
#   Startup database
VLM_FORCE_UNMAP=n
export VLM_FORCE_UNMAP
mv $ORACLE_HOME/rdbms/log/alert_tpcc.log
$ORACLE_HOME/rdbms/log/alert_log.old
do_priocntl
svrmgrl <<!
   connect internal;
   startup pfile=admin/p_run_unisys.ora;
   exit;
!
lsnrctl <<!
start tcp_listener
exit
!
```

create_cache_views.sh

```
#!/bin/sh
# set -x
#
# This script creates four views that when queried will return
# the total number of buffers in the buffer cache and the total
# number of cloned buffers from each of the database's tablespaces.
#
# This assumes that each table and index is in its own tablespace.
# If this is not the case, another query can be used which uses the
# database's object tables to decipher the different objects.  However,
# this query is slower.
#
# This script assumes 7.3.x.  If you are using V7.2.x or below, please
# replace svrmgrl with sqldba lmode=y.
#
# Modification History:
#
# wbattist      16-Jun-1996     Create two additional views to keep
#                               track of the number of clones in each
#                               tablespace.
#
# wbattist      24-May-1995     Add the state check for the cbf view
#                               to ensure that cloned blocks are not
#                               counted.
#
svrmgrl <<EOF
set echo on
connect internal
drop view cbf;
create view cbf as
 select distinct(dbarfil) file#, count(1) blocks
 from x\$bh
   where dbarfil > 0 and state <> 3
 group by dbarfil;
drop view cbt;
create view cbt as
  select ts\$.name name,sum(cbf.blocks) buffers
  from cbf, file\$, ts\$
    where cbf.file#=file\$.file# and file\$.ts#=ts\$.ts#
  group by file\$.ts#, ts\$.name;
drop view cbfcln;
create view cbfcln as
 select distinct(dbarfil) file#, count(1) blocks
 from x\$bh
   where dbarfil > 0
 group by dbarfil;
drop view cbtcln;
create view cbtcln as
  select ts\$.name name,sum(cbfcln.blocks) buffers
  from cbfcln, file\$, ts\$
    where cbfcln.file#=file\$.file# and file\$.ts#=ts\$.ts#
  group by file\$.ts#, ts\$.name;
EOF
```

Appendix C - Tunable Parameters

Oracle Configuration Information

Oracle Makefile

The starting address of Oracle was modified from the default by changing the following line in the file ins_rdbms.mk:

```
LDCCOM=$(LINK) $(EXOSFLAGS) $(LDFLAGS) -Wl,-T01002000 -o $@
```

Oracle Executable Optimization

The default Oracle executable code was optimized using the operating system utility **fur(1)** and the scripts makelogging.unisys and mkfurred.unisys. The instructions to fur Oracle are as follows:

Login as oracle
Run the script makelogging.unisys
Start up the database
Run the workload for approximately 20 minutes
Shut down the database
Run the script mkfurred.unisys

makelogging.unisys

```
ld -r  -o oracle.o \
       -L/home/oracle/rdbms/lib/ \
       -L/home/oracle/lib/  \
       -L/usr/lib \
       /home/oracle/rdbms/lib/opimai.o      \
       /home/oracle/rdbms/lib/ssoraed.o \
       /home/oracle/rdbms/lib/ttcsoi.o      \
       /home/oracle/rdbms/lib/config.o \
       -ulxgucs2utf \
       -uremainder \
       -urint \
       -ujnistatic \
       -ulxgutf2ucs \
       -lserver8 \
       -lclient8 \
       -lvsn8 \
       -lcommon8 \
       -lskgxp8 \
       -lgeneric8 \
       -lpls8 \
       /home/oracle/rdbms/lib/defopt.o \
       /home/oracle/lib/ncrstab.o \
       -lknlopt \
       -lslax8 \
       -lpls8 \
       -lplp8 \
       -lserver8 \
       -lclient8 \
       -lvsn8 \
       -lcommon8 \
       -lskgxp8 \
       -lgeneric8 \
       -lpls8 \
       -lknlopt \
       -lslax8 \
       -lpls8 \
       -lplp8 \
       -ldbicx8 \
       -ldbicx8 \
       -lserver8 \
       /home/oracle/lib/nautab.o \
       /home/oracle/lib/naeet.o \
       /home/oracle/lib/naect.o \
       /home/oracle/lib/naedhs.o  \
       -lnbeq8 \
       -lnhost8 \
       -ln8 \
       -lncrypt8 \
       -lnoss8 \
       -lnidx8 \
       -ln8 \
       -lncrypt8 \
       -lnus8 \
       -ln8 \
       -lncrypt8 \
       -lnoss8 \
       -lnk58 \
       -ln8 \
       -lncrypt8 \
       -lnldap8 \
       -lldapclnt8 \
       -lnsslb8 \
       -lnoss8 \
       -ln8 \
       -lncrypt8 \
       -lnoss8 \
```

```
-ln8 \
-lncrypt8 \
-lnoname8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnsid8 \
-ln8 \
-lncrypt8 \
-lntcp8 \
-lntcps8 \
-lnsslb8 \
-lntcp8 \
-lntns8 \
-ln8 \
-lnl8 \
-lnro8  \
-lnbeq8 \
-lnhost8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnidx8 \
-ln8 \
-lncrypt8 \
-lnus8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnk58 \
-ln8 \
-lncrypt8 \
-lnldap8 \
-lldapclnt8 \
-lnsslb8 \
-lnoss8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-ln8 \
-lncrypt8 \
-lnoname8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnsid8 \
-ln8 \
-lncrypt8 \
-lntcp8 \
-lntcps8 \
-lnsslb8 \
-lntcp8 \
-lntns8 \
-ln8 \
-lnl8 \
-lclient8 \
-lvsn8 \
-lcommon8 \
-lskgxp8 \
-lgeneric8 \
-lpls8 \
-lnls8  \
-lcore8 \
-lnls8 \
-lcore8 \
-lnls8   \
-ln8 \
-lnbeq8 \
-lnhost8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnidx8 \
-ln8 \
-lncrypt8 \
-lnus8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnk58 \
-ln8 \
-lncrypt8 \
-lnldap8 \
-lldapclnt8 \
-lnsslb8 \
-lnoss8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-ln8 \
-lncrypt8 \
-lnoname8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnsid8 \
-ln8 \
-lncrypt8 \
-lntcp8 \
-lntcps8 \
-lnsslb8 \
-lntcp8 \
-lntns8 \
-ln8 \
-lnl8 \
-lnro8  \
-lnbeq8 \
-lnhost8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnidx8 \
-ln8 \
-lncrypt8 \
-lnus8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
```

```
-lnk58 \
-ln8 \
-lncrypt8 \
-lnldap8 \
-lldapclnt8 \
-lnsslb8 \
-lnoss8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-ln8 \
-lncrypt8 \
-lnoname8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnsid8 \
-ln8 \
-lncrypt8 \
-lntcp8 \
-lntcps8 \
-lnsslb8 \
-lntcp8 \
-lntns8 \
-ln8 \
-lnl8 \
-lclient8 \
-lvsn8 \
-lcommon8 \
-lskgxp8 \
-lgeneric8 \
-lpls8 \
-ltrace8 \
-lnls8  \
-lcore8 \
-lnls8 \
-lcore8 \
-lnls8  \
-lnbeq8 \
-lnhost8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnidx8 \
-ln8 \
-lncrypt8 \
-lnus8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnk58 \
-ln8 \
-lncrypt8 \
-lnldap8 \
-lldapclnt8 \
-lnsslb8 \
-lnoss8 \
-ln8 \
-lncrypt8 \
```

```
-lnoss8 \
-ln8 \
-lncrypt8 \
-lnoname8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnsid8 \
-ln8 \
-lncrypt8 \
-lntcp8 \
-lntcps8 \
-lnsslb8 \
-lntcp8 \
-lntns8 \
-ln8 \
-lnl8 \
-lnro8  \
-lnbeq8 \
-lnhost8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnidx8 \
-ln8 \
-lncrypt8 \
-lnus8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnk58 \
-ln8 \
-lncrypt8 \
-lnldap8 \
-lldapclnt8 \
-lnsslb8 \
-lnoss8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-ln8 \
-lncrypt8 \
-lnoname8 \
-ln8 \
-lncrypt8 \
-lnoss8 \
-lnsid8 \
-ln8 \
-lncrypt8 \
-lntcp8 \
-lntcps8 \
-lnsslb8 \
-lntcp8 \
-lntns8 \
-ln8 \
-lnl8 \
-lclient8 \
-lvsn8 \
-lcommon8 \
```

```
        -lskgxp8 \
        -lgeneric8 \
        -lpls8 \
        -lnls8  \
        -lcore8 \
        -lnls8 \
        -lcore8 \
        -lnls8 \
        -lgen \
        -lelf \
        -lm \
        -lgen \
        -lm \
        -lserver8 \
        /home/oracle/rdbms/lib/skgxns.o \
        -lordsdo8 \
        -lordts8 \
        -lctxc8 \
        -lctx8 \
        -lnls8  \
        -lcore8 \
        -lnls8 \
        -lcore8 \
        -lnls8     \
        -lgen \
        -lelf \
        -lm \
        -lgen \
        -lm  \
        -lordvirt8


# save oracle.o for later (makefurred)
cp oracle.o save.oracle.o

# setup logging directory path, if needed.
LOG_DIR=`pwd`/blocklog.dir
if [ ! -d $LOG_DIR ]
then
        rm -rf $LOG_DIR
        mkdir $LOG_DIR
fi
rm -rf $LOG_DIR/*

# create logging version of oracle.o and log.oracle.o
rm -f keep.oracle
fur -O DEBUG=1 -b all -K keep.oracle -c "mkblocklog -p
$LOG_DIR/block.oracle.o" oracle.o

LD_RUN_PATH=/home/oracle/lib
export LD_RUN_PATH

# create logging version of oracle
cc   -Wl,-T01002000 -o /home/oracle/bin/oracle oracle.o \
        -L/home/oracle/rdbms/lib/ -L/home/oracle/lib/  \
        /home/oracle/lib/libjox8.so \
        -lobk \
        -ldshm  \
        -lsocket \
        -lnsl \
        -ldl \
        -Kthread log.oracle.o

#To ensure oracle has the right owner, group, modes, and privileges,run
cd /home/oracle/bin
chown oracle oracle
chgrp dba oracle
chmod 6755 oracle
st filepriv -f plock `pwd`/oracle
```

mkfurred.unisys

```
function mk_list {
    if [[ $# = 0 ]]
    then
      return
    fi
    LIST=$1
    shift
    while [[ $# > 0 ]] do
      LIST=$LIST,$1
      shift
    done
    echo $LIST
}

LD_RUN_PATH=/home/oracle/lib
export LD_RUN_PATH

# retreive copy of oracle.o from save version
cp save.oracle.o oracle.o

# setup logging directory path, if needed.
LOG_DIR=`pwd`/blocklog.dir
if [ ! -d $LOG_DIR ]
then
        rm -rf $LOG_DIR
        mkdir $LOG_DIR
fi

# determine ordering information based on collected logs
fur -r -o oracle.order -k keep.oracle -f `mk_list
$LOG_DIR/block.oracle.o.*` \
        -m oracle.o

# fur -k keep.oracle -o oracle.order oracle.o
fur -o oracle.order oracle.o

# create logging version of oracle
cc    -Wl,-T01002000 -o /home/oracle/bin/oracle oracle.o \
        -L/home/oracle/rdbms/lib/ -L/home/oracle/lib/  \
        /home/oracle/lib/libjox8.so \
        -lobk \
        -ldshm  \
        -lsocket \
```

```
        -lnsl \
        -ldl \
        -Kthread

#To ensure oracle has the right owner, group, modes, and privileges, run
cd /home/oracle/bin
chown oracle oracle
chgrp dba oracle
chmod 6755 oracle
st filepriv -f plock `pwd`/oracle
```

prun_unisys.ora

```
control_files                       = (/oracle/8i/dbs/tpcc_disks/ctrl1,
/oracle/8i/dbs/tpcc_disks/ctrl2)
lock_sga                           = TRUE
disk_asynch_io                     = TRUE
fast_start_io_target = 0
db_block_max_dirty_target          = 3000000
_disable_incremental_checkpoints = TRUE
_db_writer_max_writes = 512
_db_writer_chunk_writes = 512
create_bitmap_area_size = 51200
parallel_max_servers                = 80
recovery_parallelism                = 20
compatible                          = 8.1.3.0.0
db_name                             = tpcc
db_files                            = 550
_db_file_noncontig_mblock_read_count     = 1
db_file_multiblock_read_count       = 1
db_block_lru_latches                = 9
buffer_pool_recycle                  = (buffers:148660, lru_latches:3)
buffer_pool_keep                     = (buffers:5946400, lru_latches:3)
dml_locks                           = 520
enqueue_resources                   = 34000
hash_join_enabled                   = FALSE
java_pool_size                     = 1048576
shared_pool_size                    = 20480000
log_archive_start                    = FALSE
log_checkpoint_interval             = 0
log_checkpoint_timeout               = 1000000000
log_checkpoints_to_alert            = TRUE
log_buffer                          = 8388608
lm_ress                            = 256
gc_releasable_locks                  = 0
max_rollback_segments               = 320
open_cursors                        = 400
session_cached_cursors              = 255
cursor_space_for_time               = TRUE
processes                           = 300
sessions                            = 300
distributed_transactions            = 0
timed_statistics                    = FALSE
transactions_per_rollback_segment = 1
rollback_segments                   = (
t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,\
t11,t12,t13,t14,t15,t16,t17,t18,t19,\
t20,t21,t22,t23,t24,t25,t26,t27,t28,t29,\
t30,t31,t32,t33,t34,t35,t36,t37,t38,t39,\
t40,t41,t42,t43,t44,t45,t46,t47,t48,t49,\
t50,t51,t52,t53,t54,t55,t56,t57,t58,t59,\
t60,t61,t62,t63,t64,t65,t66,t67,t68,t69,\
t70,t71,t72,t73,t74,t75,t76,t77,t78,t79,\
t80,t81,t82,t83,t84,t85,t86,t87,t88,t89,\
t90,t91,t92,t93,t94,t95,t96,t97,t98,t99,t100,\
t101,t102,t103,t104,t105,t106,t107,t108,t109,t110,\
t111,t112,t113,t114,t115,t116,t117,t118,t119,t120,\
t121,t122,t123,t124,t125,t126,t127,t128,t129,t130,\
t131,t132,t133,t134,t135,t136,t137,t138,t139,t140,\
t141,t142,t143,t144,t145,t146,t147,t148,t149,t150,\
t151,t152,t153,t154,t155,t156,t157,t158,t159,t160,\
t161,t162,t163,t164,t165,t166,t167,t168,t169,t170,\
t171,t172,t173,t174,t175,t176,t177,t178,t179,t180,\
t181,t182,t183,t184,t185,t186,t187,t188,t189,t190,\
t191,t192,t193,t194,t195,t196,t197,t198,t199,t200,\
t201,t202,t203,t204,t205,t206,t207,t208,t209,t210,\
t211,t212,t213,t214,t215,t216,t217,t218,t219,t220,)

_db_aging_hot_criteria = 2
_db_aging_stay_count = 0
max_dump_file_size     = 20
db_block_buffers                    = 7433000
use_indirect_data_buffers           = true
replication_dependency_tracking = FALSE
transaction_auditing = FALSE
_spin_count = 8000
_enable_list_io = TRUE
```

Checkpoint.sh

```
#================================================================+
#        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA     |
#                     OPEN SYSTEMS PERFORMANCE GROUP             |
#                          All Rights Reserved                   |
#================================================================+
# FILENAME
#     switchlog.sh
# DESCRIPTION
#     Switch to next log file twice.
# USAGE
#     switchlog.sh
#================================================================*/

I=1
while [ $I -le 16 ]
do
sleep 900
/oracle/8i/bin/svrmgrl > switch.out <<!
   connect internal;
   alter system switch logfile;
   exit;
```

```
!

I=`expr $I + 1`
done
```

Profile for Oracle User

```
export TERM=at386
export EDITOR=vi
export PS1='orunix:$PWD> '
#export USE_PW=1
export CPDIR=/oracle/8i/bench/tpc/tpcc/scripts/build3500
export ADIR=/oracle/8i/bench/tpc/tpcc/audit-uw
stty intr ^C
export JAVA_HOME=/usr/java
export ORACLE_BASE=/oracle
export ORACLE_HOME=/oracle/8i
export TMPDIR=/oracle/temp
export SRCHOME=/oracle/8i
export
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/java/lib/x86at/native_threads:/usr/j
ava/lib
export ORACLE_SID=tpcc
export
PATH=$ORACLE_HOME/bin:/bin:/opt/bin:/usr/bin:/usr/sbin:/usr/java/bin:$PATH
:$ORACLE_HOME/bench/tpc/bin:$ORACLE_HOME/bench/tpc/tpcc/scripts:/usr/ccs/b
in:/etc:.:/usr/bin/X11:/usr/ucb
export ORA_NLS=$ORACLE_HOME/ocommon/nls/admin/data
export ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data
export TNS_ADMIN=$ORACLE_HOME/network/admin
#export MAP_SIZE=511
do_priocntl
export DSHM_DEBUG=TRUE
export VLM_FORCE_UNMAP=n
export USE_PW=1              # use postwait instead of semaphores
export VLM_FORCE_UNMAP=n     # don't force dshm unmaps
#export SCO_OPTIM1=1   # skgfdisp: synchronous pread =>
lio_listio(LIO_WAIT).
export SCO_OPTIM2=1     # skgfrsini: sets SKGFR_USELIO if 1, clears it if
0.
export SCO_OPTIM3=1     # skgflio: lio_listio(1 req) ->
aio_read/aio_write.
export SCO_OPTIM4=1     # skgfrliopo: disables the first call to
aio_suspend
export SCO_OPTIM5=1     # future
```

listener.ora

```
############
#
#Listener name = TCP_LISTENER
#Created by Unisys team
#Date : 10/03/98
#
# ############

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
       (SDU = 4096)
       (TDU = 16384)
      (SID_NAME=tpcc)
      (ORACLE_HOME=/oracle/8i)
    ))
TCP_LISTENER=
    (ADDRESS_LIST =
          (ADDRESS=
               (PROTOCOL=TCP)
               (HOST= 192.59.13.114)
               (Port=1521)
           ))

SID_LIST_TCP_LISTENER =
  (SID_LIST =
    (SID_DESC =
       (SDU = 4096)
       (TDU = 16384)
      (SID_NAME=tpcc)
      (ORACLE_HOME=/oracle/8i)
    ))

STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER = OFF
```

tnsnames.ora

```
# TNSNAMES.ORA Configuration File:/oracle/8i/network/admin/tnsnames.ora
# Generated by Oracle Net8 Assistant

orunix =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.59.13.114)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = tpcc )
    )
  )

EXTPROC_CONNECTION_DATA.MV.UNISYS.COM =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
```

```
      (PRESENTATION = RO)
    )
  )
```

do_priocntl

```
#!/usr/bin/xksh -p
call setuid 0
# put ppid into FP class at pri 105 w/ 1 sec time slice
priocntl -s -c FP -p 5 -t 100 -r 100 $PPID

# put inetd into FP class at pri 105 w/ 1 sec time slice
set -- `ps -e | egrep 'inetd$'`
inetdpid=$1
if [ -n "$inetdpid" ]
then
  priocntl -s -c FP -p 5 -t 100 -r 100 $inetdpid
else
  echo could not find inetd to make it fixed priority >&2
fi
```

st

```
#!/usr/bin/xksh -p
PATH=$PATH:/sbin
PS1='# '
call setuid 0
if [ -n "$*" ]
then
        if type "$1" >/dev/null
        then
                exec "$@"
        else
                eval "$@"
        fi
else
        exec ksh -o vi
fi
```

Server Configuration Information

Memory and Disks

```
    SYSTEM CONFIGURATION:

    Memory Size: 16384 Megabytes
    System Peripherals:

        Floppy Drive 1 - 1.44 MB 3.5
        SCSI CD-Rom Drive 1 - TOSHIBA  - CD-ROM XM-5701TA
        Disk Drive 1 - SEAGATE  - ST34501W         - 4337 MB
        Disk Drive 10 - DGC      - RAID 0           - 84681 MB
        Disk Drive 100 - DGC      - RAID 5
        Disk Drive 101 - DGC      - RAID 5
        Disk Drive 102 - DGC      - RAID 0           - 84681 MB
        Disk Drive 103 - DGC      - RAID 0           - 84681 MB
        Disk Drive 104 - DGC      - RAID 0           - 84681 MB
        Disk Drive 105 - DGC      - RAID 0           - 84681 MB
        Disk Drive 106 - DGC      - RAID 0           - 84681 MB
        Disk Drive 107 - DGC      - RAID 0           - 84681 MB
        Disk Drive 108 - DGC      - RAID 0
        Disk Drive 109 - DGC      - RAID 0
        Disk Drive 11 - DGC      - RAID 0           - 84681 MB
        Disk Drive 110 - DGC      - RAID 0
        Disk Drive 111 - DGC      - RAID 0
        Disk Drive 112 - DGC      - RAID 0
        Disk Drive 113 - DGC      - RAID 0
        Disk Drive 114 - DGC      -
        Disk Drive 115 - DGC      -
        Disk Drive 116 - DGC      - RAID 5           - 152426 MB
        Disk Drive 117 - DGC      - RAID 0
        Disk Drive 118 - DGC      - RAID 0
        Disk Drive 119 - DGC      - RAID 0
        Disk Drive 12 - DGC      - RAID 0
        Disk Drive 120 - DGC      - RAID 0
        Disk Drive 121 - DGC      - RAID 0
        Disk Drive 122 - DGC      - RAID 0
        Disk Drive 123 - DGC      - RAID 0           - 84681 MB
        Disk Drive 124 - DGC      - RAID 0           - 84681 MB
        Disk Drive 125 - DGC      - RAID 0           - 84681 MB
        Disk Drive 126 - DGC      - RAID 0           - 84681 MB
        Disk Drive 127 - DGC      - RAID 0           - 84681 MB
        Disk Drive 128 - DGC      - RAID 0           - 84681 MB
        Disk Drive 129 - DGC      -
        Disk Drive 13 - DGC      - RAID 0
        Disk Drive 130 - DGC      -
        Disk Drive 131 - DGC      - RAID 5           - 152426 MB
        Disk Drive 132 - DGC      - RAID 5
        Disk Drive 133 - DGC      - RAID 5
        Disk Drive 134 - DGC      - RAID 0           - 84681 MB
        Disk Drive 135 - DGC      - RAID 0           - 84681 MB
        Disk Drive 136 - DGC      - RAID 0           - 84681 MB
        Disk Drive 137 - DGC      - RAID 0           - 84681 MB
        Disk Drive 138 - DGC      - RAID 0           - 84681 MB
        Disk Drive 139 - DGC      - RAID 0           - 84681 MB
        Disk Drive 14 - DGC      - RAID 0
        Disk Drive 140 - DGC      - RAID 0
        Disk Drive 141 - DGC      - RAID 0
        Disk Drive 142 - DGC      - RAID 0
        Disk Drive 143 - DGC      - RAID 0
        Disk Drive 144 - DGC      - RAID 0
        Disk Drive 145 - DGC      - RAID 0
        Disk Drive 146 - DGC      -
        Disk Drive 147 - DGC      -
        Disk Drive 148 - DGC      - RAID 5           - 152426 MB
        Disk Drive 149 - DGC      - RAID 0
        Disk Drive 15 - DGC      - RAID 0
        Disk Drive 150 - DGC      - RAID 0
        Disk Drive 151 - DGC      - RAID 0
```

```
Disk Drive 152 - DGC      - RAID 0
Disk Drive 153 - DGC      - RAID 0
Disk Drive 154 - DGC      - RAID 0
Disk Drive 155 - DGC      - RAID 0          - 84681 MB
Disk Drive 156 - DGC      - RAID 0          - 84681 MB
Disk Drive 157 - DGC      - RAID 0          - 84681 MB
Disk Drive 158 - DGC      - RAID 0          - 84681 MB
Disk Drive 159 - DGC      - RAID 0          - 84681 MB
Disk Drive 16 - DGC      - RAID 0
Disk Drive 160 - DGC      - RAID 0          - 84681 MB
Disk Drive 161 - DGC      -
Disk Drive 162 - DGC      -
Disk Drive 163 - DGC      - RAID 10         - 84680 MB
Disk Drive 164 - DGC      - RAID 10
Disk Drive 165 - DGC      - RAID 10         - 84680 MB
Disk Drive 166 - DGC      - RAID 10
Disk Drive 167 - DGC      -
Disk Drive 168 - DGC      -
Disk Drive 169 - DGC      -
Disk Drive 17 - DGC      - RAID 0
Disk Drive 170 - DGC      -
Disk Drive 171 - DGC      -
Disk Drive 172 - DGC      -
Disk Drive 173 - DGC      -
Disk Drive 174 - DGC      -
Disk Drive 175 - DGC      -
Disk Drive 176 - DGC      -
Disk Drive 177 - DGC      -
Disk Drive 178 - DGC      -
Disk Drive 179 - DGC      -
Disk Drive 18 - DGC      -
Disk Drive 180 - DGC      - RAID 10
Disk Drive 181 - DGC      - RAID 10         - 84680 MB
Disk Drive 182 - DGC      -                 - 84680 MB
Disk Drive 183 - DGC      -
Disk Drive 184 - DGC      -
Disk Drive 185 - DGC      -
Disk Drive 186 - DGC      -
Disk Drive 187 - DGC      -
Disk Drive 188 - DGC      -
Disk Drive 189 - DGC      -
Disk Drive 19 - DGC      -
Disk Drive 190 - DGC      -
Disk Drive 191 - DGC      -
Disk Drive 192 - DGC      -
Disk Drive 193 - DGC      -
Disk Drive 194 - DGC      -
Disk Drive 195 - DGC      - RAID 10
Disk Drive 196 - DGC      - RAID 10         - 84680 MB
Disk Drive 2 - UNISYS    - 003665MAB3045SC  - 4343 MB
Disk Drive 20 - DGC      - RAID 5          - 152426 MB
Disk Drive 21 - DGC      - RAID 0
Disk Drive 22 - DGC      - RAID 0
Disk Drive 23 - DGC      - RAID 0
Disk Drive 24 - DGC      - RAID 0
Disk Drive 25 - DGC      - RAID 0
Disk Drive 26 - DGC      - RAID 0
Disk Drive 27 - DGC      - RAID 0          - 84681 MB
Disk Drive 28 - DGC      - RAID 0          - 84681 MB
Disk Drive 29 - DGC      - RAID 0          - 84681 MB
Disk Drive 3 - DGC      - RAID 5          - 152426 MB
Disk Drive 30 - DGC      - RAID 0          - 84681 MB
Disk Drive 31 - DGC      - RAID 0          - 84681 MB
Disk Drive 32 - DGC      - RAID 0          - 84681 MB
Disk Drive 33 - DGC      -
Disk Drive 34 - DGC      -
Disk Drive 35 - DGC      - RAID 5          - 152426 MB
Disk Drive 36 - DGC      - RAID 5
Disk Drive 37 - DGC      - RAID 5
Disk Drive 38 - DGC      - RAID 0          - 84681 MB
Disk Drive 39 - DGC      - RAID 0          - 84681 MB
Disk Drive 4 - DGC      - RAID 5
Disk Drive 40 - DGC      - RAID 0          - 84681 MB
Disk Drive 41 - DGC      - RAID 0          - 84681 MB
Disk Drive 42 - DGC      - RAID 0          - 84681 MB
Disk Drive 43 - DGC      - RAID 0          - 84681 MB
Disk Drive 44 - DGC      - RAID 0
Disk Drive 45 - DGC      - RAID 0
Disk Drive 46 - DGC      - RAID 0
Disk Drive 47 - DGC      - RAID 0
Disk Drive 48 - DGC      - RAID 0
Disk Drive 49 - DGC      - RAID 0
Disk Drive 5 - DGC      - RAID 5
Disk Drive 50 - DGC      -
Disk Drive 51 - DGC      -
Disk Drive 52 - DGC      - RAID 5          - 152426 MB
Disk Drive 53 - DGC      - RAID 0
Disk Drive 54 - DGC      - RAID 0
Disk Drive 55 - DGC      - RAID 0
Disk Drive 56 - DGC      - RAID 0
Disk Drive 57 - DGC      - RAID 0
Disk Drive 58 - DGC      - RAID 0
Disk Drive 59 - DGC      - RAID 0          - 84681 MB
Disk Drive 6 - DGC      - RAID 0          - 84681 MB
Disk Drive 60 - DGC      - RAID 0          - 84681 MB
Disk Drive 61 - DGC      - RAID 0          - 84681 MB
Disk Drive 62 - DGC      - RAID 0          - 84681 MB
Disk Drive 63 - DGC      - RAID 0          - 84681 MB
Disk Drive 64 - DGC      - RAID 0          - 84681 MB
Disk Drive 65 - DGC      -
Disk Drive 66 - DGC      -
Disk Drive 67 - DGC      - RAID 5          - 152425 MB
Disk Drive 68 - DGC      - RAID 5
Disk Drive 69 - DGC      - RAID 0
Disk Drive 7 - DGC      - RAID 0          - 84681 MB
Disk Drive 70 - DGC      - RAID 0          - 84681 MB
Disk Drive 71 - DGC      - RAID 0          - 84681 MB
Disk Drive 72 - DGC      - RAID 0          - 84681 MB
Disk Drive 73 - DGC      - RAID 0          - 84681 MB
Disk Drive 74 - DGC      - RAID 0          - 84681 MB
Disk Drive 75 - DGC      - RAID 0          - 84681 MB
Disk Drive 76 - DGC      - RAID 0
Disk Drive 77 - DGC      - RAID 0
Disk Drive 78 - DGC      - RAID 0
Disk Drive 79 - DGC      - RAID 0
Disk Drive 8 - DGC      - RAID 0          - 84681 MB
Disk Drive 80 - DGC      - RAID 0
Disk Drive 81 - DGC      - RAID 0
```

```
Disk Drive 82 - DGC      -
Disk Drive 83 - DGC      -
Disk Drive 84 - DGC      - RAID 0           - 169362 MB
Disk Drive 85 - DGC      - RAID 0
Disk Drive 86 - DGC      - RAID 0
Disk Drive 87 - DGC      - RAID 0
Disk Drive 88 - DGC      - RAID 0
Disk Drive 89 - DGC      - RAID 0
Disk Drive 9 - DGC      - RAID 0           - 84681 MB
Disk Drive 90 - DGC      - RAID 0
Disk Drive 91 - DGC      - RAID 0           - 84681 MB
Disk Drive 92 - DGC      - RAID 0           - 84681 MB
Disk Drive 93 - DGC      - RAID 0           - 84681 MB
Disk Drive 94 - DGC      - RAID 0           - 84681 MB
Disk Drive 95 - DGC      - RAID 0           - 84681 MB
Disk Drive 96 - DGC      - RAID 0           - 84681 MB
Disk Drive 97 - DGC      -
Disk Drive 98 - DGC      -
Disk Drive 99 - DGC      - RAID 5           - 152426 MB
80387 Math Processor
```

List of Packages

```
application BASEdoc        UnixWare Documentation
application BASEman        UnixWare Manual Pages
system      DLPI           BUILT IN TO NETWORK INTERFACE CARD SUPPORT,
CANNOT BE REMOVED.
application FTRKdoc        Netscape FastTrack Server 2.01 Documentation
system      MDI            BUILT IN TO NETWORK INTERFACE CARD SUPPORT,
CANNOT BE REMOVED.
application TEDdesk        TriTeal Enterprise Desktop (CDE Desktop)
application TEDdocs        CDE Desktop Postscript Manuals
application TEDhelp        CDE Online Help
application TEDlogin       CDE Login Manager
application TEDman         CDE Manual Pages
system      acp            Enhanced Application Compatibility
system      adsl           Adaptec 7800 Family PCI SCSI IHV HBA
system      base           Base System
graphics    basex          X11R6 Base X Runtime System
system      bsdcompat      BSD Compatibility
system      cdfs           BUILT INTO THE BASE, CANNOT BE REMOVED.
system      cmds           Advanced Commands
system      compat         BUILT INTO BSD COMPATIBILITY, CANNOT BE
REMOVED.
system      dfm            BUILT INTO THE BASE, CANNOT BE REMOVED.
utilities   dfs            Distributed File System Utilities
system      dshm           Large Memory Support Package
system      ed             BUILT INTO ADVANCED COMMANDS, CANNOT BE
REMOVED.
system      els            BUILT INTO LANGUAGE SUPPLEMENT, CANNOT BE
REMOVED.
system      eth            BUILT INTO NETWORK INTERFACE CARD SUPPORT,
CANNOT BE REMOVED.
system      expect         expect
system      fmli           BUILT INTO ADVANCED COMMANDS, CANNOT BE
REMOVED.
system      ide            Generic IDE/ATAPI/ESDI Driver
system      inet           Internet Utilities
system      jdk117         JDK 1.1.7 for SCO
system      kdb            Kernel Debugger
system      ls             Language Supplement
system      mouse          BUILT INTO THE BASE, CANNOT BE REMOVED.
graphics    mtfrun         OSF Motif 1.2.5 Runtime Environment
system      nd             Network Drivers
system      netcmds        BUILT INTO NETWORK SUPPORT UTILITIES, CANNOT BE
REMOVED.
system      nfs            Network File System Utilities
system      nics           Netdriver Infrastructure and Configuration
Subsystem
application nscomm         Netscape Communicator 4.08
application nscommD        Netscape Communicator 4.08 Documentation
system      nsu            Network Support Utilities
system      osmp           OS Multiprocessor Support (OSMP)
utilities   perl5          Perl 5.004
system      qlc2100        QLogic QLC2100 Fiber Channel IHV HBA Driver
system      qt             BUILT INTO THE BASE, CANNOT BE REMOVED.
system      rpc            Remote Procedure Calls Utilities
system      scoadmin       SCO System Administration (SCOadmin)
application scohelp        Scohelp Online Doc System
system      sys            BUILT INTO THE BASE, CANNOT BE REMOVED.
system      tclrun         Tcl Runtime package
system      termcap        BUILT INTO ENHANCED APPLICATION COMPATIBILITY,
CANNOT BE REMOVED.
system      terminf        Terminfo Utilities
system      tok            BUILT INTO NETWORK INTERFACE CARD SUPPORT,
CANNOT BE REMOVED.
update      u71rs1         UnixWare 7 Update 7.1.1
system      uccs           UDK Optimizing C Compilation System
system      uedebug        UDK Enhanced Debugger
update      update710      UnixWare 7 Update 7.1.0
system      vtclrun        Vtcl Runtime package
system      vxvm           VERITAS Volume Manager
graphics    xclients       X11R6 X Clients
graphics    xcontrib       X11R6 Contributed X Clients
graphics    xdrivers       X11R6 Graphics Drivers
graphics    xfonts         X11R6 Fonts
graphics    xserver        X11R6 X Server


Custom Installed Packages

ALL     Part    Entire Run Time System with perms lists
RTS     Part    UNIX Run Time System
ALL     Part    Entire TCP/IP Runtime System
TCPRT   Part    TCP/IP Runtime utilities and drivers
SNDMAIL Part    Sendmail Runtime
NETBIOS Part    NetBIOS Runtime utilities and drivers
PPP     Part    PPP Runtime utilities and drivers
```

stune

```
NUMREGPT        256
NUMSCOPT        32
DRV_SEGKMEM_BYTES       0x147A000
RSTCHOWN        0
CONSOLE_SECURITY        0
SHMMNI  100
SHMSEG  6
USER_RDTSC      1
SEMMSL  300
SHMMAX  2147483647
SEMMNI  1450
SEMMNS  610
NPROC   2000
SDATLIM 0x7FFFFFFF
HDATLIM 0x7FFFFFFF
SVMMLIM 0x7FFFFFFF
HVMMLIM 0x7FFFFFFF
SFSZLIM 0x7FFFFFFF
HFSZLIM 0x7FFFFFFF
MAXUP   2000
MSGSSZ  1048576
MSGMNB  131072
DEDICATED_MEMORY        16777216
MAXRSS  125000
ET_AGE_INTERVAL         10
INIT_AGEQUANTUM         100
MIN_AGEQUANTUM 50
MAX_AGEQUANTUM 120
MSGMAX  8192
MSGMNI  1650
MSGTQL  1650
SFNOLIM 128
NUMAIO  16384
AIO_LISTIO_MAX 512
FDFLUSHR        1000
LOTSFREEBYTES   262144
MINFREEBYTES    32768
DESFREEBYTES    102400
GENERAL_MEMORY 160000
FLCKREC 1024
SEGKMEM_PSE_BYTES       104857600
MAXULWP 2000
SEMOPM  20
MAXCACHEWARM    300
SYSDUMP_SELECTIVE       1
SYSDUMP_ABORT_VERIFY    1
ARG_MAX 1048576
KMAGBTIME       1200
UVIRT_EXTENSION         131072
```

Internet Protocol Stack

The default internet protocol stack was modified from the default values by issuing the commands:

inconfig tcp_delay_acks 1
inconfig tcp_do_rfc1323 0
inconfig tcp_fast_open 1
inconfig tcp_2msl 60

eeE Driver

The eeE driver was modified from the default by changing the value of TX_THRSHLD from 8 to 200 in the file /etc/conf/pack.d/eeE/space.c.

Inet Driver

The inet driver was modified from the default by changing the value of tcp_pcb_cache_size from 32 to 256 in the file /etc/conf/pack.d/inet/space.c.

Unix Kernel Optimization

The default Unix kernel executable code was optimized using the operating system utility **fur(1).** The instructions to fur the Unix kernel are:

Issue the command /etc/conf/bin/idoptimize -C as super user. This resets /etc/conf/optimize.d to its original state.
Issue the command /etc/conf/bin/idoptimize -c as super user. This clears out the logs in the running kernel.
Start Oracle and run the database workload for 10 minutes.
Issue the command /etc/conf/bin/idoptimize -g as super user. This gets the logs from the running kernel and analyzes them.
Issue the comand /etc/conf/bin/idoptimize -L as super user. This rebuilds an optimized kernel with logging turned off.
Shutdown Oracle and reboot the system.

Binding Shadow Processes

```
ps -fu oracle | grep -v grep | grep -i oracletpcc | awk '{print $2}'>
bind2
sort bind2 > bind3
paste bind.conf bind3 > bind_user_procs
sh bind_user_procs
rm bind2 bind3
```

bind.conf

```
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
```

```
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
pbind -b 1
pbind -b 2
pbind -b 3
pbind -b 4
pbind -b 5
pbind -b 6
pbind -b 7
pbind -b 0
```

Device Configurations

```
0:0,7,0: HBA     : (adsl,1) Adaptec PCI SCSI
  0,0,0: DISK    : SEAGATE ST34501W        0018
  0,1,0: DISK    : UNISYS  003665MAB3045SC 0608
  0,4,0: CDROM   : TOSHIBA CD-ROM XM-5701TA1557
  0,6,0: HBA     : ESG-SHV SCA HSBP M8     0.05
1:0,7,0: HBA     : (qlc2100,3) QLogic QLA2100
  0,0,0: DISK    : DGC     RAID 5          0412
  0,0,1: DISK    : DGC     RAID 5          0412
  0,0,2: DISK    : DGC     RAID 0          0412
  0,0,3: DISK    : DGC     RAID 0          0412
  0,0,4: DISK    : DGC     RAID 0          0412
  0,0,5: DISK    : DGC     RAID 0          0412
  0,0,6: DISK    : DGC     RAID 0          0412
  0,0,7: DISK    : DGC     RAID 0          0412
  0,0,8: DISK    : DGC     RAID 0          0412
  0,0,9: DISK    : DGC     RAID 0          0412
  0,0,10: DISK    : DGC     RAID 0          0412
  0,0,11: DISK    : DGC     RAID 0          0412
  0,0,12: DISK    : DGC     RAID 0          0412
  0,0,13: DISK    : DGC     RAID 0          0412
  0,0,14: DISK    : DGC                     0412
  0,0,15: DISK    : DGC                     0412
  0,1,0: DISK    : DGC     RAID 5          0412
```

```
  0,1,1: DISK    : DGC     RAID 5          0412
  0,1,2: DISK    : DGC     RAID 0          0412
  0,1,3: DISK    : DGC     RAID 0          0412
  0,1,4: DISK    : DGC     RAID 0          0412
  0,1,5: DISK    : DGC     RAID 0          0412
  0,1,6: DISK    : DGC     RAID 0          0412
  0,1,7: DISK    : DGC     RAID 0          0412
  0,1,8: DISK    : DGC     RAID 0          0412
  0,1,9: DISK    : DGC     RAID 0          0412
  0,1,10: DISK    : DGC     RAID 0          0412
  0,1,11: DISK    : DGC     RAID 0          0412
  0,1,12: DISK    : DGC     RAID 0          0412
  0,1,13: DISK    : DGC     RAID 0          0412
  0,1,14: DISK    : DGC                     0412
  0,1,15: DISK    : DGC                     0412
  1,7,0: HBA     : (qlc2100,3) QLogic QLA2100
  2,7,0: HBA     : (qlc2100,3) QLogic QLA2100
  3,7,0: HBA     : (qlc2100,3) QLogic QLA2100
  4,7,0: HBA     : (qlc2100,3) QLogic QLA2100
  5,7,0: HBA     : (qlc2100,3) QLogic QLA2100
  6,7,0: HBA     : (qlc2100,3) QLogic QLA2100
  7,7,0: HBA     : (qlc2100,3) QLogic QLA2100
2:0,7,0: HBA     : (qlc2100,1) QLogic QLA2100
  0,0,0: DISK    : DGC     RAID 5          0412
  0,0,1: DISK    : DGC     RAID 5          0412
  0,0,2: DISK    : DGC     RAID 0          0412
  0,0,3: DISK    : DGC     RAID 0          0412
  0,0,4: DISK    : DGC     RAID 0          0412
  0,0,5: DISK    : DGC     RAID 0          0412
  0,0,6: DISK    : DGC     RAID 0          0412
  0,0,7: DISK    : DGC     RAID 0          0412
  0,0,8: DISK    : DGC     RAID 0          0412
  0,0,9: DISK    : DGC     RAID 0          0412
  0,0,10: DISK    : DGC     RAID 0          0412
  0,0,11: DISK    : DGC     RAID 0          0412
  0,0,12: DISK    : DGC     RAID 0          0412
  0,0,13: DISK    : DGC     RAID 0          0412
  0,0,14: DISK    : DGC                     0412
  0,0,15: DISK    : DGC                     0412
  0,1,0: DISK    : DGC     RAID 5          0412
  0,1,1: DISK    : DGC     RAID 5          0412
  0,1,2: DISK    : DGC     RAID 0          0412
  0,1,3: DISK    : DGC     RAID 0          0412
  0,1,4: DISK    : DGC     RAID 0          0412
  0,1,5: DISK    : DGC     RAID 0          0412
  0,1,6: DISK    : DGC     RAID 0          0412
  0,1,7: DISK    : DGC     RAID 0          0412
  0,1,8: DISK    : DGC     RAID 0          0412
  0,1,9: DISK    : DGC     RAID 0          0412
  0,1,10: DISK    : DGC     RAID 0          0412
  0,1,11: DISK    : DGC     RAID 0          0412
  0,1,12: DISK    : DGC     RAID 0          0412
  0,1,13: DISK    : DGC     RAID 0          0412
  0,1,14: DISK    : DGC                     0412
  0,1,15: DISK    : DGC                     0412
  1,7,0: HBA     : (qlc2100,1) QLogic QLA2100
  2,7,0: HBA     : (qlc2100,1) QLogic QLA2100
  3,7,0: HBA     : (qlc2100,1) QLogic QLA2100
  4,7,0: HBA     : (qlc2100,1) QLogic QLA2100
  5,7,0: HBA     : (qlc2100,1) QLogic QLA2100
  6,7,0: HBA     : (qlc2100,1) QLogic QLA2100
  7,7,0: HBA     : (qlc2100,1) QLogic QLA2100
3:0,7,0: HBA     : (qlc2100,4) QLogic QLA2100
  0,0,0: DISK    : DGC     RAID 5          0412
  0,0,1: DISK    : DGC     RAID 0          0412
  0,0,2: DISK    : DGC     RAID 0          0412
  0,0,3: DISK    : DGC     RAID 0          0412
  0,0,4: DISK    : DGC     RAID 0          0412
  0,0,5: DISK    : DGC     RAID 0          0412
  0,0,6: DISK    : DGC     RAID 0          0412
  0,0,7: DISK    : DGC     RAID 0          0412
  0,0,8: DISK    : DGC     RAID 0          0412
  0,0,9: DISK    : DGC     RAID 0          0412
  0,0,10: DISK    : DGC     RAID 0          0412
  0,0,11: DISK    : DGC     RAID 0          0412
  0,0,12: DISK    : DGC     RAID 0          0412
  0,0,13: DISK    : DGC     RAID 0          0412
  0,0,14: DISK    : DGC                     0412
  0,0,15: DISK    : DGC                     0412
  0,1,0: DISK    : DGC     RAID 5          0412
  0,1,1: DISK    : DGC     RAID 0          0412
  0,1,2: DISK    : DGC     RAID 0          0412
  0,1,3: DISK    : DGC     RAID 0          0412
  0,1,4: DISK    : DGC     RAID 0          0412
  0,1,5: DISK    : DGC     RAID 0          0412
  0,1,6: DISK    : DGC     RAID 0          0412
  0,1,7: DISK    : DGC     RAID 0          0412
  0,1,8: DISK    : DGC     RAID 0          0412
  0,1,9: DISK    : DGC     RAID 0          0412
  0,1,10: DISK    : DGC     RAID 0          0412
  0,1,11: DISK    : DGC     RAID 0          0412
  0,1,12: DISK    : DGC     RAID 0          0412
  0,1,13: DISK    : DGC     RAID 0          0412
  0,1,14: DISK    : DGC                     0412
  0,1,15: DISK    : DGC                     0412
  1,7,0: HBA     : (qlc2100,4) QLogic QLA2100
  2,7,0: HBA     : (qlc2100,4) QLogic QLA2100
  3,7,0: HBA     : (qlc2100,4) QLogic QLA2100
  4,7,0: HBA     : (qlc2100,4) QLogic QLA2100
  5,7,0: HBA     : (qlc2100,4) QLogic QLA2100
  6,7,0: HBA     : (qlc2100,4) QLogic QLA2100
  7,7,0: HBA     : (qlc2100,4) QLogic QLA2100
4:0,7,0: HBA     : (qlc2100,5) QLogic QLA2100
  0,0,0: DISK    : DGC     RAID 5          0412
  0,0,1: DISK    : DGC     RAID 5          0412
  0,0,2: DISK    : DGC     RAID 0          0412
  0,0,3: DISK    : DGC     RAID 0          0412
  0,0,4: DISK    : DGC     RAID 0          0412
  0,0,5: DISK    : DGC     RAID 0          0412
  0,0,6: DISK    : DGC     RAID 0          0412
  0,0,7: DISK    : DGC     RAID 0          0412
  0,0,8: DISK    : DGC     RAID 0          0412
  0,0,9: DISK    : DGC     RAID 0          0412
  0,0,10: DISK    : DGC     RAID 0          0412
  0,0,11: DISK    : DGC     RAID 0          0412
  0,0,12: DISK    : DGC     RAID 0          0412
  0,0,13: DISK    : DGC     RAID 0          0412
  0,0,14: DISK    : DGC                     0412
```

```
  0,0,15: DISK    : DGC                    0412
  0,1,0: DISK    : DGC     RAID 5          0412
  0,1,1: DISK    : DGC     RAID 5          0412
  0,1,2: DISK    : DGC     RAID 0          0412
  0,1,3: DISK    : DGC     RAID 0          0412
  0,1,4: DISK    : DGC     RAID 0          0412
  0,1,5: DISK    : DGC     RAID 0          0412
  0,1,6: DISK    : DGC     RAID 0          0412
  0,1,7: DISK    : DGC     RAID 0          0412
  0,1,8: DISK    : DGC     RAID 0          0412
  0,1,9: DISK    : DGC     RAID 0          0412
  0,1,10: DISK    : DGC     RAID 0          0412
  0,1,11: DISK    : DGC     RAID 0          0412
  0,1,12: DISK    : DGC     RAID 0          0412
  0,1,13: DISK    : DGC     RAID 0          0412
  0,1,14: DISK    : DGC                     0412
  0,1,15: DISK    : DGC                     0412
  1,7,0: HBA     : (qlc2100,5) QLogic QLA2100
  2,7,0: HBA     : (qlc2100,5) QLogic QLA2100
  3,7,0: HBA     : (qlc2100,5) QLogic QLA2100
  4,7,0: HBA     : (qlc2100,5) QLogic QLA2100
  5,7,0: HBA     : (qlc2100,5) QLogic QLA2100
  6,7,0: HBA     : (qlc2100,5) QLogic QLA2100
  7,7,0: HBA     : (qlc2100,5) QLogic QLA2100
5:0,7,0: HBA     : (qlc2100,6) QLogic QLA2100
  0,0,0: DISK    : DGC     RAID 5          0412
  0,0,1: DISK    : DGC     RAID 5          0412
  0,0,2: DISK    : DGC     RAID 0          0412
  0,0,3: DISK    : DGC     RAID 0          0412
  0,0,4: DISK    : DGC     RAID 0          0412
  0,0,5: DISK    : DGC     RAID 0          0412
  0,0,6: DISK    : DGC     RAID 0          0412
  0,0,7: DISK    : DGC     RAID 0          0412
  0,0,8: DISK    : DGC     RAID 0          0412
  0,0,9: DISK    : DGC     RAID 0          0412
  0,0,10: DISK    : DGC     RAID 0          0412
  0,0,11: DISK    : DGC     RAID 0          0412
  0,0,12: DISK    : DGC     RAID 0          0412
  0,0,13: DISK    : DGC     RAID 0          0412
  0,0,14: DISK    : DGC                     0412
  0,0,15: DISK    : DGC                     0412
  0,1,0: DISK    : DGC     RAID 5          0412
  0,1,1: DISK    : DGC     RAID 5          0412
  0,1,2: DISK    : DGC     RAID 0          0412
  0,1,3: DISK    : DGC     RAID 0          0412
  0,1,4: DISK    : DGC     RAID 0          0412
  0,1,5: DISK    : DGC     RAID 0          0412
  0,1,6: DISK    : DGC     RAID 0          0412
  0,1,7: DISK    : DGC     RAID 0          0412
  0,1,8: DISK    : DGC     RAID 0          0412
  0,1,9: DISK    : DGC     RAID 0          0412
  0,1,10: DISK    : DGC     RAID 0          0412
  0,1,11: DISK    : DGC     RAID 0          0412
  0,1,12: DISK    : DGC     RAID 0          0412
  0,1,13: DISK    : DGC     RAID 0          0412
  0,1,14: DISK    : DGC                     0412
  0,1,15: DISK    : DGC                     0412
  1,7,0: HBA     : (qlc2100,6) QLogic QLA2100
  2,7,0: HBA     : (qlc2100,6) QLogic QLA2100
  3,7,0: HBA     : (qlc2100,6) QLogic QLA2100
  4,7,0: HBA     : (qlc2100,6) QLogic QLA2100
  5,7,0: HBA     : (qlc2100,6) QLogic QLA2100
  6,7,0: HBA     : (qlc2100,6) QLogic QLA2100
  7,7,0: HBA     : (qlc2100,6) QLogic QLA2100
6:0,7,0: HBA     : (qlc2100,7) QLogic QLA2100
  0,0,0: DISK    : DGC     RAID 10         0412
  0,0,1: DISK    : DGC     RAID 10         0412
  0,0,2: DISK    : DGC     RAID 10         0412
  0,0,3: DISK    : DGC     RAID 10         0412
  0,0,4: DISK    : DGC                     0412
  0,0,5: DISK    : DGC                     0412
  0,0,6: DISK    : DGC                     0412
  0,0,7: DISK    : DGC                     0412
  0,0,8: DISK    : DGC                     0412
  0,0,9: DISK    : DGC                     0412
  0,0,10: DISK    : DGC                     0412
  0,0,11: DISK    : DGC                     0412
  0,0,12: DISK    : DGC                     0412
  0,0,13: DISK    : DGC                     0412
  0,0,14: DISK    : DGC                     0412
  0,0,15: DISK    : DGC                     0412
  0,1,0: DISK    : DGC     RAID 10         0412
  0,1,1: DISK    : DGC     RAID 10         0412
  0,1,2: DISK    : DGC     RAID 10         0412
  0,1,3: DISK    : DGC     RAID 10         0412
  0,1,4: DISK    : DGC                     0412
  0,1,5: DISK    : DGC                     0412
  0,1,6: DISK    : DGC                     0412
  0,1,7: DISK    : DGC                     0412
  0,1,8: DISK    : DGC                     0412
  0,1,9: DISK    : DGC                     0412
  0,1,10: DISK    : DGC                     0412
  0,1,11: DISK    : DGC                     0412
  0,1,12: DISK    : DGC                     0412
  0,1,13: DISK    : DGC                     0412
  0,1,14: DISK    : DGC                     0412
  0,1,15: DISK    : DGC                     0412
  1,7,0: HBA     : (qlc2100,7) QLogic QLA2100
  2,7,0: HBA     : (qlc2100,7) QLogic QLA2100
  3,7,0: HBA     : (qlc2100,7) QLogic QLA2100
  4,7,0: HBA     : (qlc2100,7) QLogic QLA2100
  5,7,0: HBA     : (qlc2100,7) QLogic QLA2100
  6,7,0: HBA     : (qlc2100,7) QLogic QLA2100
  7,7,0: HBA     : (qlc2100,7) QLogic QLA2100
```

ESM7700 Disk Subsystem Configuration Information

Note: A 100MB write cache was enabled on all Database LUNs with Low Watermark set to 5 & High Watermark set to 6 and a 200MB write cache was enabled on all Log LUNs with Low Watermark set to 5 & High Watermark set to 8. No LUN read caching was enabled. Total cache size per SP was 512MB.

```
 *
 *******      SCSI3:0:1:6  LUN 0
Enclosure 0 Disk 0  Enabled
Enclosure 0 Disk 1  Enabled
Enclosure 0 Disk 2  Enabled
Enclosure 0 Disk 3  Enabled
Enclosure 0 Disk 4  Enabled
Enclosure 0 Disk 5  Enabled
Enclosure 0 Disk 6  Enabled
Enclosure 0 Disk 7  Enabled
Enclosure 0 Disk 8  Enabled
Enclosure 0 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID5
Write cache:                DISABLED
Read cache:                 DISABLED
Lun Capacity:               152426
 *
 *******      SCSI3:0:1:6  LUN 2
Enclosure 2 Disk 0  Enabled
Enclosure 2 Disk 1  Enabled
Enclosure 2 Disk 2  Enabled
Enclosure 2 Disk 3  Enabled
Enclosure 2 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI3:0:1:6  LUN 3
Enclosure 2 Disk 5  Enabled
Enclosure 2 Disk 6  Enabled
Enclosure 2 Disk 7  Enabled
Enclosure 2 Disk 8  Enabled
Enclosure 2 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI3:0:1:6  LUN 4
Enclosure 3 Disk 0  Enabled
Enclosure 3 Disk 1  Enabled
Enclosure 3 Disk 2  Enabled
Enclosure 3 Disk 3  Enabled
Enclosure 3 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI3:0:1:6  LUN 5
Enclosure 3 Disk 5  Enabled
Enclosure 3 Disk 6  Enabled
Enclosure 3 Disk 7  Enabled
Enclosure 3 Disk 8  Enabled
Enclosure 3 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI3:0:1:6  LUN 6
Enclosure 4 Disk 0  Enabled
Enclosure 4 Disk 1  Enabled
Enclosure 4 Disk 2  Enabled
Enclosure 4 Disk 3  Enabled
Enclosure 4 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI3:0:1:6  LUN 7
Enclosure 4 Disk 5  Enabled
Enclosure 4 Disk 6  Enabled
Enclosure 4 Disk 7  Enabled
Enclosure 4 Disk 8  Enabled
Enclosure 4 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI3:0:3:6  LUN 1
Enclosure 1 Disk 0  Enabled
Enclosure 1 Disk 1  Enabled
Enclosure 1 Disk 2  Enabled
Enclosure 1 Disk 3  Enabled
Enclosure 1 Disk 4  Enabled
Enclosure 1 Disk 5  Enabled
Enclosure 1 Disk 6  Enabled
Enclosure 1 Disk 7  Enabled
Enclosure 1 Disk 8  Enabled
Enclosure 1 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID5
Write cache:                DISABLED
Read cache:                 DISABLED
Lun Capacity:               152426
 *
 *******      SCSI3:0:3:6  LUN 8
Enclosure 5 Disk 0  Enabled
Enclosure 5 Disk 1  Enabled
Enclosure 5 Disk 2  Enabled
Enclosure 5 Disk 3  Enabled
Enclosure 5 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
```

```
Lun Capacity:               84681
 *
 *******      SCSI3:0:3:6  LUN 9
Enclosure 5 Disk 5  Enabled
Enclosure 5 Disk 6  Enabled
Enclosure 5 Disk 7  Enabled
Enclosure 5 Disk 8  Enabled
Enclosure 5 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI3:0:3:6  LUN 10
Enclosure 6 Disk 0  Enabled
Enclosure 6 Disk 1  Enabled
Enclosure 6 Disk 2  Enabled
Enclosure 6 Disk 3  Enabled
Enclosure 6 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI3:0:3:6  LUN 11
Enclosure 6 Disk 5  Enabled
Enclosure 6 Disk 6  Enabled
Enclosure 6 Disk 7  Enabled
Enclosure 6 Disk 8  Enabled
Enclosure 6 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI3:0:3:6  LUN 12
Enclosure 7 Disk 0  Enabled
Enclosure 7 Disk 1  Enabled
Enclosure 7 Disk 2  Enabled
Enclosure 7 Disk 3  Enabled
Enclosure 7 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI3:0:3:6  LUN 13
Enclosure 7 Disk 5  Enabled
Enclosure 7 Disk 6  Enabled
Enclosure 7 Disk 7  Enabled
Enclosure 7 Disk 8  Enabled
Enclosure 7 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *
 *******      SCSI4:0:1:6  LUN 0
Enclosure 0 Disk 0  Enabled
Enclosure 0 Disk 1  Enabled
Enclosure 0 Disk 2  Enabled
Enclosure 0 Disk 3  Enabled
Enclosure 0 Disk 4  Enabled
Enclosure 0 Disk 5  Enabled
Enclosure 0 Disk 6  Enabled
Enclosure 0 Disk 7  Enabled
Enclosure 0 Disk 8  Enabled
Enclosure 0 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID5
Write cache:                DISABLED
Read cache:                 DISABLED
Lun Capacity:               152426
 *
 *******      SCSI4:0:1:6  LUN 2
Enclosure 2 Disk 0  Enabled
Enclosure 2 Disk 1  Enabled
Enclosure 2 Disk 2  Enabled
Enclosure 2 Disk 3  Enabled
Enclosure 2 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI4:0:1:6  LUN 3
Enclosure 2 Disk 5  Enabled
Enclosure 2 Disk 6  Enabled
Enclosure 2 Disk 7  Enabled
Enclosure 2 Disk 8  Enabled
Enclosure 2 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI4:0:1:6  LUN 4
Enclosure 3 Disk 0  Enabled
Enclosure 3 Disk 1  Enabled
Enclosure 3 Disk 2  Enabled
Enclosure 3 Disk 3  Enabled
Enclosure 3 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI4:0:1:6  LUN 5
Enclosure 3 Disk 5  Enabled
```

```
Enclosure 3 Disk 6  Enabled
Enclosure 3 Disk 7  Enabled
Enclosure 3 Disk 8  Enabled
Enclosure 3 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI4:0:1:6  LUN 6
Enclosure 4 Disk 0  Enabled
Enclosure 4 Disk 1  Enabled
Enclosure 4 Disk 2  Enabled
Enclosure 4 Disk 3  Enabled
Enclosure 4 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI4:0:1:6  LUN 7
Enclosure 4 Disk 5  Enabled
Enclosure 4 Disk 6  Enabled
Enclosure 4 Disk 7  Enabled
Enclosure 4 Disk 8  Enabled
Enclosure 4 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI4:0:2:0  LUN 1
Enclosure 1 Disk 0  Enabled
Enclosure 1 Disk 1  Enabled
Enclosure 1 Disk 2  Enabled
Enclosure 1 Disk 3  Enabled
Enclosure 1 Disk 4  Enabled
Enclosure 1 Disk 5  Enabled
Enclosure 1 Disk 6  Enabled
Enclosure 1 Disk 7  Enabled
Enclosure 1 Disk 8  Enabled
Enclosure 1 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID5
Write cache:                DISABLED
Read cache:                 DISABLED
Lun Capacity:               152426
 *
 *******      SCSI4:0:2:0  LUN 8
Enclosure 5 Disk 0  Enabled
Enclosure 5 Disk 1  Enabled
Enclosure 5 Disk 2  Enabled
Enclosure 5 Disk 3  Enabled
Enclosure 5 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI4:0:2:0  LUN 9
Enclosure 5 Disk 5  Enabled
Enclosure 5 Disk 6  Enabled
Enclosure 5 Disk 7  Enabled
Enclosure 5 Disk 8  Enabled
Enclosure 5 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI4:0:2:0  LUN 10
Enclosure 6 Disk 0  Enabled
Enclosure 6 Disk 1  Enabled
Enclosure 6 Disk 2  Enabled
Enclosure 6 Disk 3  Enabled
Enclosure 6 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI4:0:2:0  LUN 11
Enclosure 6 Disk 5  Enabled
Enclosure 6 Disk 6  Enabled
Enclosure 6 Disk 7  Enabled
Enclosure 6 Disk 8  Enabled
Enclosure 6 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI4:0:2:0  LUN 12
Enclosure 7 Disk 0  Enabled
Enclosure 7 Disk 1  Enabled
Enclosure 7 Disk 2  Enabled
Enclosure 7 Disk 3  Enabled
Enclosure 7 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI4:0:2:0  LUN 13
Enclosure 7 Disk 5  Enabled
Enclosure 7 Disk 6  Enabled
Enclosure 7 Disk 7  Enabled
Enclosure 7 Disk 8  Enabled
Enclosure 7 Disk 9  Enabled
Element Size:               128
```

```
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *
 *******      SCSI5:0:0:0  LUN 0
Enclosure 0 Disk 0  Enabled
Enclosure 0 Disk 1  Enabled
Enclosure 0 Disk 2  Enabled
Enclosure 0 Disk 3  Enabled
Enclosure 0 Disk 4  Enabled
Enclosure 0 Disk 5  Enabled
Enclosure 0 Disk 6  Enabled
Enclosure 0 Disk 7  Enabled
Enclosure 0 Disk 8  Enabled
Enclosure 0 Disk 9  Enabled
Element Size:               256
RAID Type:                  RAID5
Write cache:                DISABLED
Read cache:                 DISABLED
Lun Capacity:               152425
 *
 *******      SCSI5:0:0:0  LUN 2
Enclosure 2 Disk 2  Enabled
Enclosure 2 Disk 3  Enabled
Enclosure 2 Disk 4  Enabled
Enclosure 2 Disk 5  Enabled
Enclosure 2 Disk 6  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI5:0:0:0  LUN 3
Enclosure 2 Disk 0  Enabled
Enclosure 2 Disk 1  Enabled
Enclosure 2 Disk 7  Enabled
Enclosure 2 Disk 8  Enabled
Enclosure 2 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI5:0:0:0  LUN 4
Enclosure 3 Disk 0  Enabled
Enclosure 3 Disk 1  Enabled
Enclosure 3 Disk 2  Enabled
Enclosure 3 Disk 3  Enabled
Enclosure 3 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI5:0:0:0  LUN 5
Enclosure 3 Disk 5  Enabled
Enclosure 3 Disk 6  Enabled
Enclosure 3 Disk 7  Enabled
Enclosure 3 Disk 8  Enabled
Enclosure 3 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI5:0:0:0  LUN 6
Enclosure 4 Disk 0  Enabled
Enclosure 4 Disk 1  Enabled
Enclosure 4 Disk 2  Enabled
Enclosure 4 Disk 3  Enabled
Enclosure 4 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI5:0:0:0  LUN 7
Enclosure 4 Disk 5  Enabled
Enclosure 4 Disk 6  Enabled
Enclosure 4 Disk 7  Enabled
Enclosure 4 Disk 8  Enabled
Enclosure 4 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI5:0:2:0  LUN 1
Enclosure 1 Disk 0  Enabled
Enclosure 1 Disk 1  Enabled
Enclosure 1 Disk 2  Enabled
Enclosure 1 Disk 3  Enabled
Enclosure 1 Disk 4  Enabled
Enclosure 1 Disk 5  Enabled
Enclosure 1 Disk 6  Enabled
Enclosure 1 Disk 7  Enabled
Enclosure 1 Disk 8  Enabled
Enclosure 1 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               169362
 *
 *******      SCSI5:0:2:0  LUN 8
Enclosure 5 Disk 0  Enabled
Enclosure 5 Disk 1  Enabled
Enclosure 5 Disk 2  Enabled
Enclosure 5 Disk 3  Enabled
Enclosure 5 Disk 4  Enabled
```

```
Element Size:                128
RAID Type:                   RAID0
Write cache:                 ENABLED
Read cache:                  DISABLED
Lun Capacity:                84681
 *
 *******      SCSI5:0:2:0  LUN 9
Enclosure 5 Disk 5  Enabled
Enclosure 5 Disk 6  Enabled
Enclosure 5 Disk 7  Enabled
Enclosure 5 Disk 8  Enabled
Enclosure 5 Disk 9  Enabled
Element Size:                128
RAID Type:                   RAID0
Write cache:                 ENABLED
Read cache:                  DISABLED
Lun Capacity:                84681
 *
 *******      SCSI5:0:2:0  LUN 10
Enclosure 6 Disk 0  Enabled
Enclosure 6 Disk 1  Enabled
Enclosure 6 Disk 2  Enabled
Enclosure 6 Disk 3  Enabled
Enclosure 6 Disk 4  Enabled
Element Size:                128
RAID Type:                   RAID0
Write cache:                 ENABLED
Read cache:                  DISABLED
Lun Capacity:                84681
 *
 *******      SCSI5:0:2:0  LUN 11
Enclosure 6 Disk 5  Enabled
Enclosure 6 Disk 6  Enabled
Enclosure 6 Disk 7  Enabled
Enclosure 6 Disk 8  Enabled
Enclosure 6 Disk 9  Enabled
Element Size:                128
RAID Type:                   RAID0
Write cache:                 ENABLED
Read cache:                  DISABLED
Lun Capacity:                84681
 *
 *******      SCSI5:0:2:0  LUN 12
Enclosure 7 Disk 0  Enabled
Enclosure 7 Disk 1  Enabled
Enclosure 7 Disk 2  Enabled
Enclosure 7 Disk 3  Enabled
Enclosure 7 Disk 4  Enabled
Element Size:                128
RAID Type:                   RAID0
Write cache:                 ENABLED
Read cache:                  DISABLED
Lun Capacity:                84681
 *
 *******      SCSI5:0:2:0  LUN 13
Enclosure 7 Disk 5  Enabled
Enclosure 7 Disk 6  Enabled
Enclosure 7 Disk 7  Enabled
Enclosure 7 Disk 8  Enabled
Enclosure 7 Disk 9  Enabled
Element Size:                128
RAID Type:                   RAID0
Write cache:                 ENABLED
Read cache:                  DISABLED
Lun Capacity:                84681
 *
 *
 *******      SCSI6:0:0:0  LUN 0
Enclosure 0 Disk 0  Enabled
Enclosure 0 Disk 1  Enabled
Enclosure 0 Disk 2  Enabled
Enclosure 0 Disk 3  Enabled
Enclosure 0 Disk 4  Enabled
Enclosure 0 Disk 5  Enabled
Enclosure 0 Disk 6  Enabled
Enclosure 0 Disk 7  Enabled
Enclosure 0 Disk 8  Enabled
Enclosure 0 Disk 9  Enabled
Element Size:                128
RAID Type:                   RAID5
Write cache:                 DISABLED
Read cache:                  DISABLED
Lun Capacity:                152426
 *
 *******      SCSI6:0:0:0  LUN 2
Enclosure 2 Disk 0  Enabled
Enclosure 2 Disk 1  Enabled
Enclosure 2 Disk 2  Enabled
Enclosure 2 Disk 3  Enabled
Enclosure 2 Disk 4  Enabled
Element Size:                128
RAID Type:                   RAID0
Write cache:                 ENABLED
Read cache:                  DISABLED
Lun Capacity:                84681
 *
 *******      SCSI6:0:0:0  LUN 3
Enclosure 2 Disk 5  Enabled
Enclosure 2 Disk 6  Enabled
Enclosure 2 Disk 7  Enabled
Enclosure 2 Disk 8  Enabled
Enclosure 2 Disk 9  Enabled
Element Size:                128
RAID Type:                   RAID0
Write cache:                 ENABLED
Read cache:                  DISABLED
Lun Capacity:                84681
 *
 *******      SCSI6:0:0:0  LUN 4
Enclosure 3 Disk 0  Enabled
Enclosure 3 Disk 1  Enabled
Enclosure 3 Disk 2  Enabled
Enclosure 3 Disk 3  Enabled
Enclosure 3 Disk 4  Enabled
Element Size:                128
RAID Type:                   RAID0
Write cache:                 ENABLED
Read cache:                  DISABLED
```

```
Lun Capacity:               84681
 *
 *******      SCSI6:0:0:0  LUN 5
Enclosure 3 Disk 5  Enabled
Enclosure 3 Disk 6  Enabled
Enclosure 3 Disk 7  Enabled
Enclosure 3 Disk 8  Enabled
Enclosure 3 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI6:0:0:0  LUN 6
Enclosure 4 Disk 0  Enabled
Enclosure 4 Disk 1  Enabled
Enclosure 4 Disk 2  Enabled
Enclosure 4 Disk 3  Enabled
Enclosure 4 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI6:0:0:0  LUN 7
Enclosure 4 Disk 5  Enabled
Enclosure 4 Disk 6  Enabled
Enclosure 4 Disk 7  Enabled
Enclosure 4 Disk 8  Enabled
Enclosure 4 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI6:0:2:0  LUN 1
Enclosure 1 Disk 0  Enabled
Enclosure 1 Disk 1  Enabled
Enclosure 1 Disk 2  Enabled
Enclosure 1 Disk 3  Enabled
Enclosure 1 Disk 4  Enabled
Enclosure 1 Disk 5  Enabled
Enclosure 1 Disk 6  Enabled
Enclosure 1 Disk 7  Enabled
Enclosure 1 Disk 8  Enabled
Enclosure 1 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID5
Write cache:                DISABLED
Read cache:                 DISABLED
Lun Capacity:               152426
 *
 *******      SCSI6:0:2:0  LUN 8
Enclosure 5 Disk 0  Enabled
Enclosure 5 Disk 1  Enabled
Enclosure 5 Disk 2  Enabled
Enclosure 5 Disk 3  Enabled
Enclosure 5 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI6:0:2:0  LUN 9
Enclosure 5 Disk 5  Enabled
Enclosure 5 Disk 6  Enabled
Enclosure 5 Disk 7  Enabled
Enclosure 5 Disk 8  Enabled
Enclosure 5 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI6:0:2:0  LUN 10
Enclosure 6 Disk 0  Enabled
Enclosure 6 Disk 1  Enabled
Enclosure 6 Disk 2  Enabled
Enclosure 6 Disk 3  Enabled
Enclosure 6 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI6:0:2:0  LUN 11
Enclosure 6 Disk 5  Enabled
Enclosure 6 Disk 6  Enabled
Enclosure 6 Disk 7  Enabled
Enclosure 6 Disk 8  Enabled
Enclosure 6 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI6:0:2:0  LUN 12
Enclosure 7 Disk 0  Enabled
Enclosure 7 Disk 1  Enabled
Enclosure 7 Disk 2  Enabled
Enclosure 7 Disk 3  Enabled
Enclosure 7 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI6:0:2:0  LUN 13
Enclosure 7 Disk 5  Enabled
Enclosure 7 Disk 6  Enabled
```

```
Enclosure 7 Disk 7  Enabled
Enclosure 7 Disk 8  Enabled
Enclosure 7 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI7:0:0:0  LUN 0
Enclosure 0 Disk 0  Enabled
Enclosure 0 Disk 1  Enabled
Enclosure 0 Disk 2  Enabled
Enclosure 0 Disk 3  Enabled
Enclosure 0 Disk 4  Enabled
Enclosure 0 Disk 5  Enabled
Enclosure 0 Disk 6  Enabled
Enclosure 0 Disk 7  Enabled
Enclosure 0 Disk 8  Enabled
Enclosure 0 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID5
Write cache:                DISABLED
Read cache:                 DISABLED
Lun Capacity:               152426
 *
 *******      SCSI7:0:0:0  LUN 2
Enclosure 2 Disk 0  Enabled
Enclosure 2 Disk 1  Enabled
Enclosure 2 Disk 2  Enabled
Enclosure 2 Disk 3  Enabled
Enclosure 2 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI7:0:0:0  LUN 3
Enclosure 2 Disk 5  Enabled
Enclosure 2 Disk 6  Enabled
Enclosure 2 Disk 7  Enabled
Enclosure 2 Disk 8  Enabled
Enclosure 2 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI7:0:0:0  LUN 4
Enclosure 3 Disk 0  Enabled
Enclosure 3 Disk 1  Enabled
Enclosure 3 Disk 2  Enabled
Enclosure 3 Disk 3  Enabled
Enclosure 3 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI7:0:0:0  LUN 5
Enclosure 3 Disk 5  Enabled
Enclosure 3 Disk 6  Enabled
Enclosure 3 Disk 7  Enabled
Enclosure 3 Disk 8  Enabled
Enclosure 3 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI7:0:0:0  LUN 6
Enclosure 4 Disk 0  Enabled
Enclosure 4 Disk 1  Enabled
Enclosure 4 Disk 2  Enabled
Enclosure 4 Disk 3  Enabled
Enclosure 4 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI7:0:0:0  LUN 7
Enclosure 4 Disk 5  Enabled
Enclosure 4 Disk 6  Enabled
Enclosure 4 Disk 7  Enabled
Enclosure 4 Disk 8  Enabled
Enclosure 4 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI7:0:2:0  LUN 1
Enclosure 1 Disk 0  Enabled
Enclosure 1 Disk 1  Enabled
Enclosure 1 Disk 2  Enabled
Enclosure 1 Disk 3  Enabled
Enclosure 1 Disk 4  Enabled
Enclosure 1 Disk 5  Enabled
Enclosure 1 Disk 6  Enabled
Enclosure 1 Disk 7  Enabled
Enclosure 1 Disk 8  Enabled
Enclosure 1 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID5
Write cache:                DISABLED
Read cache:                 DISABLED
Lun Capacity:               152426
 *
 *******      SCSI7:0:2:0  LUN 8
Enclosure 5 Disk 0  Enabled
Enclosure 5 Disk 1  Enabled
```

```
Enclosure 5 Disk 2  Enabled
Enclosure 5 Disk 3  Enabled
Enclosure 5 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI7:0:2:0  LUN 9
Enclosure 5 Disk 5  Enabled
Enclosure 5 Disk 6  Enabled
Enclosure 5 Disk 7  Enabled
Enclosure 5 Disk 8  Enabled
Enclosure 5 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI7:0:2:0  LUN 10
Enclosure 6 Disk 0  Enabled
Enclosure 6 Disk 1  Enabled
Enclosure 6 Disk 2  Enabled
Enclosure 6 Disk 3  Enabled
Enclosure 6 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI7:0:2:0  LUN 11
Enclosure 6 Disk 5  Enabled
Enclosure 6 Disk 6  Enabled
Enclosure 6 Disk 7  Enabled
Enclosure 6 Disk 8  Enabled
Enclosure 6 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI7:0:2:0  LUN 12
Enclosure 7 Disk 0  Enabled
Enclosure 7 Disk 1  Enabled
Enclosure 7 Disk 2  Enabled
Enclosure 7 Disk 3  Enabled
Enclosure 7 Disk 4  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI7:0:2:0  LUN 13
Enclosure 7 Disk 5  Enabled
Enclosure 7 Disk 6  Enabled
Enclosure 7 Disk 7  Enabled
Enclosure 7 Disk 8  Enabled
Enclosure 7 Disk 9  Enabled
Element Size:               128
RAID Type:                  RAID0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               84681
 *
 *******      SCSI8:0:1:0  LUN 0
Enclosure 0 Disk 0  Enabled
Enclosure 0 Disk 2  Enabled
Enclosure 0 Disk 4  Enabled
Enclosure 0 Disk 6  Enabled
Enclosure 0 Disk 8  Enabled
Enclosure 0 Disk 1  Enabled
Enclosure 0 Disk 3  Enabled
Enclosure 0 Disk 5  Enabled
Enclosure 0 Disk 7  Enabled
Enclosure 0 Disk 9  Enabled
Element Size:               256
RAID Type:                  RAID1/0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               152425
 *
 *******      SCSI8:0:1:0  LUN 1
Enclosure 1 Disk 0  Enabled
Enclosure 1 Disk 2  Enabled
Enclosure 1 Disk 4  Enabled
Enclosure 1 Disk 6  Enabled
Enclosure 1 Disk 8  Enabled
Enclosure 1 Disk 1  Enabled
Enclosure 1 Disk 3  Enabled
Enclosure 1 Disk 5  Enabled
Enclosure 1 Disk 7  Enabled
Enclosure 1 Disk 9  Enabled
Element Size:               256
RAID Type:                  RAID1/0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               152425
 *
 *******      SCSI8:0:2:0  LUN 2
Enclosure 2 Disk 0  Enabled
Enclosure 2 Disk 2  Enabled
Enclosure 2 Disk 4  Enabled
Enclosure 2 Disk 6  Enabled
Enclosure 2 Disk 8  Enabled
Enclosure 2 Disk 1  Enabled
Enclosure 2 Disk 3  Enabled
Enclosure 2 Disk 5  Enabled
Enclosure 2 Disk 7  Enabled
Enclosure 2 Disk 9  Enabled
Element Size:               256
RAID Type:                  RAID1/0
Write cache:                ENABLED
Read cache:                 DISABLED
```

```
Lun Capacity:               152425
 *
 *******      SCSI8:0:2:0  LUN 3
Enclosure 6 Disk 0  Enabled
Enclosure 6 Disk 2  Enabled
Enclosure 6 Disk 4  Enabled
Enclosure 6 Disk 6  Enabled
Enclosure 6 Disk 8  Enabled
Enclosure 6 Disk 1  Enabled
Enclosure 6 Disk 3  Enabled
Enclosure 6 Disk 5  Enabled
Enclosure 6 Disk 7  Enabled
Enclosure 6 Disk 9  Enabled
Element Size:               256
RAID Type:                  RAID1/0
Write cache:                ENABLED
Read cache:                 DISABLED
Lun Capacity:               152425
 *
```

Qlogic Controller Configuration Information

```
;Document number: FC0161802-07  File name: FC010207.DAT
;Document Name: NVRAM DATA, QLA2100, ISP2100
;
;APPLICATION:   ISP2100 FIBRE CHANNEL
;
;
;NOTE:  A companion file with ".DEF" file extension must be present to
;       insure that file comparision during DVT testing will pass.
;
;Revision History:
;REV   DATE   WHO ECO#  DESCRIPTION OF CHANGE
;
;    06-11-99  BJR    Updated Doc # and File name to dash 07.
;    04-09-99  GH     Added host parameter bit 14 to enable database load
;    03-26-99  DKW    Added host parameter bit 13 to do extra read
;                           to flush bridge cache.
;    11-11-98  BR             Dash-06 reinstated,-05 voided by Doc Cont.

;    11-06-98  SCS      Roll back the Dash Number back to FC0161802-05.
;    11-03-98  BR       Dash number change, FC0161802-06.  File now ready
;                           for updates or modifications.
;    10-22-98  GH             Changed default parameter for port login
retry
;                       count to 30. Changed loop reset delay back to 5.
;    10-15-98  GH             Changed default parameter for port down
retry
;                           count and loop reset delay.
;    08-06-98  GH       Changed some default settings
;    05-20-98 SCS       Defined bits in F/W Option filed to match F/W
Spec.
;    04-27-98 DM        Added host parameter bit12 - enable database
storage
;
;    04-16-98 TWT       Add Login Timeout RISC parameter
;                       Add host parameter bit10 - enable LIP full login
;                       Add host parameter bit11 - enable Target Reset
;    04-02-98 TWT       Add maximum LUNs per target parameter
;
; B  02-19-98 TWT XXXX Add host parameter block bits:
;                      bit8 - enable 64bit addressing
;                      bit9 - enable LIP reset
;    02-20-98 GH        Changed location where subsystem vendor and
;                       device id are stored.
; A  06-17-97 scs XXXX  Initial release. (SCS)
;
; This file is the default settings for the NVRAM and must have the
extension
; of DAT.  Edit this file to change parameters.
;
;       Numbers are in decimal. File can also contain comments.
;
;***************************************************************************
*
;     !!!!!     DO NOT CHANGE THE ORDER OF THE DATA     !!!!!!!!!!!!
;***************************************************************************
*
;
; The NVRAM will be layed out in the same order as this file.

;***********************************
; NVRAM header

id                      ["4 characters"] = "ISP "
version                 [0-255] = 1
reserved                [0-255] = 0

;***********************************
; NVRAM RISC parameter block

version                 [0-255] = 1
reserved                [0-255] = 0

; Firmware options

enable hard loop ID     [0-1] = 0
enable fairness         [0-1] = 1
enable full duplex      [0-1] = 0
enable fast posting     [0-1] = 1
enable target mode      [0-1] = 0
disable initiator mode  [0-1] = 0
Enable ADISC            [0-1] = 0
Reserved                [0-1] = 0
Enable PDBC Notify      [0-1] = 0
Disable Initial LIP     [0-1] = 0
Descending Search LoopID[0-1] = 0
Reserved                [0-1] = 0
Reserved                [0-1] = 0
Full Login After LIP    [0-1] = 1
Reserved                [0-1] = 0
unused                  [0-1] = 1

frame payload size      [0-65535] = 1024
```

```
max iocb allocation     [0-65535] = 256
execution throttle      [0-65535] = 45
retry count             [0-255] = 30
retry delay             [0-255] = 1

node name 0 (must be zero)  [0-255] = 32
node name 1 (must be zero)  [0-255] = 0
node name 2 (company ID)    [0-255] = 0
node name 3 (company ID)    [0-255] = 224
node name 4 (company ID)    [0-255] = 139
node name 5 (serial number) [0-255] = 0
node name 6 (serial number) [0-255] = 86
node name 7 (serial number) [0-255] = 47

adapter hard loop ID    [0-65535] = 125
unused                  [0-255] = 0
login timeout           [0-255] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

;**************************************
; Reserved for expanded RISC parameter block

unused                  [0-65535] = 51
unused                  [0-65535] = 2560
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

;**************************************
; NVRAM host parameter block

unused                  [0-1] = 0
disable BIOS            [0-1] = 1
disable LUNs            [0-1] = 0
enable selectable boot  [0-1] = 0
disable RISC code load  [0-1] = 0
set cache line size 1   [0-1] = 0
PCI Parity Disable      [0-1] = 0
enable extended logging [0-1] = 0
enable 64bit addressing [0-1] = 1
enable LIP reset        [0-1] = 0
enable LIP full login   [0-1] = 1
enable target reset     [0-1] = 0
enable database storage [0-1] = 0
enable cache flush read [0-1] = 0
enable database load    [0-1] = 0
unused                  [0-1] = 0

boot node name 0        [0-255] = 0
boot node name 1        [0-255] = 0
boot node name 2        [0-255] = 0
boot node name 3        [0-255] = 0
boot node name 4        [0-255] = 0
boot node name 5        [0-255] = 0
boot node name 6        [0-255] = 0
boot node name 7        [0-255] = 0
boot LUN number         [0-255] = 0

reset delay             [0-255] = 5
port down retry count   [0-255] = 30
reserved1               [0-255] = 0

maximum LUNs per target [0-65535] = 16
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 100

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 150

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
```

```
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 200

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Subsystem vendor ID must be at offset 244

subsystem vendor ID     [0-65535] = 8430

unused                  [0-65535] = 0

; Subsystem device ID must be at offset 248

subsystem device ID     [0-65535] = 2

; Subsystem vendor ID for ISP2100a

subsystem vendor ID     [0-65535] = 4215

; Subsystem device ID for ISP2100a

subsystem device ID     [0-65535] = 1
unused                  [0-255] = 0
checksum                [0-255] = 91
;Document number: FC0161802-07  File name: FC010207.DAT
;Document Name: NVRAM DATA, QLA2100, ISP2100
;
;APPLICATION:   ISP2100 FIBRE CHANNEL
;
;
;NOTE:  A companion file with ".DEF" file extension must be present to
;       insure that file comparision during DVT testing will pass.
;
;Revision History:
;REV   DATE   WHO ECO#  DESCRIPTION OF CHANGE
;
;    06-11-99  BJR    Updated Doc # and File name to dash 07.
;    04-09-99  GH     Added host parameter bit 14 to enable database load
;    03-26-99  DKW    Added host parameter bit 13 to do extra read
;                            to flush bridge cache.
;    11-11-98  BR            Dash-06 reinstated,-05 voided by Doc Cont.

;    11-06-98  SCS      Roll back the Dash Number back to FC0161802-05.
;    11-03-98  BR       Dash number change, FC0161802-06.  File now ready
;                            for updates or modifications.
;    10-22-98  GH            Changed default parameter for port login
retry
;                       count to 30. Changed loop reset delay back to 5.
;    10-15-98  GH            Changed default parameter for port down
retry
;                            count and loop reset delay.
;    08-06-98  GH       Changed some default settings
;    05-20-98 SCS       Defined bits in F/W Option filed to match F/W
Spec.
;    04-27-98 DM        Added host parameter bit12 - enable database
storage
;
;    04-16-98 TWT       Add Login Timeout RISC parameter
;                       Add host parameter bit10 - enable LIP full login
;                       Add host parameter bit11 - enable Target Reset
;    04-02-98 TWT       Add maximum LUNs per target parameter
;
; B  02-19-98 TWT XXXX Add host parameter block bits:
;                      bit8 - enable 64bit addressing
;                      bit9 - enable LIP reset
;    02-20-98 GH        Changed location where subsystem vendor and
;                       device id are stored.
; A  06-17-97 scs XXXX  Initial release. (SCS)
;
; This file is the default settings for the NVRAM and must have the
extension
; of DAT.  Edit this file to change parameters.
;
;       Numbers are in decimal. File can also contain comments.
;
```

```
;***********************************************************************
*
;     !!!!!     DO NOT CHANGE THE ORDER OF THE DATA     !!!!!!!!!!!!
;***********************************************************************
*
;
; The NVRAM will be layed out in the same order as this file.

;***********************************
; NVRAM header

id                      ["4 characters"] = "ISP "
version                 [0-255] = 1
reserved                [0-255] = 0

;***********************************
; NVRAM RISC parameter block

version                 [0-255] = 1
reserved                [0-255] = 0

; Firmware options

enable hard loop ID     [0-1] = 0
enable fairness         [0-1] = 1
enable full duplex      [0-1] = 0
enable fast posting     [0-1] = 1
enable target mode      [0-1] = 0
disable initiator mode  [0-1] = 0
Enable ADISC            [0-1] = 0
Reserved                [0-1] = 0
Enable PDBC Notify      [0-1] = 0
Disable Initial LIP     [0-1] = 0
Descending Search LoopID[0-1] = 0
Reserved                [0-1] = 0
Reserved                [0-1] = 0
Full Login After LIP    [0-1] = 1
Reserved                [0-1] = 0
unused                  [0-1] = 1

frame payload size      [0-65535] = 1024
max iocb allocation     [0-65535] = 256
execution throttle      [0-65535] = 45
retry count             [0-255] = 30
retry delay             [0-255] = 1

node name 0 (must be zero)  [0-255] = 32
node name 1 (must be zero)  [0-255] = 0
node name 2 (company ID)    [0-255] = 0
node name 3 (company ID)    [0-255] = 224
node name 4 (company ID)    [0-255] = 139
node name 5 (serial number) [0-255] = 0
node name 6 (serial number) [0-255] = 8
node name 7 (serial number) [0-255] = 81

adapter hard loop ID    [0-65535] = 125
unused                  [0-255] = 0
login timeout           [0-255] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

;***********************************
; Reserved for expanded RISC parameter block

unused                  [0-65535] = 51
unused                  [0-65535] = 2560
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

;***********************************
; NVRAM host parameter block

unused                  [0-1] = 0
disable BIOS            [0-1] = 1
disable LUNs            [0-1] = 0
enable selectable boot  [0-1] = 0
disable RISC code load  [0-1] = 0
set cache line size 1   [0-1] = 0
PCI Parity Disable      [0-1] = 0
enable extended logging [0-1] = 0
enable 64bit addressing [0-1] = 1
enable LIP reset        [0-1] = 0
enable LIP full login   [0-1] = 1
enable target reset     [0-1] = 0
enable database storage [0-1] = 0
enable cache flush read [0-1] = 0
enable database load    [0-1] = 0
unused                  [0-1] = 0

boot node name 0        [0-255] = 0
boot node name 1        [0-255] = 0
boot node name 2        [0-255] = 0
boot node name 3        [0-255] = 0
boot node name 4        [0-255] = 0
boot node name 5        [0-255] = 0
boot node name 6        [0-255] = 0
boot node name 7        [0-255] = 0
boot LUN number         [0-255] = 0

reset delay             [0-255] = 5
port down retry count   [0-255] = 30
reserved1               [0-255] = 0

maximum LUNs per target [0-65535] = 16
```

```
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 100

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 150

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
```

```
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 200

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Subsystem vendor ID must be at offset 244

subsystem vendor ID     [0-65535] = 8430

unused                  [0-65535] = 0

; Subsystem device ID must be at offset 248

subsystem device ID     [0-65535] = 2

; Subsystem vendor ID for ISP2100a

subsystem vendor ID     [0-65535] = 4215

; Subsystem device ID for ISP2100a

subsystem device ID     [0-65535] = 1
unused                  [0-255] = 0
checksum                [0-255] = 135
;Document number: FC0161802-07  File name: FC010207.DAT
;Document Name: NVRAM DATA, QLA2100, ISP2100
;
;APPLICATION:   ISP2100 FIBRE CHANNEL
;
;
;NOTE:  A companion file with ".DEF" file extension must be present to
;       insure that file comparision during DVT testing will pass.
;
;Revision History:
```

```
;REV   DATE   WHO ECO#  DESCRIPTION OF CHANGE
;
;    06-11-99  BJR     Updated Doc # and File name to dash 07.
;    04-09-99  GH      Added host parameter bit 14 to enable database load
;    03-26-99  DKW     Added host parameter bit 13 to do extra read
;                           to flush bridge cache.
;    11-11-98  BR            Dash-06 reinstated,-05 voided by Doc Cont.

;    11-06-98  SCS      Roll back the Dash Number back to FC0161802-05.
;    11-03-98  BR       Dash number change, FC0161802-06.  File now ready
;                           for updates or modifications.
;    10-22-98  GH            Changed default parameter for port login
retry
;                       count to 30. Changed loop reset delay back to 5.
;    10-15-98  GH            Changed default parameter for port down
retry
;                           count and loop reset delay.
;    08-06-98  GH       Changed some default settings
;    05-20-98 SCS       Defined bits in F/W Option filed to match F/W
Spec.
;    04-27-98 DM        Added host parameter bit12 - enable database
storage
;
;    04-16-98 TWT       Add Login Timeout RISC parameter
;                       Add host parameter bit10 - enable LIP full login
;                       Add host parameter bit11 - enable Target Reset
;    04-02-98 TWT       Add maximum LUNs per target parameter
;
; B  02-19-98 TWT XXXX Add host parameter block bits:
;                      bit8 - enable 64bit addressing
;                      bit9 - enable LIP reset
;    02-20-98 GH        Changed location where subsystem vendor and
;                       device id are stored.
; A  06-17-97 scs XXXX  Initial release. (SCS)
;
; This file is the default settings for the NVRAM and must have the
extension
; of DAT.  Edit this file to change parameters.
;
;       Numbers are in decimal. File can also contain comments.
;
;*******************************************************************
*
;     !!!!!     DO NOT CHANGE THE ORDER OF THE DATA     !!!!!!!!!!!!
;*******************************************************************
*
;
; The NVRAM will be layed out in the same order as this file.

;***********************************
; NVRAM header

id                      ["4 characters"] = "ISP "
version                 [0-255] = 1
reserved                [0-255] = 0

;***********************************
; NVRAM RISC parameter block

version                 [0-255] = 1
reserved                [0-255] = 0

; Firmware options

enable hard loop ID     [0-1] = 0
enable fairness         [0-1] = 1
enable full duplex      [0-1] = 0
enable fast posting     [0-1] = 1
enable target mode      [0-1] = 0
disable initiator mode  [0-1] = 0
Enable ADISC            [0-1] = 0
Reserved                [0-1] = 0
Enable PDBC Notify      [0-1] = 0
Disable Initial LIP     [0-1] = 0
Descending Search LoopID[0-1] = 0
Reserved                [0-1] = 0
Reserved                [0-1] = 0
Full Login After LIP    [0-1] = 1
Reserved                [0-1] = 0
unused                  [0-1] = 1

frame payload size      [0-65535] = 1024
max iocb allocation     [0-65535] = 256
execution throttle      [0-65535] = 45
retry count             [0-255] = 30
retry delay             [0-255] = 1

node name 0 (must be zero)  [0-255] = 32
node name 1 (must be zero)  [0-255] = 0
node name 2 (company ID)    [0-255] = 0
node name 3 (company ID)    [0-255] = 224
node name 4 (company ID)    [0-255] = 139
node name 5 (serial number) [0-255] = 0
node name 6 (serial number) [0-255] = 174
node name 7 (serial number) [0-255] = 81

adapter hard loop ID    [0-65535] = 125
unused                  [0-255] = 0
login timeout           [0-255] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

;************************************
; Reserved for expanded RISC parameter block

unused                  [0-65535] = 51
unused                  [0-65535] = 2560
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
```

```
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

;*************************************
; NVRAM host parameter block

unused                  [0-1] = 0
disable BIOS            [0-1] = 1
disable LUNs            [0-1] = 0
enable selectable boot  [0-1] = 0
disable RISC code load  [0-1] = 0
set cache line size 1   [0-1] = 0
PCI Parity Disable      [0-1] = 0
enable extended logging [0-1] = 0
enable 64bit addressing [0-1] = 1
enable LIP reset        [0-1] = 0
enable LIP full login   [0-1] = 1
enable target reset     [0-1] = 0
enable database storage [0-1] = 0
enable cache flush read [0-1] = 0
enable database load    [0-1] = 0
unused                  [0-1] = 0

boot node name 0        [0-255] = 0
boot node name 1        [0-255] = 0
boot node name 2        [0-255] = 0
boot node name 3        [0-255] = 0
boot node name 4        [0-255] = 0
boot node name 5        [0-255] = 0
boot node name 6        [0-255] = 0
boot node name 7        [0-255] = 0
boot LUN number         [0-255] = 0

reset delay             [0-255] = 5
port down retry count   [0-255] = 30
reserved1               [0-255] = 0

maximum LUNs per target [0-65535] = 16
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 100

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
```

```
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 150

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 200

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
```

```
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Subsystem vendor ID must be at offset 244

subsystem vendor ID     [0-65535] = 8430

unused                  [0-65535] = 0

; Subsystem device ID must be at offset 248

subsystem device ID     [0-65535] = 2

; Subsystem vendor ID for ISP2100a

subsystem vendor ID     [0-65535] = 4215

; Subsystem device ID for ISP2100a

subsystem device ID     [0-65535] = 1
unused                  [0-255] = 0
checksum                [0-255] = 225
;Document number: FC0161802-07  File name: FC010207.DAT
;Document Name: NVRAM DATA, QLA2100, ISP2100
;
;APPLICATION:   ISP2100 FIBRE CHANNEL
;
;
;NOTE:  A companion file with ".DEF" file extension must be present to
;       insure that file comparision during DVT testing will pass.
;
;Revision History:
;REV   DATE   WHO ECO#  DESCRIPTION OF CHANGE
;
;    06-11-99  BJR     Updated Doc # and File name to dash 07.
;    04-09-99  GH      Added host parameter bit 14 to enable database load
;    03-26-99  DKW     Added host parameter bit 13 to do extra read
;                            to flush bridge cache.
;    11-11-98  BR              Dash-06 reinstated,-05 voided by Doc Cont.

;    11-06-98  SCS       Roll back the Dash Number back to FC0161802-05.
;    11-03-98  BR        Dash number change, FC0161802-06.  File now ready
;                            for updates or modifications.
;    10-22-98  GH              Changed default parameter for port login
retry
;                         count to 30. Changed loop reset delay back to 5.
;    10-15-98  GH              Changed default parameter for port down
retry
;                            count and loop reset delay.
;    08-06-98  GH        Changed some default settings
;    05-20-98 SCS        Defined bits in F/W Option filed to match F/W
Spec.
;    04-27-98 DM         Added host parameter bit12 - enable database
storage
;
;    04-16-98 TWT        Add Login Timeout RISC parameter
;                        Add host parameter bit10 - enable LIP full login
;                        Add host parameter bit11 - enable Target Reset
;    04-02-98 TWT        Add maximum LUNs per target parameter
;
; B  02-19-98 TWT XXXX Add host parameter block bits:
;                      bit8 - enable 64bit addressing
;                      bit9 - enable LIP reset
;    02-20-98 GH         Changed location where subsystem vendor and
;                        device id are stored.
; A  06-17-97 scs XXXX  Initial release. (SCS)
;
; This file is the default settings for the NVRAM and must have the
extension
; of DAT.  Edit this file to change parameters.
;
;       Numbers are in decimal. File can also contain comments.
;
;***************************************************************************
*
;     !!!!!     DO NOT CHANGE THE ORDER OF THE DATA     !!!!!!!!!!!!
;***************************************************************************
*
;
; The NVRAM will be layed out in the same order as this file.

;************************************
; NVRAM header

id                      ["4 characters"] = "ISP "
version                 [0-255] = 1
reserved                [0-255] = 0

;************************************
; NVRAM RISC parameter block

version                 [0-255] = 1
reserved                [0-255] = 0

; Firmware options

enable hard loop ID     [0-1] = 0
enable fairness         [0-1] = 1
enable full duplex      [0-1] = 0
enable fast posting     [0-1] = 1
enable target mode      [0-1] = 0
disable initiator mode  [0-1] = 0
Enable ADISC            [0-1] = 0
Reserved                [0-1] = 0
Enable PDBC Notify      [0-1] = 0
Disable Initial LIP     [0-1] = 0
Descending Search LoopID[0-1] = 0
Reserved                [0-1] = 0
Reserved                [0-1] = 0
```

```
Full Login After LIP    [0-1] = 1
Reserved                [0-1] = 0
unused                  [0-1] = 1

frame payload size      [0-65535] = 1024
max iocb allocation     [0-65535] = 256
execution throttle      [0-65535] = 45
retry count             [0-255] = 30
retry delay             [0-255] = 1

node name 0 (must be zero)  [0-255] = 32
node name 1 (must be zero)  [0-255] = 0
node name 2 (company ID)    [0-255] = 0
node name 3 (company ID)    [0-255] = 224
node name 4 (company ID)    [0-255] = 139
node name 5 (serial number) [0-255] = 0
node name 6 (serial number) [0-255] = 10
node name 7 (serial number) [0-255] = 47

adapter hard loop ID    [0-65535] = 125
unused                  [0-255] = 0
login timeout           [0-255] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

;*************************************
; Reserved for expanded RISC parameter block

unused                  [0-65535] = 51
unused                  [0-65535] = 2560
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

;*************************************
; NVRAM host parameter block

unused                  [0-1] = 0
disable BIOS            [0-1] = 1
disable LUNs            [0-1] = 0
enable selectable boot  [0-1] = 0
disable RISC code load  [0-1] = 0
set cache line size 1   [0-1] = 0
PCI Parity Disable      [0-1] = 0
enable extended logging [0-1] = 0
enable 64bit addressing [0-1] = 1
enable LIP reset        [0-1] = 0
enable LIP full login   [0-1] = 1
enable target reset     [0-1] = 0
enable database storage [0-1] = 0
enable cache flush read [0-1] = 0
enable database load    [0-1] = 0
unused                  [0-1] = 0

boot node name 0        [0-255] = 0
boot node name 1        [0-255] = 0
boot node name 2        [0-255] = 0
boot node name 3        [0-255] = 0
boot node name 4        [0-255] = 0
boot node name 5        [0-255] = 0
boot node name 6        [0-255] = 0
boot node name 7        [0-255] = 0
boot LUN number         [0-255] = 0

reset delay             [0-255] = 5
port down retry count   [0-255] = 30
reserved1               [0-255] = 0

maximum LUNs per target [0-65535] = 16
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 100

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
```

```
; Offset 150

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 200

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Subsystem vendor ID must be at offset 244

subsystem vendor ID     [0-65535] = 8430

unused                  [0-65535] = 0
```

```
; Subsystem device ID must be at offset 248

subsystem device ID     [0-65535] = 2

; Subsystem vendor ID for ISP2100a

subsystem vendor ID     [0-65535] = 4215

; Subsystem device ID for ISP2100a

subsystem device ID     [0-65535] = 1
unused                  [0-255] = 0
checksum                [0-255] = 167
;Document number: FC0161802-07  File name: FC010207.DAT
;Document Name: NVRAM DATA, QLA2100, ISP2100
;
;APPLICATION:   ISP2100 FIBRE CHANNEL
;
;
;NOTE:  A companion file with ".DEF" file extension must be present to
;       insure that file comparision during DVT testing will pass.
;
;Revision History:
;REV   DATE    WHO ECO#  DESCRIPTION OF CHANGE
;
;    06-11-99  BJR    Updated Doc # and File name to dash 07.
;    04-09-99  GH     Added host parameter bit 14 to enable database load
;    03-26-99  DKW    Added host parameter bit 13 to do extra read
;                             to flush bridge cache.
;    11-11-98  BR             Dash-06 reinstated,-05 voided by Doc Cont.

;    11-06-98  SCS     Roll back the Dash Number back to FC0161802-05.
;    11-03-98  BR      Dash number change, FC0161802-06.  File now ready
;                             for updates or modifications.
;    10-22-98  GH             Changed default parameter for port login
retry
;                      count to 30. Changed loop reset delay back to 5.
;    10-15-98  GH             Changed default parameter for port down
retry
;                             count and loop reset delay.
;    08-06-98  GH      Changed some default settings
;    05-20-98 SCS      Defined bits in F/W Option filed to match F/W
Spec.
;    04-27-98 DM       Added host parameter bit12 - enable database
storage
;
;    04-16-98 TWT      Add Login Timeout RISC parameter
;                      Add host parameter bit10 - enable LIP full login
;                      Add host parameter bit11 - enable Target Reset
;    04-02-98 TWT      Add maximum LUNs per target parameter
;
; B  02-19-98 TWT XXXX Add host parameter block bits:
;                     bit8 - enable 64bit addressing
;                     bit9 - enable LIP reset
;    02-20-98 GH       Changed location where subsystem vendor and
;                      device id are stored.
; A  06-17-97 scs XXXX  Initial release. (SCS)
;
```

```
; This file is the default settings for the NVRAM and must have the
extension
; of DAT.  Edit this file to change parameters.
;
;       Numbers are in decimal. File can also contain comments.
;
;***************************************************************************
*
;     !!!!!     DO NOT CHANGE THE ORDER OF THE DATA     !!!!!!!!!!!!
;***************************************************************************
*
;
; The NVRAM will be layed out in the same order as this file.

;************************************
; NVRAM header

id                      ["4 characters"] = "ISP "
version                 [0-255] = 1
reserved                [0-255] = 0

;************************************
; NVRAM RISC parameter block

version                 [0-255] = 1
reserved                [0-255] = 0

; Firmware options

enable hard loop ID     [0-1] = 0
enable fairness         [0-1] = 1
enable full duplex      [0-1] = 0
enable fast posting     [0-1] = 1
enable target mode      [0-1] = 0
disable initiator mode  [0-1] = 0
Enable ADISC            [0-1] = 0
Reserved                [0-1] = 0
Enable PDBC Notify      [0-1] = 0
Disable Initial LIP     [0-1] = 0
Descending Search LoopID[0-1] = 0
Reserved                [0-1] = 0
Reserved                [0-1] = 0
Full Login After LIP    [0-1] = 1
Reserved                [0-1] = 0
unused                  [0-1] = 1

frame payload size      [0-65535] = 1024
max iocb allocation     [0-65535] = 256
execution throttle      [0-65535] = 45
retry count             [0-255] = 30
retry delay             [0-255] = 1

node name 0 (must be zero)  [0-255] = 32
node name 1 (must be zero)  [0-255] = 0
node name 2 (company ID)    [0-255] = 0
node name 3 (company ID)    [0-255] = 224
node name 4 (company ID)    [0-255] = 139
node name 5 (serial number) [0-255] = 0
node name 6 (serial number) [0-255] = 98
node name 7 (serial number) [0-255] = 47

adapter hard loop ID    [0-65535] = 125
unused                  [0-255] = 0
login timeout           [0-255] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

;************************************
; Reserved for expanded RISC parameter block

unused                  [0-65535] = 51
unused                  [0-65535] = 2560
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

;************************************
; NVRAM host parameter block

unused                  [0-1] = 0
disable BIOS            [0-1] = 1
disable LUNs            [0-1] = 0
enable selectable boot  [0-1] = 0
disable RISC code load  [0-1] = 0
set cache line size 1   [0-1] = 0
PCI Parity Disable      [0-1] = 0
enable extended logging [0-1] = 0
enable 64bit addressing [0-1] = 1
enable LIP reset        [0-1] = 0
enable LIP full login   [0-1] = 1
enable target reset     [0-1] = 0
enable database storage [0-1] = 0
enable cache flush read [0-1] = 0
enable database load    [0-1] = 0
unused                  [0-1] = 0

boot node name 0        [0-255] = 0
boot node name 1        [0-255] = 0
boot node name 2        [0-255] = 0
boot node name 3        [0-255] = 0
boot node name 4        [0-255] = 0
boot node name 5        [0-255] = 0
boot node name 6        [0-255] = 0
boot node name 7        [0-255] = 0
boot LUN number         [0-255] = 0
```

```
reset delay              [0-255] = 5
port down retry count    [0-255] = 30
reserved1                [0-255] = 0

maximum LUNs per target  [0-65535] = 16
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0

; Offset 100

unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0

; Offset 150

unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0

; Offset 200

unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0
unused                   [0-65535] = 0

; Subsystem vendor ID must be at offset 244

subsystem vendor ID      [0-65535] = 8430

unused                   [0-65535] = 0

; Subsystem device ID must be at offset 248

subsystem device ID      [0-65535] = 2

; Subsystem vendor ID for ISP2100a

subsystem vendor ID      [0-65535] = 4215

; Subsystem device ID for ISP2100a

subsystem device ID      [0-65535] = 1
unused                   [0-255] = 0
checksum                 [0-255] = 79
;Document number: FC0161802-07  File name: FC010207.DAT
;Document Name: NVRAM DATA, QLA2100, ISP2100
;
;APPLICATION:   ISP2100 FIBRE CHANNEL
```

```
;
;
;NOTE:  A companion file with ".DEF" file extension must be present to
;       insure that file comparision during DVT testing will pass.
;
;Revision History:
;REV   DATE   WHO ECO#  DESCRIPTION OF CHANGE
;
;    06-11-99  BJR     Updated Doc # and File name to dash 07.
;    04-09-99  GH      Added host parameter bit 14 to enable database load
;    03-26-99  DKW     Added host parameter bit 13 to do extra read
;                            to flush bridge cache.
;    11-11-98  BR              Dash-06 reinstated,-05 voided by Doc Cont.

;    11-06-98  SCS      Roll back the Dash Number back to FC0161802-05.
;    11-03-98  BR       Dash number change, FC0161802-06.  File now ready
;                            for updates or modifications.
;    10-22-98  GH              Changed default parameter for port login
retry
;                       count to 30. Changed loop reset delay back to 5.
;    10-15-98  GH              Changed default parameter for port down
retry
;                            count and loop reset delay.
;    08-06-98  GH       Changed some default settings
;    05-20-98 SCS       Defined bits in F/W Option filed to match F/W
Spec.
;    04-27-98 DM        Added host parameter bit12 - enable database
storage
;
;    04-16-98 TWT       Add Login Timeout RISC parameter
;                       Add host parameter bit10 - enable LIP full login
;                       Add host parameter bit11 - enable Target Reset
;    04-02-98 TWT       Add maximum LUNs per target parameter
;
; B  02-19-98 TWT XXXX Add host parameter block bits:
;                      bit8 - enable 64bit addressing
;                      bit9 - enable LIP reset
;    02-20-98 GH        Changed location where subsystem vendor and
;                       device id are stored.
; A  06-17-97 scs XXXX  Initial release. (SCS)
;
; This file is the default settings for the NVRAM and must have the
extension
; of DAT.  Edit this file to change parameters.
;
;       Numbers are in decimal. File can also contain comments.
;
;***************************************************************************
*
;     !!!!!     DO NOT CHANGE THE ORDER OF THE DATA     !!!!!!!!!!!!
;***************************************************************************
*
;
; The NVRAM will be layed out in the same order as this file.

;***********************************
; NVRAM header

id                      ["4 characters"] = "ISP "
version                 [0-255] = 1
reserved                [0-255] = 0

;***********************************
; NVRAM RISC parameter block

version                 [0-255] = 1
reserved                [0-255] = 0

; Firmware options

enable hard loop ID     [0-1] = 0
enable fairness         [0-1] = 1
enable full duplex      [0-1] = 0
enable fast posting     [0-1] = 1
enable target mode      [0-1] = 0
disable initiator mode  [0-1] = 0
Enable ADISC            [0-1] = 0
Reserved                [0-1] = 0
Enable PDBC Notify      [0-1] = 0
Disable Initial LIP     [0-1] = 0
Descending Search LoopID[0-1] = 0
Reserved                [0-1] = 0
Reserved                [0-1] = 0
Full Login After LIP    [0-1] = 1
Reserved                [0-1] = 0
unused                  [0-1] = 1

frame payload size      [0-65535] = 1024
max iocb allocation     [0-65535] = 256
execution throttle      [0-65535] = 45
retry count             [0-255] = 30
retry delay             [0-255] = 1

node name 0 (must be zero)  [0-255] = 32
node name 1 (must be zero)  [0-255] = 0
node name 2 (company ID)    [0-255] = 0
node name 3 (company ID)    [0-255] = 224
node name 4 (company ID)    [0-255] = 139
node name 5 (serial number) [0-255] = 0
node name 6 (serial number) [0-255] = 224
node name 7 (serial number) [0-255] = 82

adapter hard loop ID    [0-65535] = 125
unused                  [0-255] = 0
login timeout           [0-255] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

;***********************************
; Reserved for expanded RISC parameter block

unused                  [0-65535] = 51
unused                  [0-65535] = 2560
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
```

```
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

;*************************************
; NVRAM host parameter block

unused                  [0-1] = 0
disable BIOS            [0-1] = 1
disable LUNs            [0-1] = 0
enable selectable boot  [0-1] = 0
disable RISC code load  [0-1] = 0
set cache line size 1   [0-1] = 0
PCI Parity Disable      [0-1] = 0
enable extended logging [0-1] = 0
enable 64bit addressing [0-1] = 1
enable LIP reset        [0-1] = 0
enable LIP full login   [0-1] = 1
enable target reset     [0-1] = 0
enable database storage [0-1] = 0
enable cache flush read [0-1] = 0
enable database load    [0-1] = 0
unused                  [0-1] = 0

boot node name 0        [0-255] = 0
boot node name 1        [0-255] = 0
boot node name 2        [0-255] = 0
boot node name 3        [0-255] = 0
boot node name 4        [0-255] = 0
boot node name 5        [0-255] = 0
boot node name 6        [0-255] = 0
boot node name 7        [0-255] = 0
boot LUN number         [0-255] = 0

reset delay             [0-255] = 5
port down retry count   [0-255] = 30
reserved1               [0-255] = 0

maximum LUNs per target [0-65535] = 16
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 100

unused                  [0-65535] = 0
unused                  [0-65535] = 0
```

```
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 150

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Offset 200

unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
```

```
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0
unused                  [0-65535] = 0

; Subsystem vendor ID must be at offset 244

subsystem vendor ID     [0-65535] = 8430

unused                  [0-65535] = 0

; Subsystem device ID must be at offset 248

subsystem device ID     [0-65535] = 2

; Subsystem vendor ID for ISP2100a

subsystem vendor ID     [0-65535] = 4215

; Subsystem device ID for ISP2100a

subsystem device ID     [0-65535] = 1
unused                  [0-255] = 0
checksum                [0-255] = 174
```

Client Configuration Information

Tuxedo Client

System Configuration

```
# psrinfo
0 on-line since 12/09/99 18:20:35
1 on-line since 12/09/99 18:21:26
2 on-line since 12/09/99 18:21:28
3 on-line since 12/09/99 18:21:30
#
#
# memsize -a
  1073324032  Total memory
  1073307648  Usable memory
  1040351232  General memory
           0  Dedicated memory

Memory and Disks
================
    SYSTEM CONFIGURATION:

    Memory Size: 1024 Megabytes
    System Peripherals:

        Floppy Drive 1 - 1.44 MB 3.5
        SCSI CD-Rom Drive 1 - TOSHIBA  - CD-ROM XM-6201TA
        Disk Drive 1 - UNISYS   - 003557M2954E-512 - 4141 MB
        Disk Drive 2 - SEAGATE  - ST34501W         - 4337 MB
        Disk Drive 3 - DGC      - RAID 0
        Disk Drive 4 - DGC      - RAID 0
        Disk Drive 5 - DGC      - RAID 0
        Disk Drive 6 - DGC      - RAID 0
        Disk Drive 7 - DGC      - RAID 10
        Disk Drive 8 - DGC      - DISK
       80387 Math Processor

List of packages
================
application BASEdoc        UnixWare Documentation
application BASEman        UnixWare Manual Pages
system      DLPI           BUILT IN TO NETWORK INTERFACE CARD SUPPORT,
CANNOT BE REMOVED.
application FTRKdoc        Netscape FastTrack Server 2.01 Documentation
system      MDI            BUILT IN TO NETWORK INTERFACE CARD SUPPORT,
CANNOT BE REMOVED.
application TEDdesk        TriTeal Enterprise Desktop (CDE Desktop)
application TEDdocs        CDE Desktop Postscript Manuals
application TEDhelp        CDE Online Help
application TEDlogin       CDE Login Manager
application TEDman         CDE Manual Pages
system      acl            Access Control List Utilities
system      acp            Enhanced Application Compatibility
system      adsl           Adaptec 7800 Family PCI SCSI IHV HBA
system      audio          Audio Subsystem
system      audit          Auditing Subsystem
system      base           Base System
graphics    basex          X11R6 Base X Runtime System
system      bsdcompat      BSD Compatibility
system      cdfs           BUILT INTO THE BASE, CANNOT BE REMOVED.
system      cmds           Advanced Commands
system      compat         BUILT INTO BSD COMPATIBILITY, CANNOT BE
REMOVED.
system      dfm            BUILT INTO THE BASE, CANNOT BE REMOVED.
utilities   dfs            Distributed File System Utilities
system      dmi            Desktop Management Interface
system      dshm           Large Memory Support Package
system      ed             BUILT INTO ADVANCED COMMANDS, CANNOT BE
REMOVED.
system      els            BUILT INTO LANGUAGE SUPPLEMENT, CANNOT BE
REMOVED.
system      eth            BUILT INTO NETWORK INTERFACE CARD SUPPORT,
CANNOT BE REMOVED.
```

```
system      expect          expect
system      fmli            BUILT INTO ADVANCED COMMANDS, CANNOT BE
REMOVED.
system      ide             Generic IDE/ATAPI/ESDI Driver
system      inet            Internet Utilities
system      jdk117          JDK 1.1.7 for SCO
system      kdb             Kernel Debugger
system      ldap            Lightweight Directory Access Protocol services
system      lp              Printer Support
system      ls              Language Supplement
system      mouse           BUILT INTO THE BASE, CANNOT BE REMOVED.
system      mpio            Multipath I/O Driver
graphics    mtfrun          OSF Motif 1.2.5 Runtime Environment
system      nd              Network Drivers
system      netbios         TPI NetBIOS
system      netcmds         BUILT INTO NETWORK SUPPORT UTILITIES, CANNOT BE
REMOVED.
system      netmgt          Network Management Utilities
system      nfs             Network File System Utilities
system      nics            Netdriver Infrastructure and Configuration
Subsystem
system      nis             Network Information Service
application nsadmin         Netscape Administration Server 2.13
application nscomm          Netscape Communicator 4.08
application nscommD         Netscape Communicator 4.08 Documentation
application nsfast          Netscape FastTrack Server 2.01a
system      nsu             Network Support Utilities
system      nuc             NetWare UNIX Client
system      nwnet           NetWare Networking
system      osmp            OS Multiprocessor Support (OSMP)
utilities   perl5           Perl 5.004
system      ppp             The Point-to-Point Protocol (PPP)
system      qlc2100         QLogic QLC2100 Fibre Channel IHV HBA Driver
system      qt              BUILT INTO THE BASE, CANNOT BE REMOVED.
system      rpc             Remote Procedure Calls Utilities
system      scoadmin        SCO System Administration (SCOadmin)
application scohelp         Scohelp Online Doc System
system      server          Server Utilities
system      sys             BUILT INTO THE BASE, CANNOT BE REMOVED.
system      tclrun          Tcl Runtime package
system      termcap         BUILT INTO ENHANCED APPLICATION COMPATIBILITY,
CANNOT BE REMOVED.
system      terminf         Terminfo Utilities
system      tok             BUILT INTO NETWORK INTERFACE CARD SUPPORT,
CANNOT BE REMOVED.
system      uccs            UDK Optimizing C Compilation System
system      uedebug         UDK Enhanced Debugger
update      update710       UnixWare 7 Update 7.1.0
update      update711       UnixWare 7 Update 7.1.1
system      vtclrun         Vtcl Runtime package
application vxva            VERITAS ODM Visual Administrator
system      vxvm            VERITAS Volume Manager
graphics    xclients        X11R6 X Clients
graphics    xcontrib        X11R6 Contributed X Clients
graphics    xdrivers        X11R6 Graphics Drivers
graphics    xfonts          X11R6 Fonts
graphics    xserver         X11R6 X Server
```

```
Custom Installed Packages

ALL     Part    Entire Run Time System with perms lists
RTS     Part    UNIX Run Time System
ALL     Part    Entire TCP/IP Runtime System
TCPRT   Part    TCP/IP Runtime utilities and drivers
SNDMAIL Part    Sendmail Runtime
NETBIOS Part    NetBIOS Runtime utilities and drivers
PPP     Part    PPP Runtime utilities and drivers

Oracle profile
==============
export TERM=at386
export EDITOR=vi
export PS1='oratest2:$PWD> '
alias ls='/bin/ls -CF $*'
alias env='/bin/env | sort'
alias bench='cd /home/oracle/bench/tpc/tpcc'
alias rund='cd /home/oracle/8i/bench/tpc/tpcc/scripts/build3500'
alias bkup='cd /home/oracle/8i/dbs/tpcc_backup'
alias term1='export TERM=vt100;'
alias ls='ls -F'
stty intr ^C

export ORACLE_HOME=/home/oracle/8i
export SRCHOME=/home/oracle/8i
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/java/lib/x86at/native_threads
export ORACLE_SID=tpcc
export
PATH=$ORACLE_HOME/bin:$PATH:$ORACLE_HOME/bench/tpc/bin:$ORACLE_HOME/bench/
tpc/tpcc/scripts:/usr/ccs/bin:/etc:.:/usr/bin/X11
export ORA_NLS=$ORACLE_HOME/ocommon/nls/admin/data
export ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data
export TNS_ADMIN=$ORACLE_HOME/network/admin
export DSHM_DEBUG=TRUE
export VLM_FORCE_UNMAP=n
PASSREQ=NO

Tuxedo Environment
==================
export TUXDIR=/home/tuxedo
export APPDIR=/home/tuxedo/runtime
export PATH=$PATH:$TUXDIR/bin:$TUXDIR/runtime
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$TUXDIR/lib
export TUXCONFIG=$TUXDIR/runtime/tuxconfig

TNSNAMES.ORA
============
# TNSNAMES.ORA Configuration File:/oracle/8i/network/admin/tnsnames.ora
# Generated by Oracle Net8 Assistant

tpcc =
  (DESCRIPTION =
       (SDU = 4096)
       (TDU = 16384)
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.59.13.114)(PORT = 1521))
    )
    (CONNECT_DATA =
```

```
      (SERVICE_NAME = tpcc)
    )
  )

EXTPROC_CONNECTION_DATA.MV.UNISYS.COM =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )

Profile of the Root user
========================
ksh -o vi
inconfig tcp_do_rfc1323 0

stune parameters
================
NUMREGPT       256
NUMSCOPT       32
DRV_SEGKMEM_BYTES      0x147A000
RSTCHOWN       0
CONSOLE_SECURITY       0
SHMMNI  200
SHMSEG  15
NPROC   2000
MAXUP   2000
GENERAL_MEMORY 230000
SHMMAX  2097152
SEMMNI  2048
SEMMSL  300
MSGMAX  2048
MSGMNB  2097152
MSGSSZ  1048576
MSGTQL  1650
MSGMNI  255
MSGSEG  16
SEMMNS  2048

Device Configurations
=====================
0:0,7,0: HBA     : (adsl,1) Adaptec PCI SCSI
  0,0,0: DISK    : UNISYS  003557M2954E-5120657
  0,1,0: DISK    : SEAGATE ST34501W        0018
  0,4,0: CDROM   : TOSHIBA CD-ROM XM-6201TA1037
  0,6,0: HBA     : ESG-SHV SCA HSBP M6     0.63
```

Tuxedo Configuration

```
*RESOURCES
IPCKEY         133133
MAXACCESSERS   1500
MAXSERVERS     250
MAXSERVICES    1100
MODEL          MP
MASTER         oratest2
LDBAL          Y
SCANUNIT       60
BLOCKTIME      60
BBLQUERY       60
OPTIONS        LAN

*MACHINES
DEFAULT:

oratest2       LMID=oratest2
               TUXDIR="/home/tuxedo"
               APPDIR="/home/tuxedo/runtime"
               TUXCONFIG="/home/tuxedo/runtime/tuxconfig"
               ULOGPFX="/home/tuxedo/runtime/ulog/ULOG"
               UID=101
               GID=101

CLIENT1        LMID=client1
               TUXDIR="c:\tuxedo"
               APPDIR="c:\tuxedo\runtime"
               TUXCONFIG="c:\tuxedo\runtime\tuxconfig"
               ULOGPFX="c:\tuxedo\runtime\ulog\ULOG"
               UID=0
               GID=0

CLIENT2        LMID=client2
               TUXDIR="c:\tuxedo"
               APPDIR="c:\tuxedo\runtime"
               TUXCONFIG="c:\tuxedo\runtime\tuxconfig"
               ULOGPFX="c:\tuxedo\runtime\ulog\ULOG"
               UID=0
               GID=0

CLIENT3        LMID=client3
               TUXDIR="c:\tuxedo"
               APPDIR="c:\tuxedo\runtime"
               TUXCONFIG="c:\tuxedo\runtime\tuxconfig"
               ULOGPFX="c:\tuxedo\runtime\ulog\ULOG"
               UID=0
               GID=0

CLIENT4        LMID=client4
               TUXDIR="c:\tuxedo"
               APPDIR="c:\tuxedo\runtime"
               TUXCONFIG="c:\tuxedo\runtime\tuxconfig"
               ULOGPFX="c:\tuxedo\runtime\ulog\ULOG"
               UID=0
               GID=0

*GROUPS
GROUPUX
       LMID=oratest2    GRPNO=1   OPENINFO=NONE

*SERVERS
```

```
DEFAULT:
            CLOPT="-A -- -utpcc -ptpcc -stpcc -a75 -l1"

tpccsvr     SRVGRP=GROUPUX
            SRVID=101
            MIN=12 MAX=25
            RQADDR=tpccq1 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=126
            MIN=12 MAX=25
            RQADDR=tpccq2 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=151
            MIN=12 MAX=25
            RQADDR=tpccq3 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=176
            MIN=12 MAX=25
            RQADDR=tpccq4 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=201
            MIN=12 MAX=25
            RQADDR=tpccq5 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=226
            MIN=12 MAX=25
            RQADDR=tpccq6 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=251
            MIN=12 MAX=25
            RQADDR=tpccq7 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=276
            MIN=12 MAX=25
            RQADDR=tpccq8 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=301
            MIN=12 MAX=25
            RQADDR=tpccq9 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=326
            MIN=11 MAX=25
            RQADDR=tpccq10 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=351
            MIN=11 MAX=25
            RQADDR=tpccq11 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=376
            MIN=11 MAX=25
            RQADDR=tpccq12 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=401
            MIN=11 MAX=25
            RQADDR=tpccq13 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=426
            MIN=11 MAX=25
            RQADDR=tpccq14 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=451
            MIN=11 MAX=25
            RQADDR=tpccq15 REPLYQ=Y

tpccsvr     SRVGRP=GROUPUX
            SRVID=476
            MIN=11 MAX=25
            RQADDR=tpccq16 REPLYQ=Y

*SERVICES

*NETWORK
client1     NADDR="//192.158.91.1:3051"
            NLSADDR="//192.158.91.1:3050"

client2     NADDR="//192.158.91.2:3051"
            NLSADDR="//192.158.91.2:3050"

client3     NADDR="//192.158.91.3:3051"
            NLSADDR="//192.158.91.3:3050"

client4     NADDR="//192.158.91.4:3051"
            NLSADDR="//192.158.91.4:3050"

oratest2    NADDR="//192.158.91.132:8001" BRIDGE="/dev/tcp"
            NLSADDR="//192.158.91.132:8000"
```

Web Clients

System Configuration

```
Microsoft Diagnostics Report For \\CLIENT1
------------------------------------------------------------------------

OS Version Report
------------------------------------------------------------------------
Microsoft (R) Windows NT (TM) Server
Version 4.0 (Build 1381: Service Pack 4) x86 Multiprocessor Free
Registered Owner: Unisys, Unisys
```

```
Product Number: 28997-OEM-0026051-97737
------------------------------------------------------------------------

System Report
------------------------------------------------------------------------
System: AT/AT COMPATIBLE
Hardware Abstraction Layer: MPS 1.4 - APIC platform
BIOS Date: 01/11/99
BIOS Version: <unavailable>

Processor list:
   0:  x86 Family 6 Model 7 Stepping 2 GenuineIntel ~499 Mhz
   1:  x86 Family 6 Model 7 Stepping 2 GenuineIntel ~499 Mhz
------------------------------------------------------------------------

Video Display Report
------------------------------------------------------------------------
BIOS Date: 05/21/97
BIOS Version: CL-GD5446 PCI VGA BIOS Version 1.33

Adapter:
   Setting: 800 x 600 x 65536
            60 Hz
   Type: cirrus compatible display adapter
   String: Cirrus Logic Compatible
   Memory: 1 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: 4,160,803 KB, Free: 2,652,870 KB
   Serial Number: 5C93 - 5932
   Bytes per cluster: 512
   Sectors per cluster: 1
   Filename length: 255

Memory Report
------------------------------------------------------------------------
Handles: 1,075
Threads: 109
Processes: 17

Physical Memory (K)
   Total: 1,047,984
   Available: 979,984
   File Cache: 14,372

Kernel Memory (K)
   Total: 22,524
   Paged: 8,308
   Nonpaged: 14,216

Commit Charge (K)
   Total: 34,644
   Limit: 2,049,184
   Peak: 224,836

Pagefile Space (K)
   Total: 1,048,576
   Total in use: 14,448
   Peak: 15,924

   C:\pagefile.sys
      Total: 1,048,576
      Total in use: 14,448
      Peak: 15,924



Services Report
------------------------------------------------------------------------
Alerter                                          Stopped   (Disabled)
   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
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                                    Stopped   (Disabled)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Service Dependencies:
      LanmanWorkstation
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
OracleOraHome81ClientCache                   Stopped   (Manual)
   C:\Oracle\Ora81\BIN\ONRSD.EXE
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
OracleWebAssistant0                          Stopped   (Disabled)
   C:\Oracle\Ora81\BIN\OWASTSVR.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
Protected Storage                            Running   (Automatic)
   c:\winnt\system32\pstores.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process, Interactive
   Service Dependencies:
      RpcSs
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                                     Running   (Automatic)
   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                            Stopped   (Automatic)
   C:\TUXEDO\bin\tuxipc.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
TListen (Port: 3050)                         Stopped   (Disabled)
```

```
   C:\TUXEDO\bin\slisten.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
World Wide Web Publishing Service          Stopped   (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)                    Running   (Boot)
   C:\WINNT\System32\DRIVERS\aic78xx.sys
   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
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)                       Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
Fd16_700 (SCSI miniport)                         Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Fd7000ex (SCSI miniport)                         Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Fd8xx (SCSI miniport)                            Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
flashpnt (SCSI miniport)                         Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Floppy (Primary disk)                            Running   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Ftdisk (Filter)                                  Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
HP 10/100TX PCI Ethernet Adapter Driver (NDIS) Running   (Automatic)
   C:\WINNT\System32\drivers\hptxnt.sys
   Error Severity: Normal
   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
NetBEUI Protocol (PNP_TDI)                       Running   (Automatic)
   C:\WINNT\System32\drivers\nbf.sys
   Error Severity: Normal
   Service Flags: Kernel 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
WINS Client(TCP/IP) (PNP_TDI)                    Stopped   (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)                      Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Serial (Extended base)                     Running   (Automatic)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Sermouse (Pointer Port)                    Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Sfloppy (Primary disk)                     Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Simbad (Filter)                            Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
slcd32 (SCSI miniport)                     Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Sparrow (SCSI miniport)                    Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Spock (SCSI miniport)                      Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Srv (Network)                              Running   (Manual)
   C:\WINNT\System32\drivers\srv.sys
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
symc810 (SCSI miniport)                    Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
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
update (Base)                              Stopped   (System)
   Error Severity: Ignore
   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	0x00000003
MPS 1.4 - APIC platform	0	0	0x00000003
MPS 1.4 - APIC platform	1	1	0x00000003
MPS 1.4 - APIC platform	2	2	0x00000003
MPS 1.4 - APIC platform	3	3	0x00000003
MPS 1.4 - APIC platform	4	4	0x00000003
MPS 1.4 - APIC platform	5	5	0x00000003
MPS 1.4 - APIC platform	6	6	0x00000003
MPS 1.4 - APIC platform	7	7	0x00000003
MPS 1.4 - APIC platform	8	8	0x00000003
MPS 1.4 - APIC platform	9	9	0x00000003
MPS 1.4 - APIC platform	10	10	0x00000003
MPS 1.4 - APIC platform	11	11	0x00000003
MPS 1.4 - APIC platform	12	12	0x00000003
MPS 1.4 - APIC platform	13	13	0x00000003
MPS 1.4 - APIC platform	14	14	0x00000003
MPS 1.4 - APIC platform	15	15	0x00000003
MPS 1.4 - APIC platform	16	16	0x00000003
MPS 1.4 - APIC platform	17	17	0x00000003
MPS 1.4 - APIC platform	18	18	0x00000003
MPS 1.4 - APIC platform	19	19	0x00000003
MPS 1.4 - APIC platform	20	20	0x00000003
MPS 1.4 - APIC platform	21	21	0x00000003
MPS 1.4 - APIC platform	22	22	0x00000003
MPS 1.4 - APIC platform	23	23	0x00000003
MPS 1.4 - APIC platform	24	24	0x00000003
MPS 1.4 - APIC platform	25	25	0x00000003
MPS 1.4 - APIC platform	26	26	0x00000003
MPS 1.4 - APIC platform	27	27	0x00000003
MPS 1.4 - APIC platform	28	28	0x00000003
MPS 1.4 - APIC platform	29	29	0x00000003
MPS 1.4 - APIC platform	30	30	0x00000003
MPS 1.4 - APIC platform	31	31	0x00000003
MPS 1.4 - APIC platform	32	32	0x00000003
MPS 1.4 - APIC platform	33	33	0x00000003
MPS 1.4 - APIC platform	34	34	0x00000003
MPS 1.4 - APIC platform	35	35	0x00000003
MPS 1.4 - APIC platform	36	36	0x00000003
MPS 1.4 - APIC platform	37	37	0x00000003
MPS 1.4 - APIC platform	38	38	0x00000003
MPS 1.4 - APIC platform	39	39	0x00000003
MPS 1.4 - APIC platform	40	40	0x00000003
MPS 1.4 - APIC platform	41	41	0x00000003
MPS 1.4 - APIC platform	42	42	0x00000003
MPS 1.4 - APIC platform	43	43	0x00000003
MPS 1.4 - APIC platform	44	44	0x00000003
MPS 1.4 - APIC platform	45	45	0x00000003
MPS 1.4 - APIC platform	46	46	0x00000003
MPS 1.4 - APIC platform	47	47	0x00000003
MPS 1.4 - APIC platform	61	61	0x00000003
MPS 1.4 - APIC platform	65	65	0x00000003
MPS 1.4 - APIC platform	80	80	0x00000003
MPS 1.4 - APIC platform	193	193	0x00000003
MPS 1.4 - APIC platform	225	225	0x00000003
MPS 1.4 - APIC platform	253	253	0x00000003
MPS 1.4 - APIC platform	254	254	0x00000003
MPS 1.4 - APIC platform	255	255	0x00000003
i8042prt	1	1	0xffffffff
i8042prt	12	12	0xffffffff
Serial	4	4	0x00000000
Serial	3	3	0x00000000
Floppy	6	6	0x00000000
HPTX	36	36	0x00000000
HPTX	32	32	0x00000000
aic78xx	40	40	0x00000000
atapi	0	14	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
Floppy	0x000003f0	0x0000000006
Floppy	0x000003f7	0x0000000001
HPTX	0x0000fca0	0x000000001c
HPTX	0x0000fcc0	0x000000001c

```
aic78xx                         0x0000f800  0x0000000100
atapi                           0x000001f0  0x0000000008
atapi                           0x000003f6  0x0000000001
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         0xfec00000  0x00000400
MPS 1.4 - APIC platform         0xfee00000  0x00000400
HPTX                            0xfecfc000  0x0000001c
HPTX                            0xfecfd000  0x0000001c
aic78xx                         0xfecff000  0x00001000
cirrus                          0x000a0000  0x00020000
cirrus                          0xfc000000  0x02000000



Environment Report
-------------------------------------------------------------------------



System Environment Variables
   APPDIR=c:\tuxedo\runtime
   ComSpec=C:\WINNT\system32\cmd.exe
   LIBPATH=c:\tuxedo\lib
   NUMBER_OF_PROCESSORS=2
   ORACLE_HOME=c:\oracle\ora81
   OS=Windows_NT
   Os2LibPath=C:\WINNT\system32\os2\dll;
   Path=C:\Oracle\Ora81\bin;C:\Program
Files\Oracle\jre\1.1.7\bin;C:\WINNT\system32;C:\WINNT;C:\TUXEDO\bin
   PROCESSOR_ARCHITECTURE=x86
   PROCESSOR_IDENTIFIER=x86 Family 6 Model 7 Stepping 2, GenuineIntel
   PROCESSOR_LEVEL=6
   PROCESSOR_REVISION=0702
   TMCONTEXTS=1
   TUXCONFIG=c:\tuxedo\runtime\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: WORKGROUP
Network Version: 4.0
LanRoot: WORKGROUP
Logged On Users: 1
Current User (1): Administrator
  Logon Domain: CLIENT1
  Logon Server: CLIENT1

Transport: Nbf_HPTX1, 00-90-27-72-5C-15, VC's: 0, Wan: Wan
Transport: Nbf_HPTX2, 00-90-27-72-5B-E3, 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: 0
SMB's Received: 0
Paged Read Bytes Requested: 0
Non Paged Read Bytes Requested: 0
Cache Read Bytes Requested: 0
Network Read Bytes Requested: 0
Bytes Transmitted: 0
SMB's Transmitted: 0
Paged Read Bytes Requested: 0
Non Paged Read Bytes Requested: 0
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: 0
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: 0
Failed Sessions: 0
Reconnects: 0
Core Connects: 0
LM 2.0 Connects: 0
LM 2.x Connects: 0
Windows NT Connects: 0
Server Disconnects: 0
Hung Sessions: 0
Use Count: 0
Failed Use Count: 0
Current Commands: 0
Server File Opens: 439
Server Device Opens: 0
Server Jobs Queued: 0
Server Session Opens: 1
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: 143,659,209
Server Bytes Received: 11,581,125
Server Average Response Time: 0
Server Request Buffers Needed: 0
Server Big Buffers Needed: 0
```

Internet Information Server Registry Parameters

```
\registry\machine\system\currentcontrolset\services\inetinfo
    Parameters
        BandwidthLevel = REG_DWORD 0xffffffff
        ListenBackLog = REG_DWORD 0x00000019
        DisableMemoryCache = REG_DWORD 0x00000001
        MemoryCacheSize = REG_DWORD 0x00000000
        PoolThreadLimit = REG_DWORD 0x000000ff
        ObjectCacheTTL = REG_DWORD 0xffffffff
        Filter
            FilterType = REG_DWORD 0x00000000
            NumGrantSites = REG_DWORD 0x00000000
            NumDenySites = REG_DWORD 0x00000000
        MimeMap
            text/html,htm,,h =
            image/gif,gif,,g =
            image/jpeg,jpg,,: =
            text/plain,txt,,0 =
            text/html,html,,h =
            image/jpeg,jpeg,,: =
            image/jpeg,jpe,,: =
            image/bmp,bmp,,: =
            application/octet-stream,*,,5 =
            application/pdf,pdf,,5 =
            application/octet-stream,bin,,5 =
            application/oda,oda,,5 =
            application/zip,zip,,9 =
            application/rtf,rtf,,5 =
            application/postscript,ps,,5 =
            application/postscript,ai,,5 =
            application/postscript,eps,,5 =
            application/mac-binhex40,hqx,,4 =
            application/msword,doc,,5 =
            application/msword,dot,,5 =
            application/winhlp,hlp,,5 =
            video/mpeg,mpeg,,; =
            video/mpeg,mpg,,; =
            video/mpeg,mpe,,; =
            video/x-msvideo,avi,,< =
            video/quicktime,qt,,; =
            video/quicktime,mov,,; =
            video/x-sgi-movie,movie,,< =
            x-world/x-vrml,wrl,,5 =
            x-world/x-vrml,xaf,,5 =
            x-world/x-vrml,xof,,5 =
            x-world/x-vrml,flr,,5 =
            x-world/x-vrml,wrz,,5 =
            application/x-director,dcr,,5 =
            application/x-director,dir,,5 =
            application/x-director,dxr,,5 =
            image/cis-cod,cod,,5 =
            image/x-cmx,cmx,,5 =
            application/envoy,evy,,5 =
            application/x-msaccess,mdb,,5 =
            application/x-mscardfile,crd,,5 =
            application/x-msclip,clp,,5 =
            application/octet-stream,exe,,5 =
            application/x-msexcel,xla,,5 =
            application/x-msexcel,xlc,,5 =
            application/x-msexcel,xlm,,5 =
            application/x-msexcel,xls,,5 =
            application/x-msexcel,xlt,,5 =
            application/x-msexcel,xlw,,5 =
            application/x-msmediaview,m13,,5 =
            application/x-msmediaview,m14,,5 =
            application/x-msmoney,mny,,5 =
            application/x-mspowerpoint,ppt,,5 =
            application/x-msproject,mpp,,5 =
            application/x-mspublisher,pub,,5 =
            application/x-msterminal,trm,,5 =
            application/x-msworks,wks,,5 =
```

```
application/x-mswrite,wri,,5 =
application/x-msmetafile,wmf,,5 =
application/x-csh,csh,,5 =
application/x-dvi,dvi,,5 =
application/x-hdf,hdf,,5 =
application/x-latex,latex,,5 =
application/x-netcdf,nc,,5 =
application/x-netcdf,cdf,,5 =
application/x-sh,sh,,5 =
application/x-tcl,tcl,,5 =
application/x-tex,tex,,5 =
application/x-texinfo,texinfo,,5 =
application/x-texinfo,texi,,5 =
application/x-troff,t,,5 =
application/x-troff,tr,,5 =
application/x-troff,roff,,5 =
application/x-troff-man,man,,5 =
application/x-troff-me,me,,5 =
application/x-troff-ms,ms,,5 =
application/x-wais-source,src,,7 =
application/x-bcpio,bcpio,,5 =
application/x-cpio,cpio,,5 =
application/x-gtar,gtar,,9 =
application/x-shar,shar,,5 =
application/x-sv4cpio,sv4cpio,,5 =
application/x-sv4crc,sv4crc,,5 =
application/x-tar,tar,,5 =
application/x-ustar,ustar,,5 =
audio/basic,au,,< =
audio/basic,snd,,< =
audio/x-aiff,aif,,< =
audio/x-aiff,aiff,,< =
audio/x-aiff,aifc,,< =
audio/x-wav,wav,,< =
audio/x-pn-realaudio,ram,,< =
image/ief,ief,,: =
image/tiff,tiff,,: =
image/tiff,tif,,: =
image/x-cmu-raster,ras,,: =
image/x-portable-anymap,pnm,,: =
image/x-portable-bitmap,pbm,,: =
image/x-portable-graymap,pgm,,: =
image/x-portable-pixmap,ppm,,: =
image/x-rgb,rgb,,: =
image/x-xbitmap,xbm,,: =
image/x-xpixmap,xpm,,: =
image/x-xwindowdump,xwd,,: =
text/html,stm,,h =
text/plain,bas,,0 =
text/plain,c,,0 =
text/plain,h,,0 =
text/richtext,rtx,,0 =
text/tab-separated-values,tsv,,0 =
text/x-setext,etx,,0 =
application/x-perfmon,pmc,,5 =
application/x-perfmon,pma,,5 =
application/x-perfmon,pmr,,5 =
application/x-perfmon,pml,,5 =
application/x-perfmon,pmw,,5 =
```

```
    Performance
        Library = infoctrs.DLL
        Open = OpenINFOPerformanceData
        Close = CloseINFOPerformanceData
        Collect = CollectINFOPerformanceData
        Last Counter = REG_DWORD 0x00000756
        Last Help = REG_DWORD 0x00000757
        First Counter = REG_DWORD 0x00000738
        First Help = REG_DWORD 0x00000739
```

World Wide Web Server Registry Parameters

```
\registry\machine\system\currentcontrolset\services\w3svc [17 1]
    Type = REG_DWORD 0x00000020
    Start = REG_DWORD 0x00000003
    ErrorControl = REG_DWORD 0x00000000
    ImagePath = REG_EXPAND_SZ C:\WINNT\System32\inetsrv\inetinfo.exe
    DisplayName = World Wide Web Publishing Service
    DependOnService = REG_MULTI_SZ "RPCSS"  \
                                   "NTLMSSP"
    DependOnGroup = REG_MULTI_SZ
    ObjectName = LocalSystem
    Parameters
        MajorVersion = REG_DWORD 0x00000002
        MinorVersion = REG_DWORD 0x00000000
        AdminName = Administrator
        AdminEmail = Admin@corp.com
        MaxConnections = REG_DWORD 0x00002710
        LogType = REG_DWORD 0x00000000
        LogFileDirectory = REG_EXPAND_SZ %SystemRoot%\System32\LogFiles
        LogFileTruncateSize = REG_DWORD 0x01388000
        LogFilePeriod = REG_DWORD 0x00000001
        LogFileFormat = REG_DWORD 0x00000000
        LogSqlDataSource = HTTPLOG
        LogSqlTableName = Internetlog
        LogSqlUserName = InternetAdmin
        LogSqlPassword = sqllog
        Authorization = REG_DWORD 0x00000005
        AnonymousUserName = IUSR_CLIENT1
        Default Load File = Default.htm
        Dir Browse Control = REG_DWORD 0x4000001e
        CheckForWAISDB = REG_DWORD 0x00000000
        CacheExtensions = REG_DWORD 0x00000001
        GlobalExpire = REG_DWORD 0xffffffff
        ServerSideIncludesEnabled = REG_DWORD 0x00000001
        ServerSideIncludesExtension = .stm
        DebugFlags = REG_DWORD 0x00000008
        ScriptTimeout = REG_DWORD 0x00000384
        ConnectionTimeOut = REG_DWORD 0x00001c20
        InstallPath = C:\WINNT\System32\inetsrv
        SecurePort = REG_DWORD 0x000001bb
        Filter DLLs = C:\WINNT\System32\inetsrv\sspifilt.dll
        AccessDeniedMessage = Error: Access is Denied.
        NTAuthenticationProviders = NTLM
        ServerComment =
        ADCLaunch
```

```
            AdvancedDataFactory
            RDSServer.DataFactory
        Script Map
            .idc = C:\WINNT\System32\inetsrv\httpodbc.dll
        Virtual Roots
            /, = C:\InetPub\wwwroot,,5
            /Scripts, = C:\InetPub\scripts,,4
            /MSADC, = C:\Program Files\Common Files\System\MSADC,,5
            /iisadmin, = C:\WINNT\System32\inetsrv\iisadmin,,1
    Performance
        Library = w3ctrs.DLL
        Open = OpenW3PerformanceData
        Close = CloseW3PerformanceData
        Collect = CollectW3PerformanceData
        Last Counter = REG_DWORD 0x00000790
        Last Help = REG_DWORD 0x00000791
        First Counter = REG_DWORD 0x00000758
        First Help = REG_DWORD 0x00000759
    Security [17 1]
        Security = REG_BINARY 0x000000d8 0x80140001 0x000000c0 0x000000cc
0x00000014 0x00000034 0x00200002 0x00000001 0x00188002 0x000f01ff
0x00000101 0x01000000 0x00000000 0x00000220 0x008c0002 0x00000005
0x00180000 0x0002018d 0x00000101 \
                    0x01000000 0x00000000 0x00000000 0x001c0000 0x000201fd
0x00000201 0x05000000 0x00000020 0x00000223 0x00000000 0x001c0000
0x000f01ff 0x00000201 0x05000000 0x00000020 0x00000220 0x00000000
0x001c0000 0x000f01ff 0x00000201 \
                    0x05000000 0x00000020 0x00000225 0x00000000 0x00180000
0x000201fd 0x00000101 0x05000000 0x00000012 0x00000225 0x00000101
0x05000000 0x00000012 0x00000101 0x05000000 0x00000012
    Enum
        0 = Root\LEGACY_W3SVC\0000
        Count = REG_DWORD 0x00000001
        NextInstance = REG_DWORD 0x00000001
```

TPCC Application Settings

```
\registry\machine\software\unisys
    TPCC
        MAXTERMS = 10000
        DELIVERYTHREADS = 12
        DQSIZE = 3000
```

Appendix D - RTE Code

Admin Environment

```
if '%1'==' goto usage
if '%1'==' goto usage
if '%2'==' goto usage
if '%3'==' goto usage

:paramok

set WEBCHECKWIDS=1
set WEBDIAGLEVEL=4
set WEBEVENTLOG=0
set WEBEVENTHOST=
set WEBCHECKLEVEL=2

c:\webdriver\webadmin.exe -cweb%1.cfg -m%2 -d%3 -s160
if %ERRORLEVEL% NEQ 0 pause

goto end

:usage
@ECHO You must supply the following parameters:
@ECHO "webnnn.cmd <cfg file suffix> <min driver #> <max driver #>"
pause

:end
```

Profiles used for Performance Run

Web3300.cfg

```
//
//
//  Common Driver Configuration
//
INITBASEPORT 4300
INITSYNCMAX 4
INITPAUSE 1
INITRSCALE 250
INITTSCALE 120
INITRWID 1,3300
INITFIXEDWID 1
INITCCLAST 87
INITCCID 511
INITCITEMID 4093
//
//  Configuration Driver 1 - RTE1
//
1 INITIPADDR 192.158.100.31
1 INITIISADDR 192.158.11.1
1 INITIISPORT 80
1 INITBROWSERS 910
1 INITMYWID 1,91
//
//  Configuration Driver 2 - RTE2
//
2 INITIPADDR 192.158.100.32
2 INITIISADDR 192.158.12.1
2 INITIISPORT 80
2 INITBROWSERS 910
2 INITMYWID 92,182
//
//  Configuration Driver 3 - RTE3
//
3 INITIPADDR 192.158.100.33
3 INITIISADDR 192.158.13.1
3 INITIISPORT 80
3 INITBROWSERS 910
3 INITMYWID 183,273
//
//  Configuration Driver 4 - RTE4
//
4 INITIPADDR 192.158.100.34
4 INITIISADDR 192.158.14.1
4 INITIISPORT 80
4 INITBROWSERS 920
4 INITMYWID 274,365
//
//  Configuration Driver 5 - RTE5
//
5 INITIPADDR 192.158.100.35
5 INITIISADDR 192.158.15.1
5 INITIISPORT 80
5 INITBROWSERS 920
5 INITMYWID 366,457
//
//  Configuration Driver 6 - RTE6
//
6 INITIPADDR 192.158.100.36
6 INITIISADDR 192.158.16.1
6 INITIISPORT 80
6 INITBROWSERS 920
6 INITMYWID 458,549
//
//  Configuration Driver 7 - RTE7
//
7 INITIPADDR 192.158.100.37
7 INITIISADDR 192.158.17.1
7 INITIISPORT 80
7 INITBROWSERS 920
7 INITMYWID 550,641
//
```

```
//  Configuration Driver 8 - RTE8
//
8 INITIPADDR 192.158.100.38
8 INITIISADDR 192.158.18.1
8 INITIISPORT 80
8 INITBROWSERS 920
8 INITMYWID 642,733
//
//  Configuration Driver 9 - RTE9
//
9 INITIPADDR 192.158.100.39
9 INITIISADDR 192.158.19.1
9 INITIISPORT 80
9 INITBROWSERS 920
9 INITMYWID 734,825
//
//  Configuration Driver 10 - RTE1
//
10 INITIPADDR 192.158.100.31
10 INITIISADDR 192.158.21.2
10 INITIISPORT 80
10 INITBROWSERS 920
10 INITMYWID 826,917
//
//  Configuration Driver 11 - RTE2
//
11 INITIPADDR 192.158.100.32
11 INITIISADDR 192.158.22.2
11 INITIISPORT 80
11 INITBROWSERS 920
11 INITMYWID 918,1009
//
//  Configuration Driver 12 - RTE3
//
12 INITIPADDR 192.158.100.33
12 INITIISADDR 192.158.23.2
12 INITIISPORT 80
12 INITBROWSERS 920
12 INITMYWID 1010,1101
//
//  Configuration Driver 13 - RTE4
//
13 INITIPADDR 192.158.100.34
13 INITIISADDR 192.158.24.2
13 INITIISPORT 80
13 INITBROWSERS 910
13 INITMYWID 1102,1192
//
//  Configuration Driver 14 - RTE5
//
14 INITIPADDR 192.158.100.35
14 INITIISADDR 192.158.25.2
14 INITIISPORT 80
14 INITBROWSERS 910
14 INITMYWID 1193,1283
//
//  Configuration Driver 15 - RTE6
//
15 INITIPADDR 192.158.100.36
15 INITIISADDR 192.158.26.2
15 INITIISPORT 80
15 INITBROWSERS 910
15 INITMYWID 1284,1374
//
//  Configuration Driver 16 - RTE7
//
16 INITIPADDR 192.158.100.37
16 INITIISADDR 192.158.27.2
16 INITIISPORT 80
16 INITBROWSERS 920
16 INITMYWID 1375,1466
//
//  Configuration Driver 17 - RTE8
//
17 INITIPADDR 192.158.100.38
17 INITIISADDR 192.158.28.2
17 INITIISPORT 80
17 INITBROWSERS 920
17 INITMYWID 1467,1558
//
//  Configuration Driver 18 - RTE9
//
18 INITIPADDR 192.158.100.39
18 INITIISADDR 192.158.29.2
18 INITIISPORT 80
18 INITBROWSERS 920
18 INITMYWID 1559,1650
//
//  Configuration Driver 19 - RTE1
//
19 INITIPADDR 192.158.100.31
19 INITIISADDR 192.158.31.3
19 INITIISPORT 80
19 INITBROWSERS 920
19 INITMYWID 1651,1742
//
//  Configuration Driver 20 - RTE2
//
20 INITIPADDR 192.158.100.32
20 INITIISADDR 192.158.32.3
20 INITIISPORT 80
20 INITBROWSERS 920
20 INITMYWID 1743,1834
//
//  Configuration Driver 21 - RTE3
//
21 INITIPADDR 192.158.100.33
21 INITIISADDR 192.158.33.3
21 INITIISPORT 80
21 INITBROWSERS 920
21 INITMYWID 1835,1926
//
//  Configuration Driver 22 - RTE4
//
22 INITIPADDR 192.158.100.34
22 INITIISADDR 192.158.34.3
22 INITIISPORT 80
22 INITBROWSERS 920
```

```
22 INITMYWID 1927,2018
//
//  Configuration Driver 23 - RTE5
//
23 INITIPADDR 192.158.100.35
23 INITIISADDR 192.158.35.3
23 INITIISPORT 80
23 INITBROWSERS 920
23 INITMYWID 2019,2110
//
//  Configuration Driver 24 - RTE6
//
24 INITIPADDR 192.158.100.36
24 INITIISADDR 192.158.36.3
24 INITIISPORT 80
24 INITBROWSERS 920
24 INITMYWID 2111,2202
//
//  Configuration Driver 25 - RTE7
//
25 INITIPADDR 192.158.100.37
25 INITIISADDR 192.158.37.3
25 INITIISPORT 80
25 INITBROWSERS 910
25 INITMYWID 2203,2293
//
//  Configuration Driver 26 - RTE8
//
26 INITIPADDR 192.158.100.38
26 INITIISADDR 192.158.38.3
26 INITIISPORT 80
26 INITBROWSERS 910
26 INITMYWID 2294,2384
//
//  Configuration Driver 27 - RTE9
//
27 INITIPADDR 192.158.100.39
27 INITIISADDR 192.158.39.3
27 INITIISPORT 80
27 INITBROWSERS 910
27 INITMYWID 2385,2475
//
//  Configuration Driver 28 - RTE1
//
28 INITIPADDR 192.158.100.31
28 INITIISADDR 192.158.41.4
28 INITIISPORT 80
28 INITBROWSERS 910
28 INITMYWID 2476,2566
//
//  Configuration Driver 29 - RTE2
//
29 INITIPADDR 192.158.100.32
29 INITIISADDR 192.158.42.4
29 INITIISPORT 80
29 INITBROWSERS 910
29 INITMYWID 2567,2657
//
//  Configuration Driver 30 - RTE3
//
30 INITIPADDR 192.158.100.33
30 INITIISADDR 192.158.43.4
30 INITIISPORT 80
30 INITBROWSERS 910
30 INITMYWID 2658,2748
//
//  Configuration Driver 31 - RTE4
//
31 INITIPADDR 192.158.100.34
31 INITIISADDR 192.158.44.4
31 INITIISPORT 80
31 INITBROWSERS 920
31 INITMYWID 2749,2840
//
//  Configuration Driver 32 - RTE5
//
32 INITIPADDR 192.158.100.35
32 INITIISADDR 192.158.45.4
32 INITIISPORT 80
32 INITBROWSERS 920
32 INITMYWID 2841,2932
//
//  Configuration Driver 33 - RTE6
//
33 INITIPADDR 192.158.100.36
33 INITIISADDR 192.158.46.4
33 INITIISPORT 80
33 INITBROWSERS 920
33 INITMYWID 2933,3024
//
//  Configuration Driver 34 - RTE7
//
34 INITIPADDR 192.158.100.37
34 INITIISADDR 192.158.47.4
34 INITIISPORT 80
34 INITBROWSERS 920
34 INITMYWID 3025,3116
//
//  Configuration Driver 35 - RTE8
//
35 INITIPADDR 192.158.100.38
35 INITIISADDR 192.158.48.4
35 INITIISPORT 80
35 INITBROWSERS 920
35 INITMYWID 3117,3208
//
//  Configuration Driver 36 - RTE9
//
36 INITIPADDR 192.158.100.39
36 INITIISADDR 192.158.49.4
36 INITIISPORT 80
36 INITBROWSERS 920
36 INITMYWID 3209,3300
//
//
```

Driver Environment

```
if '%1'==' goto usage

:paramok

set WEBDRIVERNO=%1
set WEBADMBASEPORT=4300
set WEBDIAGLEVEL=2
set WEBEVENTLOG=1
set WEBEVENTHOST=
set WEBLOGLEVEL=1
set WEBSINGLETRAN=0
set WEBTPCCAUDIT=0
set WEBRTFUDGETM=110
set WEBNEWORDERPROB=4484
set WEBPAYMENTPROB=4307
set WEBORDERSTATUSPROB=403
set WEBDELIVERYPROB=403
set WEBSTOCKLEVELPROB=403
set WEBTTNEWORDER=12030
set WEBTTPAYMENT=12030
set WEBTTDELIVERY=5060
set WEBTTORDERSTATUS=10070
set WEBTTSTOCKLEVEL=5060

c:\webdriver\webdriver.exe

goto end

:usage
@ECHO  You must supply the following parameters:
@ECHO  "webdriver.cmd <driver number>"
pause

:end
exit
```

Appendix E - Disk Storage

TPC-C 180-day Disk Space Calculations

All numbers are in Blocks (2KB) unless otherwise specified

Warehouses	3,300	**tpmC**	41085.43	**tpmC / W**	12.45	
Table	**Rows**	**Data**	**Index**	**5% Space**	**8H Space**	**Total Space**
WAREHOUSE	3,500	3,511	35	177		3,723
DISTRICT	35,000	35,023	430	1,773		37,226
ITEM	100,000	6,667	871	377		7,915
CUSTOMER	105,000,000	52,500,001	4,001,724	2,825,086		59,326,811
STOCK	350,000,000	70,000,009	4,147,403	3,707,371		77,854,783
NEW_ORDER	31,500,000	276,555		13,828		290,383
HISTORY	105,000,000	3,170,719		158,536	599,895	3,929,150
ORDERS	105,000,000	2,056,071	3,314,895	268,548	399,930	6,039,444
ORDER_LINE	1,050,000,000	41,574,400		2,078,720	7,998,599	51,651,719
Totals		169,622,956	11,465,358	9,054,416	8,998,424	199,141,153
TABLESPACE	**Blocks**	**Used**	**Reqd. with 5%**	**Reqd. with 8H**	**Overhead**	**Oversize**
CUST	56,832,000	52,500,001	55,125,001	55,125,001	551,250	1,155,749
HIST	12,800,000	3,170,719	3,329,255	3,929,150	39,291	8,831,559
ICUST2	18,432,000	2,582,866	2,712,009	2,712,009	27,120	15,692,871
IORD1	7,680,000	1,244,121	1,306,327	1,306,327	13,063	6,360,610
IORD2	6,144,000	2,070,774	2,174,313	2,174,313	21,743	3,947,944
MISC*	3,840,000	46,072	48,376	48,376	484	3,791,140
NORD	18,432,000	276,555	290,383	290,383	2,904	18,138,713
ORDL	81,408,000	41,574,400	43,653,120	51,651,719	516,517	29,239,764
ORDR	3,840,000	2,056,071	2,158,875	2,558,805	25,588	1,255,607
ROLL	1,792,000	1,792,000	1,792,000	1,792,000	17,920	0
STOCK	84,480,000	73,500,009	77,175,009	77,175,009	771,750	6,533,241
SYSTEM	1,536,000	1,536,000	1,536,000	1,536,000	15,360	0
UNUSED**	36,864,000	5,566,726	5,845,062	5,845,062	58,451	30,960,487
Totals	**334,080,000**	**187,916,314**	**197,145,730**	**206,144,154**	**2,061,442**	125,907,685
Dynamic Space	**46,801,190**	Sum of History, Order and Order-line (Initial Used space)				
Static Space	145,402,981	(Data+Index+5%) - Dynamic Space				
Free Space	15,968,144	Total Blocks - Dynamic Space - Static Space - Overhead				
Daily Growth	8,998,423.74	Actual growth of History, Order and Order line				
Daily Spread	**(12,916,238.88)**	Freespace (in dynamic tables) - 1.5*Daily-Growth				
180 Day	1,765,119,255	Static-space + 180*(Daily-Growth+Daily-Spread)				
180 Day GB	3,367	(180-Day * Block Size of 2048) / (1024*1024*1024)				
Log used	144335499	log file used during an entire run				
Total N_O Txn	5049462	Total count of NEW_ORDER transactions during the entire run				
Log per N_O Txn	28.58433215	Number of blocks per NEW_ORDER transaction				
8 Hour Log (GB)	268.80	(Log blocks per N_O * Redo Block Size 512 * 60 * tpmC * 8)/(1024*1024*1024)				
Space Usage	**GB Reqd.**	**Disk Size(GB)**	**Disks Priced**	**GB Priced**		
180 day space	3,366.70	16.54	300	4,961.78		
Logs (with mirror)	537.60	14.89	40	595.41		
OS, Oracle		8.54	1	8.54		
Total Space				5565.73		

* = DISTRICT, IITEM, ITEM, WAREHOUSE

** = ICUSTOMER, IDISTRICT, ISTOCK, IWAREHOUSE

TPC-C 180-Day Table Growth Rates for 8 Hours				tpmC	41,085.43
Tables	**Initial (KB)**	**Final (KB)**	**Change (KB)**	**Blks/N-O**	**8-Hr Blocks**
History	9,501,216	9,347,616	153,600	0.0304191	599,895
Orders	1,689,570	1,587,170	102,400	0.0203	399,930
Order_line	38,041,540	35,993,540	2,048,000	0.4056	7,998,599
Log Blocks Used	883	144,336,382	144,335,499	**28.5843**	563,711,797
SUM(d_next_o_id)	109,311,470	114,360,932	5,049,462		

Appendix F - Third-Party Price Quotations

Western Micro Technology **12/10/99**
(800)937-8446

Quoted to: Jerry Buggert/Unisys for TPC.org
Prepared by: Tony Jacobs

Qty.	Description	Style	Price	Extended Price
1	SYS: e-@ction ES2085R, w/ CDRom, 0 Proc, 0MB	ESR208151-GZU	$13,997	$13,997
8	PROC: 550MHz P-III Xeon /2MB Cache & VRMs	XEO3550-2MB	$5,893	$47,144
1	BRD: Processor Mezzanine Board, 0 Proc.	ESR81-MEZ	$1,179	$1,179
32	MEM: 512 MB Memory, SDRAM, Buf 6ns	DIM6168-512	$1,989	$63,648
1	BRD: Memory Carrier Board, 0 Mem.	ESR81-MCB	$737	$737
2	MEM: Cache Coherency Filter, 4x SRAM	ESR81-CC4	$921	$1,842
6	CTRL: Fiber Channel HBA, 32/64-bit PCI	PCI1100-FC	$1,175	$7,050
1	DISK: 9GB, 10K SCSI LVD, SCA	HDL91102-CX1	$516	$516
1	ACC: HBA, SCSI w/ VHD Connector	PCI300-SUW	$249	$249
1	ETHERNET: 10/100Mbit/sec, PCI 32-bit	ETH1010052-PCI	$100	$100
1	MONITOR: 15-inch Color	EVG2100-P	$221	$221
	Server Total			**$136,683**
185	DISK: 18GB Drive, 10Krpm, Fiber Channel + 10% spares	ESM18201-F96	$1,250	$231,250
6	DPE: 2 RAID Cntrls, 2x256MB, 10 18GB Disk, BBU	ESM7700-A10	$36,611	$219,666
28	CAB: DAE Expn Enclsr, w/ 4 18GB Disk	ESM700-A04	$8,406	$235,368
6	MEM: 256MB DPE Cache Expansion SIMM	ESM7700-44M	$1,479	$8,874
6	CBL: FC, HSSDC->DB9 Conn's, 10m	CBL136-10	$165	$990
6	CAB: Rackmount Kit for DPE	ESM7700-PRM	$84	$504
28	CAB: Rackmount Kit for DAE	ESM700-DRM	$84	$2,352
24	PWR: Distribution Kit, 220V	SFR220-PWR	$42	$1,008
6	CAB: 36U x 19" x 34" Cabinet, Open	RM361934-OFT	$884	$5,304
6	DOOR: 36U x 19", Rear	RM3619-RDR	$277	$1,662
6	PNL: 36U x 34" Side Skins, L&R	RM3634-SDS	$221	$1,326
	Storage Total			**$708,304**
4	SYS: NetServer LC3, w/1 500MHz Proc., 0MB Mem	D7127-AV	$2,018	$8,072
4	PROC:1x500MHz Pentium III/512KB Cache UPG	D7130-AV	$1,300	$5,200
16	MEM: 256 MB SDRAM Memory Upgrade	D6099-AV	$655	$10,480
4	DISK: 9GB SCSI 3.5 Internal	D4911-AV	$338	$1,352
8	ETHERNET: 10/100TX Mbit/sec, PCI 32-bit	D5013-AV	$68	$544
1	SYS: e-@ction ES2043, w/CD-ROM, 0 Proc, 0MB	ES204131-GCU	$3,536	$3,536
4	PROC: 500MZ/1MB PIII Xeon/1MB Cache	XEO3500-1MB	$3,536	$14,144
4	ACC: Voltage Reg Mod	XEO24001-VRM	$44	$176
2	ETHERNET: 10/100Mbit/sec, PCI 32-bit	ETH1010052-PCI	$100	$200
8	MEM: 128MB 50ns DIMMs	DIM5072-128	$398	$3,184
1	DISK: 9GB, 10K SCSI LVD, SCA	HDL91102-CX1	$516	$516
1	ACC: HBA, SCSI w/ VHD Connector	PCI300-SUW	$249	$249
5	MONITOR: 15-inch Color	EVG2100-P	$221	$1,105
	Client Total			**$48,758**
1	O/S: UnixWare 7 Data Center Edition, Ver 7.1.1, 150 users	UXW71-DCL	$7,366	$7,366
1	O/S: UnixWare 7 Media Kit, Version 7.1.1	UNX971-PSK	$277	$277
1	O/S: UnixWare 7 Business Edition, Version7.1.1, 5 users	UXW71-BZL	$1,030	$1,030
3	O/S: Proc Upgrade, UnixWare 7	UNX71-PU	$553	$1,659
1	O/S: UnixWare 7 Media Kit, Version 7.1.1	UNX971-PSK	$277	$277
	Software Total			**$10,609**
	Server, Storage, Client and Software Total			**$904,354**
	Discount based on total dollar volume			**($90,435)**
Quote Total				**$813,919**
Quote valid for 90 days.				
Disks come with return to factory, 5 year warranty, 7 day replenishment				

THE ECOMMERCE TRANSACTION PLATFORM

December 8, 1999

Mr. Jerrold Buggert
Director, Systems Analysis, Modeling, Measurement
Unisys Corporation
25725 Jeronimo Road
Mission Viejo, CA 92691
Fax (949) 465-2552

Dear Mr. Buggert:

Per your request I am enclosing the pricing information regarding TUXEDO 6.5 and 6.4 that you requested. This pricing applies to Tuxedo 6.1, 6.2, 6.3, 6.4 and 6.5. Please note that Tuxedo 6.5 is our most recent version of Tuxedo but that all 6.x releases are generally available.
Core functionality services pricing is appropriate for your activities. As per the table below Unisys Intel-based server systems are classified as either a Tier 1, Tier 2 or Tier 3 server depending on the CPU capacity of the system. The e-@ction ES2085R Enterprise Server (newly renamed, 8-way 550MHz Pentium III Xeon technology) is a Tier 3 system, the e-@ction ES2043 (4-way 550MHz Pentium III Xeon)) UnixWare client is a tier 2 system and the four NetServer LC3 NT 4.0 clients (2-way 500MHz Pentium III technology) are tier 1 systems. This quote is valid for 90 days from the date of this letter.

Tuxedo Core Functionality Services (CFS) Program Product Pricing and Description

TUX-CFS provides a basic level of middleware support for distributed computing, and is best used by organizations with substantial resources and knowledge for advanced distributed computing implementations.

TUX-CFS prices are server only and are based on the overall performance characteristics of the server and uses the same five tier computer classification as TUXEDO 6.x. Prices range from $3,000 for Tier 1 to $250,000 for Tier 5. Under this pricing option EVERY system running TUX-CFS at the user site must have a TUXEDO license installed and pay the appropriate per server license fees.

Very Truly Yours,

Lewis D. Brentano,
Director, Market Planning

BEA Tux/CFS Unlimited User License Fees Per Server

Unlimited User License fees per server	Number of Users	Dollar Amount	Maintenance (5 x 8) per year	Maintenance (7 x 24) per year
Tier 1 -- PC Servers with 1 or 2 CPUs, entry level RISC Uni-processor workstations and servers (Class 1and Class 2)	Unlimited	$3,000.00	$480.00	$690.00
Tier 2 - PC Servers with 3 or 4 CPUs, Midrange RISC Uni-processor servers and workstations (class 3)	Unlimited	$12,000.00	$1,920.00	$2,760.00
Tier 3 - Midrange Multiprocessors, up to 8 CPUs per system capacity (Class 4 and 5)	Unlimited	$30,000.00	$4,800.00	$6,900.00
Tier 4 - Large (more than 8, less than 32 CPUs) and Mainframe Systems (Class 6)	Unlimited	$100,000.00	$16,000.00	$23,000.00
Tier 5 - Massively Parallel Systems, > 32 processors	Unlimited	$250,000.00	$40,000.00	$57,500.00

Intel based server tier classifications:

		Tier 1	Tier 1	Tier 2	Tier 3
Platform	**Operating System**				
Intel Pentium/ Pentium Pro PCs	Interactive R3.2 ESIX SVR 4.0 SCO UNIX 3.2.2 and 3.2.4 SCO ODT 2.x,3.x Solaris x86 2.X UnixWare, Windows NT 3.5/4.0	All 386/486 PCs are Class 1	ALL Pentium PCs with 1 or 2 CPUs capacity are Tier 1	ALL Pentium PCs with 3 or 4 CPUs capacity are Tier 2	ALL Pentium PCs with 5,6,7, or 8 CPUs Capacity are Tier 3

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
http://www.microsoft.com/

Microsoft

December 9, 1999

Rick Freeman
Unisys Corporation
25725 Jeronimo Road
Mission Viejo, CA 92691

Dear Rick,

Here is the information you requested regarding US pricing of certain Microsoft products:

Microsoft Windows NT Server 4.0, including 5 CALs (4 copies @ $809)	$3,236
Visual C++ Professional 6.0	$549
5-yr maintenance for above software @ $2095/yr	$10,475

This quote is valid for the next 90 days. Please let me know if I can be of any further assistance.

Best regards,

Mark Hassall
Product Manager, Microsoft Windows NT Server

** TOTAL PAGE.02 **

NETLUX

14180 Live Oak Ave., Unit E
Baldwin Park, Ca. 91760

1-800-789-1780
Phone#626-851-9737
Fax #626-851-9837

December 8, 1999

Rick Freeman
Unisys Corporation
25725 Jeronimo Road
Mission Viejo, CA 92691
Fax: (949) 380-5539
cc: (949) 380-5344

Quotation

Quantity	Part No.	Description	Unit Price
1-50	NX-SW8	NETLUX 8-port 10/100Mbps FAST Ethernet Switch	$179.00

Terms and Conditions:
FOB Origin
Quote Valid for 90 days
5 Year Warranty

Sincerely,
Martin Parry
NETLUX

Quotation

NetCruiser Technologies, Inc.
1347 Stelton Road
Piscataway, NJ 08854
United State

ComReady.com

Quote Number: 5484

Quote Date: Dec 9, 1999

Page: 1

Quoted to:

END USER OUT STATE---Yahoo
Rick Freeman
Unisys Corp.
25725 Jeronimo Rd.
Mission Viejo, CA 92691

Customer ID	Good Thru	Payment Terms	Sales Rep
END8888	1/8/00	Prepaid	NCTI112

Quantity	Item	Description	Unit Price	Extension
10.00	HB-1009DX/P	8 PORT 10 Base-T Hub, 8 STP/UTP, 1 BNC Plastic Price Good for 90 Days Lifetime Warranty Unit Price is valid even if quantity greater then 10 Pieces. Freight charge varies depending on quantity and shipping location.	24.00	240.00
			Subtotal	240.00
			Sales Tax	
			Freight	
			Total	240.00