NEC

TPC Benchmark™ C
Full Disclosure Report

NEC Express5800/1320Xd (32 SMP)

with Oracle® Database 10g Enterprise Edition
and
SUSE LINUX Enterprise Server 9 for Itanium Processors

Second Edition
August 30, 2004

NEC, the Sponsor of this benchmark test, believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. The Sponsor assumes 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, the Sponsor provides no warranty of the pricing information in this document.
Benchmark results are highly dependent upon workload, specific application requirements, and system 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 were obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. NEC does 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.

Copyright 2004 NEC Corporation.
All rights reserved.
Permission is hereby granted to reproduce this document in whole or in part provided the copyright notice printed above is set forth in full text or on the title page of each item reproduced.

Printed in USA, 2004

NEC and Express5800 are registered trademarks of NEC Corporation.

ORACLE, SQL*DBA, SQL*Loader, SQL*Net, SQL*Plus, Oracle 10g, Pro *C, and PL/SQL are registered trademarks of Oracle Corporation.

SUSE is a registered trademark of SUSE LINUX AG, a Novell company.

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

BEA and Tuxedo are registered trademarks of BEA Systems, Inc.
Intel, Itanium and Xeon are registered trademarks and trademark of Intel Corporation.

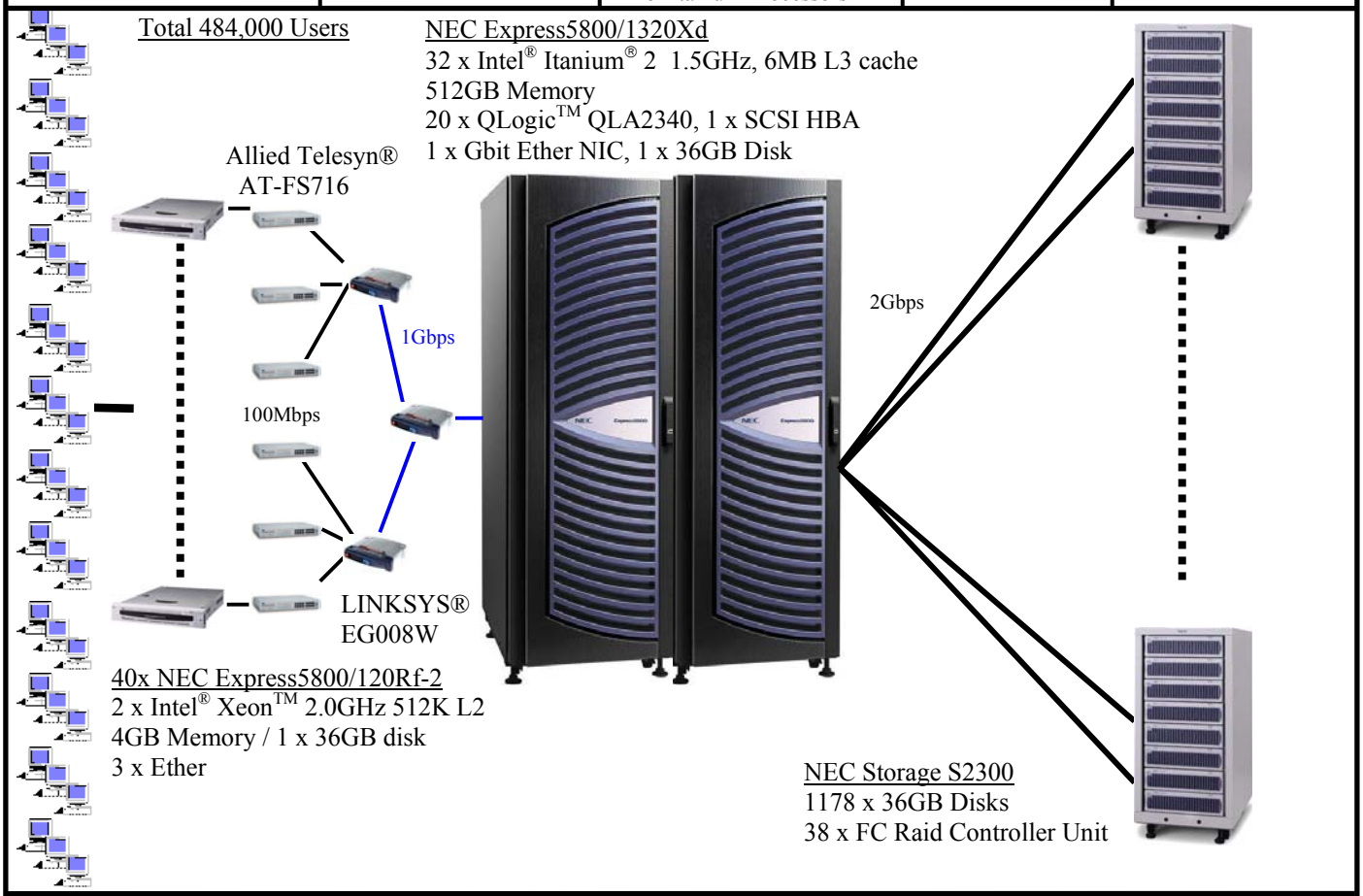
Linux is a registered trademark of Linus Torvalds.

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

NEC	NEC Express5800/1320Xd C/S with Express5800/120Rf-2	TPC-C Rev. 5.2 Reported Date Apr 6, 2004 Revised Date Aug 30, 2004

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$4,128,185	609,467 tpmC	$6.78 per tpmC	Oct 5, 2004

Processors	Database Manager	Operating System	Other Software	Number of Users
32 Intel® Itanium® 2 1.5GHz for Server	Oracle® Database 10g Enterprise Edition	SUSE LINUX Enterprise Server 9 for Itanium Processors	BEA Tuxedo 8.1	484,000



System Component	Server		Each Client	
Processors **Cache**	32	Intel® Itanium® 2 1.5GHz 6MB L3 Cache	2	Intel® Xeon™ 2.0GHz 512KB L2 Cache
Memory		512GB		4GB
Disk Controllers	20 1	QLogic™ QLA2340 SCSI HBA	1	On-board SCSI
Disk Drives	1 1178	36GB 36GB	1	36GB
Total Storage		42,444GB		36GB
Others	1 1	DVD-ROM Drive 1Gbps Ether NIC	1 2 1	CD-ROM Drive On-board Ether controller 100Mbps Ether NIC

NEC	NEC Express5800/1320Xd C/S with Express5800/120Rf-2	TPC-C REV 5.2 Report Date: 4/6/2004

Description	Part Number	Third Party Brand	Pricing	Unit Price	Qty		Extended Price	3-yr Mnt. Price
Server Hardware								
Express5800/1320Xd system	850200703	NEC	1	1,407,745	1		1,407,745	337,859
Express5800/1320Xd base system					1			
NEC 1160Xd/1320Xd Cell with 4 Itanium 2 CPUs - 1.5GHz/6MB Cache					8			
8GB (4x2GB ECC SDRAM DIMM) memory kit					64			
Additional Server Partition					3			
36GB Ultra 320/m SCSI HDD (1") 10000RPM					1			
Expansion Cabinet					1			
10/100/1000Mb Either card (3COM 3C996B-T)					1			
Ultra160 SCSI card (QLA12160/66)					1			
NEC AccuSync50 (15" monitor, +2 spares)	AS50	NEC	5	130	3		389	-
FC HBA QLA2340 (+2 spares)	QLA2340-CK	QLogic	5	1,189	22		26,169	-
Fast Ether Cable 25' RJ45-RJ45 (+10% spares)	N002-025-GY	Tripp Lite	5	7	54		395	
						Subtotal	1,434,698	337,859
Disk Subsystem								
NEC Storage S2300 Basic Unit	850161002	NEC	1	46,296	38		1,759,248	-
2 RAID_Cntrl, 1GB Cache, 1DE, 3 x 36GB Drive								
1GB(2 x 512MB) PC133 SDRAM memory	062-02301-000	NEC	1	1,999	76		151,924	-
S2300 DE (+10% spares)	062-02302-000	NEC	1	6,999	46		321,954	-
36GB 15K rpm FC HDD (+10% spares)	062-02303-000	NEC	1	1,099	1171		1,286,929	-
3 yr Telephone Upgrade Hdw & Storage Manager SW Support; 7days/ 24hours per day	IS-THSU-1095-0724	NEC	1	1499	38			56,962
42U Rackframe	050-01790-000	NEC	1	1,799	9		16,191	-
FC Cable 10M LC-LC (+10% spares)	062-02304-000	NEC	1	200	42		8,400	-
						Subtotal	3,544,646	56,962
Server Software								
3 years Maintenance for SUSE LINUX Enterprise Server 9 for Itanium Processors (per CPU)	TBD	SUSE	3	246.68	32		7,894	
3 years Premium Support for SUSE LINUX Enterprise Server 9 for Itanium Processors (per CPU)	TBD	SUSE	3	369.56	32			11,826
Oracle Database 10g Enterprise Edition for 3 years, Per Processor, Unlimited Users		Oracle	2	20,000	32		640,000	
Oracle Database Server Support Package for 3 years		Oracle	2	6,000	1			6,000
						Subtotal	647,894	17,826
Client Hardware								
NEC Express5800/120Rf-2								
Base System with 1 x Xeon Processor 2.0GHz/512KB	850166001	NEC	1	3,499	40		139,960	-
1 x Xeon Processor 2.0GHz/512KB BTO Option,	062-02259-000	NEC	1	499	40		19,960	-
2GB(2 x 1024MB) DDR266 SDRAM memory,	062-02436-000	NEC	1	1,200	80		96,000	-
1 x 36GB 10K rpm HDD,	062-02012-000	NEC	1	459	40		18,360	-
CD-ROM, 2 x On-board LAN, KB/MS	Included	NEC	1		40			
3 years of warranty service to 4-hour response, 7x24	DE-0000-1095-7244	NEC	1	1,599	40		-	63,960
NEC AccuSync50 (15" monitor, +2 spares)	AS50	NEC	5	130	3		389	-
42U Rackframe	050-01790-000	NEC	1	1,799	3		5,397	-
Intel PRO/100+ Adapter, 20-pack (+10% spares)	PILA8460BNIPK20	Intel	5	550	3		1,650	-
Fast Ether Cable 25' RJ45-RJ45 (+10% spares)	N010-010-GY	Tripp Lite	5	6	88		570	-
						Subtotal	282,286	63,960
Client Software								
3 years Maintenance for SUSE LINUX Standard Server 8	2133-1-MFJ-3Y-2-VPO	SUSE	3	704	40		28,166	
3 years Premium Support for SUSE LINUX Standard Server 8	2133-S-ST-3Y-2-VPO	SUSE	3	1,054	40			42,159
BEA Tuxedo 8.1 Core functionality Services		BEA	4	1,140	40		45,600	30,240
						Subtotal	73,766	72,399
User Connectivity								
Linksys Gigabit 8-port Workgroup Switch (+2 spares)	EG008W	Linksys	5	159	5		797	
16-port Fast Ether Switch (+2 spares)	AT-FS716	Allied Telesyn	5	79	8		632	
						Subtotal	1,429	0
						TOTAL	5,984,718	549,006
Oracle Mandatory E-Business Discount							-129,200	
NEC brand total (Pricing 1-NEC)							5,232,068	458,781
NEC brand Large Purchase Cash Prepay Discount (-40%)							-2,092,827	-183,512

Notes:

Pricing: 1-NEC 2-Oracle 3-SUSE 4-BEA 5-CDW

3-Yr. Cost of Ownership:	**$4,128,185**
tpmC Rating:	**609467**
$ / tpmC:	**$6.78**

Results and methodology audited by Francois Raab of InfoSizing, Inc. (www.sizing.com)

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

Numerical Quantities Summary

MQTh, Computed Maximum Qualified Throughput			609,467 tpmC
Response Times(in seconds)	**Average**	**Maximum**	**90%**
New-Order	0.21	53.44	0.27
Payment	0.20	45.83	0.25
Delivery(interactive portion)	0.12	56.3	0.11
Stock-Level	0.18	57.49	0.25
Order-status	0.20	41.71	0.27
Delivery(deferred portion)	0.09	57.53	0.16
Menu	0.11	0.69	0.11
Response time delay added for emulated components			0.1
Transaction Mix , in percent of total transaction			
New-Order			44.81%
Payment			43.07%
Delivery			4.04%
Stock-Level			4.04%
Order-Status			4.04%

Keying/Think Times (in seconds)	**Average**		**Min**		**Max**	
New-Order	18.01	12.07	18.01	0.00	18.01	120.7
Payment	3.01	12.07	3.01	0.00	3.01	120.7
Delivery	2.01	5.07	2.01	0.00	2.01	50.7
Stock-Level	2.01	5.08	2.01	0.00	2.01	50.7
Order-Status	2.01	10.07	2.01	0.00	2.01	100.7

Test Duration	
Ramp-up time	4068 seconds
Measurement interval	7260 seconds
Number of checkpoints	4
Checkpoint interval	1726 seconds
Number of transactions (all types) completed in measurement interval	164,615,923

ABSTRACT 8
TPC BENCHMARK™ C METRICS 8
STANDARD AND EXECUTIVE SUMMARY STATEMENTS 8
AUDITOR 8
PREFACE 9
TPC BENCHMARK™ C OVERVIEW 9
DOCUMENT STRUCTURE 9
GENERAL ITEMS 11
ORDER AND TITLES 11
SUMMARY STATEMENT 11
NUMERICAL QUANTITIES SUMMARY 11
APPLICATION PROGRAM 11
SPONSOR 11
PARAMETERS AND OPTIONS 12
CONFIGURATION DIAGRAMS 12
MEASURED CONFIGURATION 13
PRICED SYSTEM CONFIGURATION 14
CLAUSE 1 : LOGICAL DATABASE DESIGN AND RELATED ITEMS 15
TABLE DEFINITIONS 15
TABLE ORGANIZATION 15
INSERT AND DELETE OPERATIONS 15
DISCLOSURE OF PARTITIONING 15
REPLICATION OF TABLES 15
ADDITIONAL AND/OR DUPLICATED ATTRIBUTES IN ANY TABLE 15
CLAUSE 2 : TRANSACTION AND TERMINAL PROFILES RELATED ITEMS 16
RANDOM NUMBER GENERATION 16
TERMINAL INPUT/OUTPUT SCREEN LAYOUT 16
TERMINAL FEATURE VERIFICATION 16
PRESENTATION MANAGER OR INTELLIGENT TERMINAL 16
TRANSACTION PROFILES 16
TRANSACTION MIX 16
QUEUING MECHANISM 17
CLAUSE 3 : TRANSACTION AND SYSTEM PROPERTIES RELATED ITEMS 18
TRANSACTION SYSTEM PROPERTIES (ACID) 18
ATOMICITY TESTS 18
Completed Transactions *18*
Aborted Transactions *18*
CONSISTENCY TESTS 18
ISOLATION TESTS 18
DURABILITY TESTS 18
Instantaneous Interruption and Loss of Memory *19*
Loss of Log and Data Disk *19*
Loss of mirrored write-back cache *19*
CLAUSE 4 : SCALING AND DATABASE POPULATION RELATED ITEMS 20
INITIAL CARDINALITY OF TABLES 20
CONSTANT VALUE FOR THE NURAND FUNCTION 20
DISTRIBUTION OF TABLES AND LOGS 21
TYPE OF DATABASE 22
DATABASE MAPPING 23
60-DAYS SPACE 23
CLAUSE 5 : PERFORMANCE METRICS AND RESPONSE TIME RELATED ITEMS 24
THROUGHPUT 24

RESPONSE TIMES 24
KEYING AND THINK TIMES 24
RESPONSE TIME FREQUENCY DISTRIBUTION CURVES 25
RESPONSE TIME VERSUS THROUGHPUT CURVE 27
NEW-ORDER THINK TIME FREQUENCY DISTRIBUTION 28
NEW-ORDER THROUGHPUT VS. ELAPSED TIME 28
STEADY STATE 29
WORK PERFORMED DURING STEADY STATE 29
MEASUREMENT PERIOD DURATION AND CHECKPOINT DURATION 29
REGULATION OF TRANSACTION MIX 29
TRANSACTION STATISTICS 29
CHECKPOINT COUNT AND LOCATION 30

CLAUSE 6 : SUT, DRIVER, AND COMMUNICATION DEFINITION RELATED ITEMS 30

DESCRIPTIONS OF RTE 30
LOSS OF TERMINAL CONNECTIONS 30
EMULATED COMPONENTS 30
FUNCTIONAL DIAGRAMS AND DETAIL OF DRIVER SYSTEM 30
NETWORK CONFIGURATIONS AND DRIVER SYSTEM 30
NETWORK BANDWIDTH 30
OPERATOR INTERVENTION 31

CLAUSE 7 : PRICING RELATED ITEMS 31

HARDWARE AND SOFTWARE COMPONENTS 31
AVAILABILITY 31
THROUGHPUT, AND PRICE PERFORMANCE 31
COUNTRY SPECIFIC PRICING 31
USAGE PRICING 31
SYSTEM PRICING 32

CLAUSE 8 : AUDIT RELATED ITEMS 32

AUDITOR'S REPORT 32
AVAILABILITY OF THE FULL DISCLOSURE REPORT 32
AUDITOR'S LETTER 33

APPENDIX A : APPLICATION SOURCE CODE 34

APPENDIX B : DATABASE DESIGN ..... ERROR! BOOKMARK NOT DEFINED.

APPENDIX C : TUNABLE PARAMETERS ..... ERROR! BOOKMARK NOT DEFINED.

APPENDIX D : SPACE CALCULATION 342

APPENDIX E : PRICE QUOTATION 343

Abstract

This report documents the compliance of NEC Corporation's TPC Benchmark™ C tests on the NEC Express5800/1320Xd client/server system with version 5.2 of the TPC Benchmark C Standard Specification. 40 Clients (NEC Express5800/120Rf-2) were used as the front-end clients.

The operating system and the DBMS used on the server were SUSE LINUX Enterprise Server 9 for Itanium Processors (per CPU) and Oracle® Database 10g Enterprise Edition. The operating system on the clients was SUSE LINUX Standard Server 8. Those clients ran Apache HTTP Server and BEA Tuxedo® 8.1.
Two standard metrics, transaction-per-minute-C(tpmC) and price per tpmC($/tpmC) are reported, in accordance with the TPC Benchmark™ C Standard. The independent auditor's report by Francois Raab appears at the end of this report.

TPC Benchmark™ C Metrics

The standard TPC Benchmark™ C metrics, tpmC (transactions per minute), price per tpmC (three year capital cost per measured tpmC) are reported.

System	SW	Total System Cost	TpmC	$ per tpmC	Availability Date
NEC Express5800 /1320Xd	Oracle® Database 10g Enterprise Edition SUSE LINUX Enterprise Server 9 for Itanium Processors	$4,128,185	609,467	$6.78	September 1, 2004

Standard and Executive Summary Statements

The following pages contain executive summary of results for this benchmark.

Auditor

The benchmark configuration, environment and methodology were audited by Francois Raab of Info Sizing Inc. to verify compliance with the relevant TPC specifications.

Copyright © 2004 NEC Corporation

Preface

The TPC Benchmark™ C was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark™ C Standard Specifications Version 5.2.

TPC Benchmark™ C Overview

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

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 TPC-C results must include the tpmC rate, the associated price-per-tpmC, and the availability date of the priced configuration.

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

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

Despite the fact that this benchmark offers a rich environment that emulates many OLTP applications, this benchmark does not reflect the entire range of OLTP requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

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

Benchmark sponsors are permitted several possible system designs, insofar as they adhere to the model described and pictorially illustrated in Clause 6. A Full Disclosure Report of the implementation details, as specified in Clause 8, must be made available along with the reported results.

Document Structure

This TPC Benchmark™ C Full Disclosure Report is organized as follows:

Copyright © 2004 NEC Corporation

- The main body of the document lists each item in Clause 8 of the TPC-C Standard and explains how each requirement is satisfied.
- Appendix A contains the source code of the TPC-C application code used to implement the TPC-C transactions.
- Appendix B contains the database definition and population code used in the tests.
- Appendix C contains the tunable parameters used in the TPC-C tests.
- Appendix D contains space calculation table.
- Appendix E contains third-party price quotations.

Copyright © 2004 NEC Corporation

TPC Benchmark™ C Full Disclosure

The TPC Benchmark™ C Standard Specification requires test sponsors to publish, and make available to the public, a full disclosure report for the results to be considered compliant with the Standard. The required contents of the full disclosure report are specified in Clause 8. This report is intended to satisfy the Standard's requirement for full disclosure. It documents the compliance of the benchmark tests with each item listed in Clause 8 of the TPC Benchmark™ C Standard Specification.

In the Standard Specification, the main headings in Clause 8 are keyed to the other clauses. The headings in this report use the same sequence, so that they correspond to the titles or subjects referred to in Clause 8.

Each section in this report begins with the text of the corresponding item from Clause 8 of the Standard Specification, printed in italic type. The plain text that follows explains how the tests comply with the TPC Benchmark™ C requirement. In sections where Clause 8 requires extensive listings, the section refers to the appropriate appendix at the end of this report.

General Items

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 sections in this report correspond with that of the TPC-C standard specification.

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 at the beginning of this report.

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,*
- *longest checkpoint interval in minutes,*
- *number of transactions (all types) completed within the measurement interval,*
- *computed Maximum Qualified Throughput in tpmC,*
- *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 at the beginning of this report.

Application Program

The application program (as defined in 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 application source codes used in the TPC-C benchmark.

Sponsor

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

Copyright © 2004 NEC Corporation

This benchmark test was sponsored by NEC Corporation. NEC has authorized NEC Corp. to publish TPC-C performance and price/performance results for the NEC Express5800/1320Xd. Price quotations contained in Appendix E correspond to the NEC Express5800/1320Xd server.

Parameters and Options

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:

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

Appendix C contains the tunable parameters used in the TPC-C tests.

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.).*

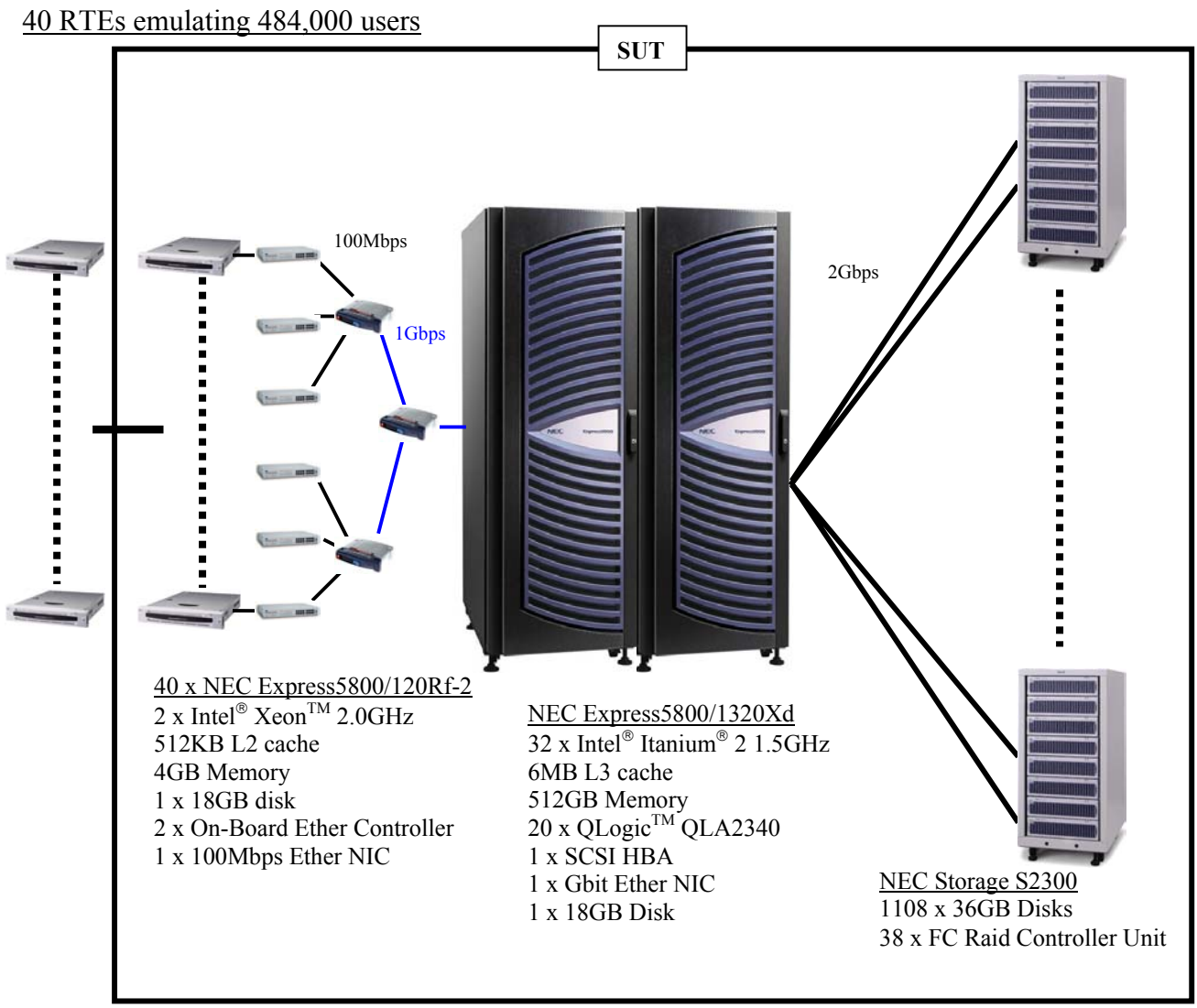
Figure 1.1 shows the measured configuration diagram.
Figure 1.2 shows the priced configuration diagram.

Copyright © 2004 NEC Corporation

Measured Configuration

The following figure represents the measured configuration. The benchmark system used a remote terminal emulator (RTE) to initiate transactions and measure response times of transactions, as well as record various data for each transaction.

Figure 1.1 Express5800/1320Xd, Measured Configuration Diagram

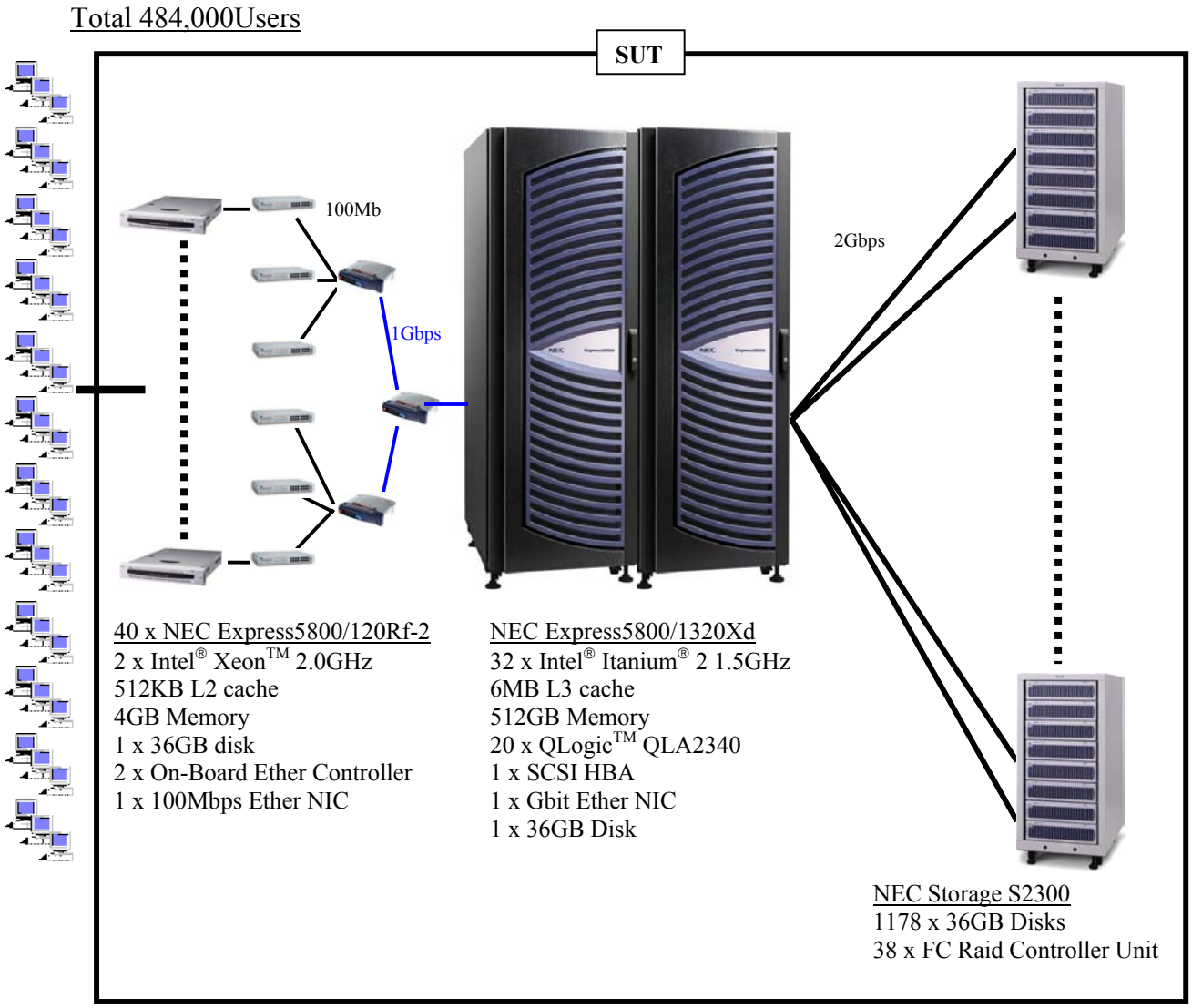


Copyright © 2004 NEC Corporation

Priced System Configuration

The following figure depicts the priced system, whose cost determines the normalized price per tpmC reported for the test.

Figure1.2: Express5800/1320Xd, Priced Configuration Diagram



Copyright © 2004 NEC Corporation

Clause 1 : Logical Database Design and Related Items

Table Definitions

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

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

Table Organization

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

Appendix B contains the code used to define the physical organization of tables and indices

Insert and Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restriction in the SUT database implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the maximum key value for these new rows

All insert and delete functions were fully operational during the entire benchmark.

Disclosure of Partitioning

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

Partitioning was not used on any table in this benchmark.

Replication of Tables

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

No tables were replicated in this benchmark test.

Additional and/or Duplicated Attributes in any Table

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

No duplications or additional attributes were used in this benchmark.

Copyright © 2004 NEC Corporation

Clause 2 : Transaction and Terminal profiles Related Items

Random Number Generation

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

Ramdom numbers generation in RTE and DB generator were verified by auditor independently.

Terminal Input/Output Screen Layout

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

All screen layouts were verified to be exactly followed the specification by auditor.

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).

Each of five transaction types was tested by the auditor. The auditor verified that all the features specified in Clause 2.2.2.4 were provided.

Presentation Manager or Intelligent Terminal

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

Application code running on the client machines implemented the TPC-C user interface. No presentation manager software or intelligent terminal features were used. The source code for the applications is listed in Appendix A.

Transaction Profiles

- *The percentage of home and remote order-lines in the New-Order transactions must be disclosed.*
- *The percentage of New-Order transactions that were rolled back as a result of an unused item number must be disclosed.*
- *The number of items per orders entered by New-Order transactions must be disclosed.*
- *The percentage of home and remote Payment transactions must be disclosed.*
- *The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the database must be disclosed.*
- *The percentage of Delivery transactions that were skipped as a result of an insufficient number of rows in the NEW-ORDER table must be disclosed.*

Table 1 shows the numerical quantities required by Clause 8.1.3.5 through 8.1.3.10.

Transaction Mix

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

Table 1 shows the mix of transaction types seen by the SUT during the reported measurement interval.

Following table summarizes the data required for disclosure in Clause 8.1.3.5 through 8.1.3.11

Copyright © 2004 NEC Corporation

Table 1 Transaction Statistics

	Statistic	Value
New Order	Home warehouse order lines	**98.99%**
	Remote warehouse order lines	**1.01%**
	Rolled back transactions	**0.99%**
	Average items per order	**9.9**
Payment	Home warehouse payments	**85.02%**
	Remote warehouse payments	**14.98%**
	Accessed by last name	**59.98%**
Order Status	Accessed by last name	**59.98%**
Delivery	Skipped deliveries	**0**
Transaction Mix	New Order	**44.81%**
	Payment	**43.07%**
	Delivery	**4.04%**
	Stock Level	**4.04%**
	Order Status	**4.04%**

Queuing Mechanism

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

The client application processes submitted delivery transactions to server software running on the client machines. These delivery servers were responsible for processing queued deliveries to submit to the database server.

The source code is listed in Appendix A.

Copyright © 2004 NEC Corporation

Clause 3 : Transaction and System Properties Related Items

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 quotes the specification definition of each of those properties and describes the tests done as specified and monitored by the auditor, to demonstrate compliance.

Atomicity Tests

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

Completed Transactions

Perform the Payment transaction for a randomly selected warehouse, district and customer(by customer number as specified in Clause 2.5.1.2) and verify that the records in the CUSTOMER,DISTRICT and WAREHOUSE tables have been changed appropriately.

The value of w_ytd, d_ytd, c_balance, c_ytd_payment and c_payment_cnt of a randomly selected warehouse, district, and customer were retrieved. The Payment transaction was executed on the same warehouse, district, and customer. The transaction was committed. The values w_ytd, d_ytd, c_balance, c_ytd_payment, and c_payment_cnt were retrieved again. It was verified that all values had been changed appropriately.

Aborted Transactions

Perform the Payment transaction for a randomly selected warehouse, district and customer (by customer number as specified in Clause 2.5.1.2) and substitute a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the records in CUSTOMER, DISTRICT and WAREHOUSE tables have Not been changed.

The value of w_ytd, d_ytd, c_balance, c_ytd_payment and c_payment_cnt of randomly selected warehouse, district, and customer were retrieved. The Payment transaction was executed on the same warehouse, district, and customer. The transaction was rolled back. The values of w_ytd, d_ytd, c_balance, c_ytd_payment, c_payment_cnt were retrieved again. It was verified that none of the values had changed.

Consistency Tests

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

Consistency conditions one through four were tested using a script to issue queries to the database. The results of the queries verified that the database was consistent for all four tests. A run was executed over 5 minutes under 50,000 users condition on a 50,000 warehouse database. The transaction rate was within 10% of the reported tpmC rate. A checkpoint generated in the test. The shell script of consistency was executed before and after the run. The result of the same queries verified that the database remained consistent after the run.

Isolation Tests

Sufficient conditions must be enabled at either the system or application level to ensure the required isolation level is obtained.

Isolation tests one through nine were executed using shell scripts to issue queries to the database. Each script included timestamps to demonstrate the concurrency of operations. The results of the queries were captured to files. The captured files were verified to demonstrate the required isolation had been met.
Case D was followed for Isolation Test 7.

Durability Tests

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

Copyright © 2004 NEC Corporation

- *Permanent irrecoverable failure of any single durable medium during the Measurement Interval containing TPC-C database tables or recovery log data.*
- *Instantaneous interruption (system or subsystem crash/system hang) in processing which causes all or part of the processing of atomic transactions to halt.*
- *Failure of all or part of memory(loss of contents)*

Instantaneous Interruption and Loss of Memory

Because the loss of power erases the contents of memory, both of instantaneous interruption and loss of memory were combined into a single test.

The following steps were performed on the full scaled database under the full load of users.

1. A sum of D_NEXT_O_ID of all rows in the district table was taken.
2. Full load of users were logged in to the database and kept running transactions about 5 minutes in steady state.
3. The system was powered off.
4. The RTE was shut down.
5. The system was powered up. Oracle® Database 10g was restarted and automatically recovered.
6. A new count of D_NEXT_O_ID was taken.
7. This number was compared with the number of new orders reported by the RTE.

Loss of Log and Data Disk

Loss of log and loss of data disk were combined to a one test and were demonstrated on a 50,000 warehouse database. The standard driving mechanism was used to generate the transaction load of 50,000 users for the test. To demonstrate recovery from a permanent failure of durable media containing TPC-C tables and log, the following steps were performed.

1. A sum of D_NEXT_O_ID of all rows in the district table was taken.
2. 50,000 users were logged in to the database and kept running transactions about 5 minutes in steady state.
3. Removed one of mirrored log disk. The running couninued without any interruptions.
4. One disk drive for data part in the array was removed causing Oracle® Database 10g error. Shut down Oracle® Database 10g.
5. Oracle® Database 10g was restarted and a dump of the transaction log was taken.
6. The 50,000 warehouse database was restored from backup.
7. The database was started and recovered by using recover command of Oracle® Database 10g.
8. A new count of D_NEXT_O_ID was taken.
9. This number was compared with the number of new orders reported by the RTE.

Loss of mirrored write-back cache

The Fibre Array system used for this benchmark has integrated feature of mirrored write-back cache. When a LUN is configured to enable write-back caching, the data on the cache is automatically mirrored on the RAMs in two controller modules, which are powered and protected from loss of power by the controller BBU. Loss of write-back cache was demonstrated on 50,000 warehouse database, by pulling off one of the controller module. A fully scaled database would also pass this test.

1. A sum of D_NEXT_O_ID of all rows in the district table was taken.
2. 50,000 users were logged in to the database and kept running transactions about 5 minutes in steady state.
3. A controller module, which manages write-back cache of mirrored drives, was pulled off.
4. Fibre system reported system error resulted in IO error for Oracle® Database 10g.
5. Contents on mirrored cache on another controller were automatically saved to temporal disk area and the Fibre Array house was eventually stopped. System was shut down.
6. The Fibre array was re-powered.
7. Saved data was automatically restored to mirrored cache and flushed to drives.
8. Oracle® Database 10g was started up and database was recovered.
9. A new count of D_NEXT_O_ID was taken.
10. This number was compared with the number of new orders reported by the RTE.

Copyright © 2004 NEC Corporation

Clause 4 : Scaling and Database Population Related Items

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 (see Clause 4.2), must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table were deleted (see Clause 4.2.2), the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed.

The TPC-C database was originally built with 50,000 warehouses.

Table 2 Number of Rows for Server

Table	Cardinality as benchmarked
Warehouse	48,400
District	500,000
Customer	1,500,000,000
History	1,500,000,000
Orders	1,500,000,000
New Order	450,000,000
Order Line	15,000,472,848
Stock	5,000,000,000
Item	100,000

During the measurement only 48,400 warehouses and their associated data were accessed. This was confirmed using D_NEXT_O_ID and W_YTD as described in Clause 4.2.2 Comment (2).

Constant Value for the NURand function

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

C_LAST (Build)	**1**
C_LAST (RUN)	**66**

Copyright © 2004 NEC Corporation

Distribution of Tables and Logs

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

Table 3 depicts the distribution of the database over the disks of the tested and priced system.
Figure 1.1, 1.2 shows the disk configuration for measured and priced system.

Table 3 : Data Distribution

HBA#	storage system#	SP#	DEU#	# of Disks	RAID Level	Capacity (GB)	Partition1	Partition2
Partitions for DB Log								
0	0	0	0	14	10	232.81		
			1	14	10	232.81	log_1 (110GB)	log_2 (110GB)

HBA#	storage system#	SP#	DEU#	# of Disks	RAID Level	Capacity (GB)	Partition1-12	Partition13 (4GB)	Partition14 (9GB)
Partitions for DB Data									
1	1	0	0	15	0	498.91	*1	ICUST1_0	NORDR_0
			1	15	0	498.91		ISTOK_0	TEMP_0
	2	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
2	3	0	0	15	0	498.91		ISTOK_0	TEMP_0
	4	0	0	15	0	498.91		ICUST1_0	-
			1	15	0	498.91		ISTOK_0	TEMP_0
3	5	0	0	15	0	498.91		ICUST1_0	NORDR_0
			1	15	0	498.91		ISTOK_0	TEMP_0
	6	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
4	7	0	0	15	0	498.91		ICUST1_0	control file
			1	15	0	498.91		ISTOK_0	TEMP_0
	8	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
5	9	0	0	15	0	498.91		ICUST1_0	control file
			1	15	0	498.91		ISTOK_0	TEMP_0
	10	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
6	11	0	0	15	0	498.91		ICUST1_0	SYSTEM
			1	15	0	498.91		ISTOK_0	TEMP_0
	12	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
7	13	0	0	15	0	498.91		ICUST1_0	SYSAUX
			1	15	0	498.91		ISTOK_0	TEMP_0
	14	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
8	15	0	0	15	0	498.91		ICUST1_0	UNDO_TS
			1	15	0	498.91		ISTOK_0	TEMP_0
	16	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
9	17	0	0	15	0	498.91		ICUST1_0	WARE_0
			1	15	0	498.91		ISTOK_0	TEMP_0
	18	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0

Copyright © 2004 NEC Corporation

10	19	0	0	15	0	498.91	*1	ICUST1_0	DIST_0
			1	15	0	498.91		ISTOK_0	TEMP_0
	20	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
11	21	0	0	15	0	498.91		ICUST1_0	ITEM_0
			1	15	0	498.91		ISTOK_0	TEMP_0
	22	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
12	23	0	0	15	0	498.91		ICUST1_0	IWARE_0
			1	15	0	498.91		ISTOK_0	TEMP_0
	24	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
13	25	0	0	15	0	498.91		ICUST1_0	IDIST_0
			1	15	0	498.91		ISTOK_0	TEMP_0
	26	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
14	27	0	0	15	0	498.91		ICUST1_0	IITEM_0
			1	15	0	498.91		ISTOK_0	TEMP_0
	28	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
15	29	0	0	15	0	498.91		ICUST1_0	SP_0
			1	15	0	498.91		ISTOK_0	TEMP_0
	30	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
16	31	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
	32	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
17	33	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
	34	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
18	35	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
	36	0	0	15	0	498.91		ICUST1_0	
			1	15	0	498.91		ISTOK_0	TEMP_0
19	37	0	0	15	0	498.91		ICUST1_0	NORDR_0

*1. Composition from Partition1 to Partition12 is shown in the following table.
Partition1-12

partiton#	tablespace name
Partition1	STOK_0 (7.8GB)
Partition2	STOK_0 (7.8GB)
Partition3	STOK_0 (7.8GB)
Partition4	STOK_0 (7.8GB)
Partition5	CUST_0 (7GB)
Partition6	CUST_0 (7GB)
Partition7	CUST_0 (7GB)
Partition8	HIST_0 (2.5GB)
Partition9	ORDR_0 (34GB)
Partition10	ICUST2_0 (1.4GB)
Partition11	IORDR2_0 (1.5GB)
Partition12	Reserve (2.5GB)

Type of Database

A statement must be provided that describes:

Copyright © 2004 NEC Corporation

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® Database 10g, a relational database, was used in this benchmark. Anonymous block PL/SQL and stored procedures were accessed through the ORACLE Call Interface. Application code is included in Appendix A.

Database Mapping

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

No partitioning or replication was used.

60-Days Space

Details of the 60-day space computations along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables (Order, Order-Line, and History) must be disclosed (see Clause 4.2.3).

The detail of 60-day space calculation is shown in Appendix D.
To calculate the space required to sustain the database log for 8 hours of growth at steady state, the following steps were followed:

1. Logfile usage was obtained from system view.
2. The space used was calculated as the difference between the first and second query.
3. The number of NEW-ORDERS was verified from an RTE report covering the entire run.
4. The space used was divided by the number of NEW-ORDERS giving a space used per NEW-ORDER transaction.
5. The space used per transaction was multiplied by the measured tpmC rate times 480 minutes.

The results of the above steps yielded a requirement of 1,368.76 GB of logspace (i.e 2,737.53 GB of mirrored transaction log volume) to be available to sustain 8 hours. Total space of priced disks for the transaction log volume was 3,259.28 GB. It indicates that enough storage was configured to sustain 8 hours of growth.

The same methodology was used to compute growth requirements for dynamic tables Order,Order-Line and History.

Copyright © 2004 NEC Corporation

Clause 5 : Performance Metrics and Response Time Related Items

Throughput

Measured tpmC must be reported

Table 4 : Measured tpmC

609,467 tpmC

Response Times

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

Table 5 : Response Times (in seconds)

Type	Average	Maximum	90th %
New-Order	0.21	53.44	0.27
Payment	0.20	45.83	0.25
Interactive Delivery	0.12	56.30	0.11
Stock Level	0.18	57.49	0.25
Order Status	0.20	41.71	0.27
Deferred Delivery	0.09	57.53	0.16
Menu	0.11	0.69	0.11

Keying and Think Times

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

Table 6 : Keying Times

Type	Average	Minimum	Maximum
New-Order	18.01	18.01	18.01
Payment	3.01	3.01	3.01
Interactive Delivery	2.01	2.01	2.01
Stock Level	2.01	2.01	2.01
Order Status	2.01	2.01	2.01

Table 7 : Think Times

Type	Average	Minimum	Maximum
New-Order	12.07	0.00	120.7
Payment	12.07	0.00	120.7
Interactive Delivery	5.07	0.00	50.7
Stock Level	5.08	0.00	50.7
Order Status	10.07	0.00	100.7

Copyright © 2004 NEC Corporation

Response Time Frequency Distribution Curves

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

Figure 2.1 : New-Order Response Time Distribution

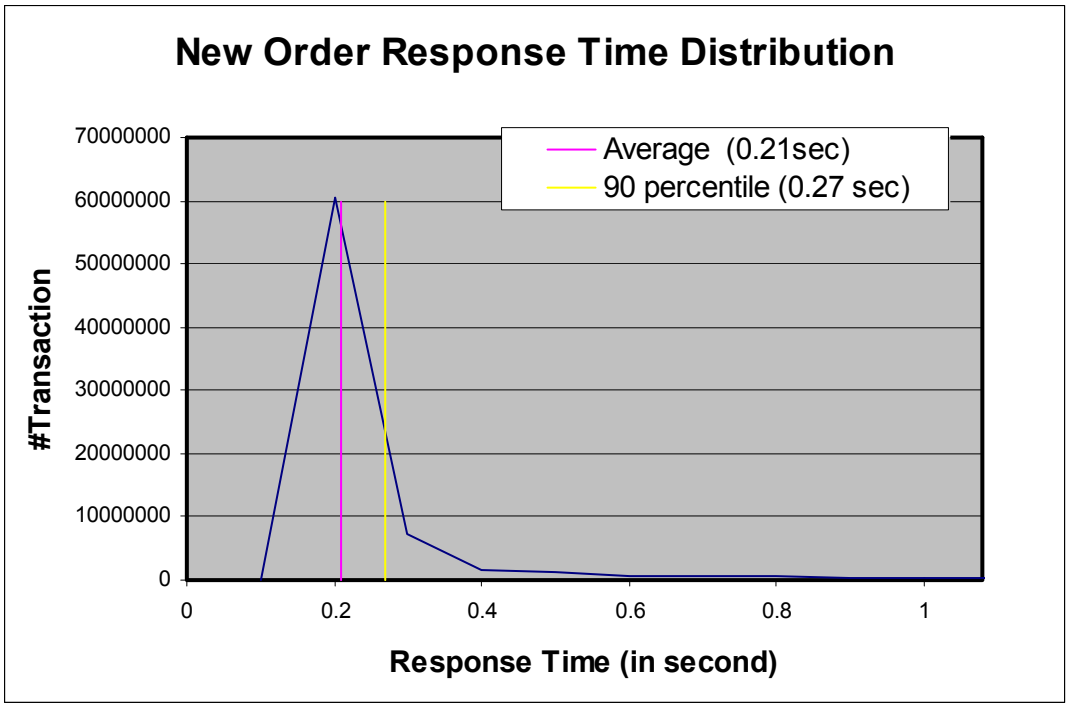
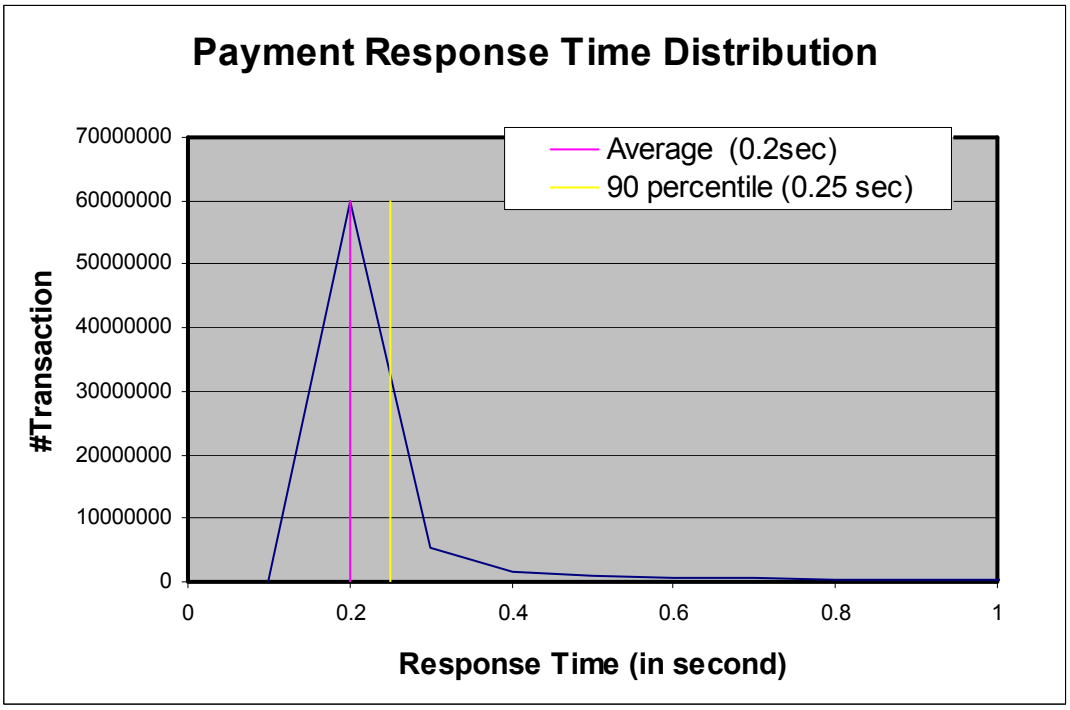


Figure 2.2 : Payment Response Time Distribution



Copyright © 2004 NEC Corporation

Figure 2.3 : Order-Status Response Time Distribution

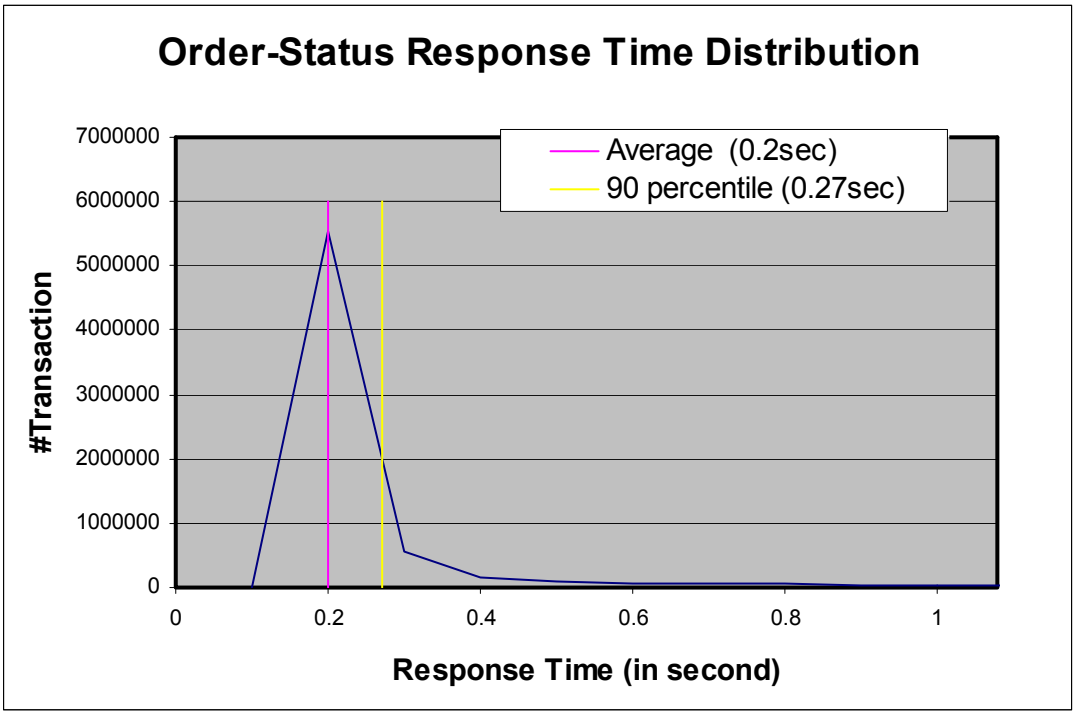
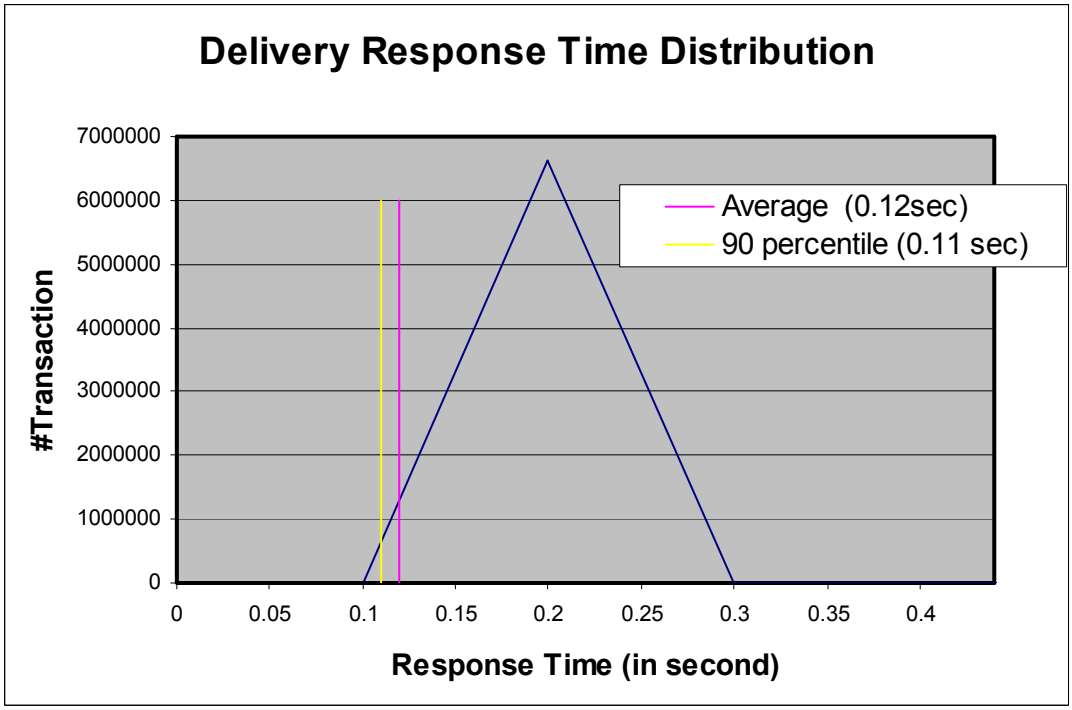
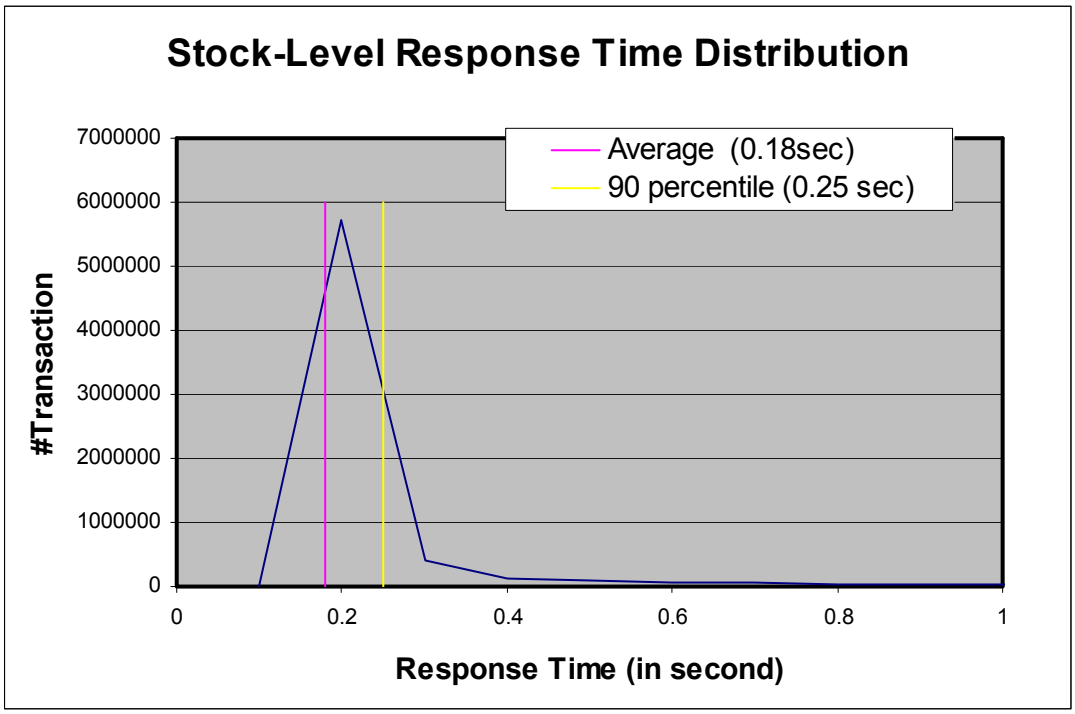


Figure 2.4 : Delivery Response Time Distribution



Copyright © 2004 NEC Corporation

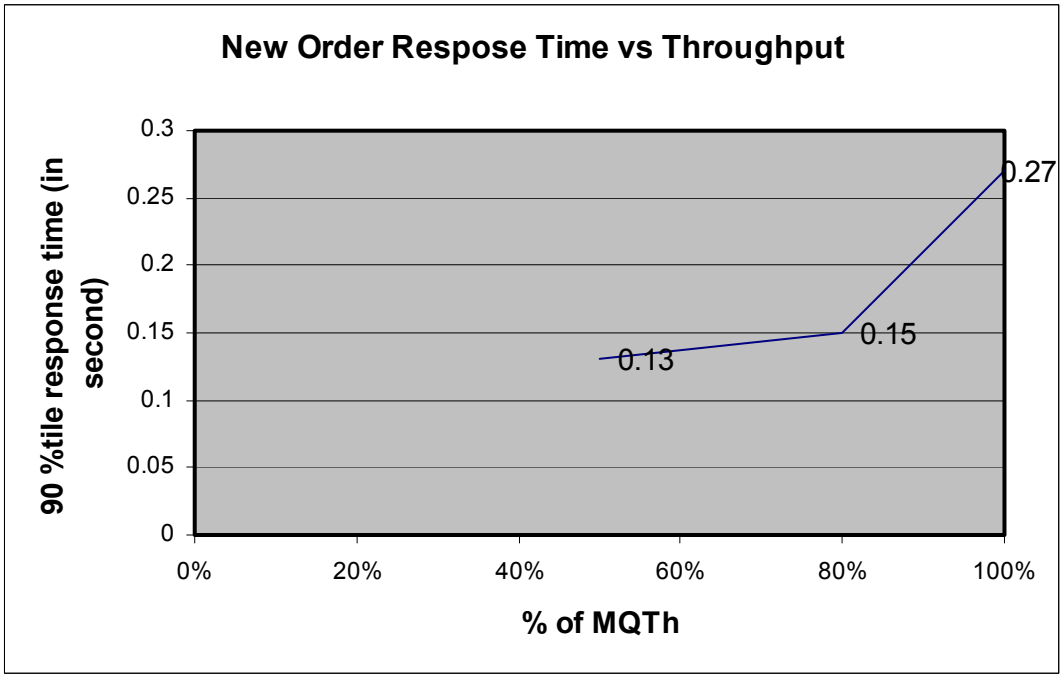
Figure 2.5 : Stock-Level Response Time Distribution



Response time versus Throughput Curve

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

Figure 2.6 Response Time vs. Throughput Curve

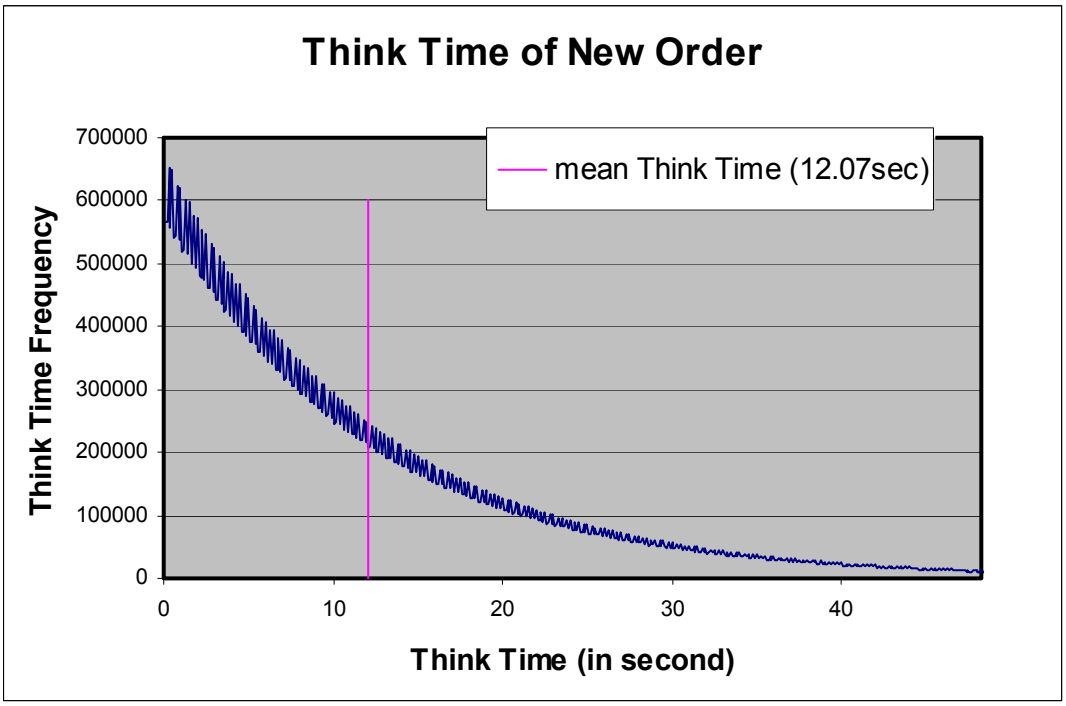


Copyright © 2004 NEC Corporation

New-Order Think Time Frequency Distribution

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

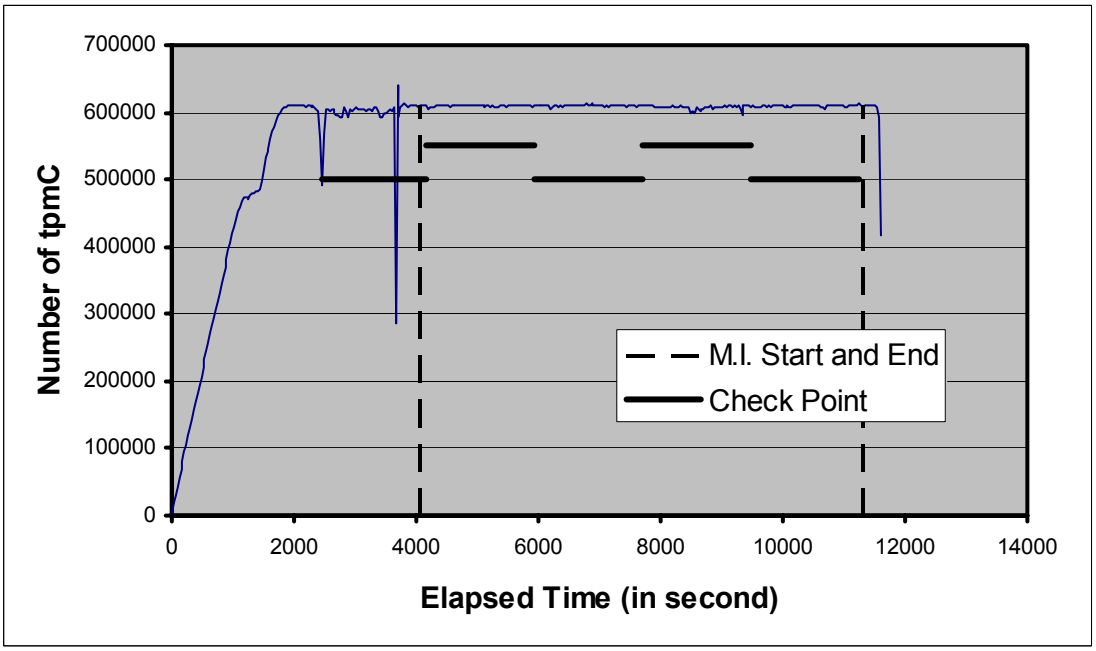
Figure 2.7 New-Order Think Time



New-Order Throughput vs. Elapsed Time

A graph of throughput versus elapsed time (see Clause 5.6.4) must be reported for the New-Order transaction.

Figure 2.8 New Order Throughput vs. Elapsed Time



Copyright © 2004 NEC Corporation

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.

Steady state was confirmed by the throughput data collected during the run and graphed in Figure 2.8.

Work Performed During Steady State

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

A checkpoint in Oracle® Database 10g writes to disk all updated memory pages that have not been yet actually written to disk. A checkpoint occurs at each switch of the redo log files. Oracle® Database 10g performs a log switch from one group to another when current redo logfiles are full. Redo log records are written to the redo log buffer (in memory) and are flushed to a redo log files on disk either when the transaction committed or when the redo log buffer became full. Oracle® Database 10g performed log switches (and checkpoints) four times during steady state of this benchmark.

Measurement Period Duration and Checkpoint Duration

- *The start time and duration in seconds of at least the four (4) longest checkpoints during the Measurement Interval must be disclosed (see Clause 5.5.2.2 (2)).*
- *A statement of the duration of the measurement interval for the reported Maximum Qualified Throughput (tpmC) must be included.*

	Start	End	Duration (in second)
M.I.	22:06:09	00:07:09	7260
1st Checkpoint	22:08:26	22:34:08	1542
2nd Checkpoint	22:36:56	23:02:53	1557
3rd Checkpoint	23:05:42	23:30:45	1503
4th Checkpoint	23:33:30	23:59:08	1538
5th Checkpoint	00:01:58	-	-

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 which could not be adjusted during the run.

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.*

Copyright © 2004 NEC Corporation

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

The above statistics are disclosed in Table 1.

Checkpoint Count and Location

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 was one checkpoint before measurement and four checkpoints during measurement.
The time of the first checkpoint during the measurement interval is 137 seconds after the start of the measurement, and the average checkpoint interval is 1701 seconds and a checkpoint lasts approximately 1535 seconds.

Clause 6 : SUT, Driver, and Communication Definition Related Items

Descriptions of RTE

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

The RTE used was NEC proprieta1ly. The RTE input parameters are listed in Appendix C.

Loss of Terminal Connections

The number of terminal connections lost during the Measurement Interval must be disclosed (see Clause 6.6.2).

No terminal connections were lost.

Emulated Components

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

As configured for this test, the driver software emulates the traffic that would be observed from the users' PCs connected by Ethernet to the front-end clients using HTTP (HyperText Transfer Protocol) over TCP/IP. One tenth of a second (100 milli seconds) was added to each transaction time to compensate for the overhead of the Web browser.

Functional Diagrams and Detail of Driver System

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

The diagrams in figure 1.1 and 1.2 show the tested and priced benchmark configurations.

Network configurations and Driver system

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

Figure 1.1 and 1.2 in this report has the network configurations of both the tested system and the priced system.

Network Bandwidth

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

The database server contains one Gigabit Ethernet adapter. This adapter is connected to a Gigabit Ethernet switch. This Gigabit Ethernet switch is cascaded to two Gigabit Ethernet switches, and these two Gigabit Ethernet switches are cascaded to six Ethernet switches with 100Mbps bandwidth. 40 front-end clients are connected to cascaded switches with 100Mbps bandwidth. Each client has two Ethernet adapters to connect to RTE system. The network bandwidth between RTE system and the front-end clients is 100Mbps. 80 segments are used for the connection of this tested configuration.

Copyright © 2004 NEC Corporation

Operator Intervention

If the configuration requires operator intervention (see Clause 6.6.6), the mechanism and the frequency of this intervention must be disclosed.111

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

Clause 7 : Pricing Related Items

Hardware and Software Components

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

The total 3-year price of the entire configuration must be reported, including: hardware, software, and maintenance charges. Separate component pricing is recommended. The justification of any discounts applied must be disclosed in the price sheet. Sufficient detail of what items are being discounted and by how much they are being discounted must be provided so that the discount amount used in the computation of the total system cost can be independently reproduced.

The detailed list of all hardware and software for the priced configuration is listed in the system pricing summary.

Availability

The committed delivery date for general availability (availability date) of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the reported availability date for the priced system must be the date at which all components are committed to be available. This single date must be reported on the first page of the Executive Summary. All availability dates, whether for individual components or for the SUT as a whole, must be disclosed to a precision of one day.

The total system as priced will be available September 1, 2004.

- "SUSE LINUX Enterprise Server 9 for Itanium Processors (per CPU)" will be available by September 1, 2004.
- "3 years Premium Support for SUSE LINUX Standard Server 8" for Clinent Hardware will be available by September 1, 2004.
- The other products are already available.

Throughput, and Price Performance

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

- Maximum Qualified Throughput : 609,467 tpmC
- Price per tpmC : $6.78 per tpmC
- Total 3-year cost of ownership : $4,128,185

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.

This system is priced for the United States of America.

Usage Pricing

For any usage pricing, the sponsor must disclose:

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

Copyright © 2004 NEC Corporation

The component pricing based on usage is shown below:

- 32 Oracle® Database 10g Enterprise Edition for 3 years, Per Processor, Unlimited Users
- 32 SUSE LINUX Enterprise Server 9 for Itanium Processors (per CPU)
- 40 SUSE LINUX Standard Server 8
- 40 BEA® Tuxedo 8.1 Core Functionality Services (CFS-R)

System Pricing

System pricing should include subtotals for the following components: Server Hardware, Server Software, Client Hardware, Client Software, and Network Components. Clause 6.1 describes the Server and Client components.

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

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

Clause 8 : Audit Related Items

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.

Next page contains the complete independent auditor's report by Francois Raab of Info Sizing Inc. for the test described in this report.

Availability of the Full Disclosure Report

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. The report must be made available when results are made public. In order to use the phrase "TPC Benchmark™ C", the Full Disclosure Report must have been submitted to the TPC Administrator as well as written permission obtained to distribute same.

Requests for this TPC Benchmark™ C Full Disclosure Report should be sent to:
Transaction Processing Performance Council
Presidio of San Francisco
Building 572B (surface)
P.O. Box 29920 (mail) San Francisco, CA 94129-0920
Voice: 415-561-6272
Fax: 415-561-6120
Email: info@tpc.org

Copyright © 2004 NEC Corporation

Auditor's letter

Sponsor: Koichi Aoyagi
NEC Corporation
4-14-22 Shibaura, Minato-ku
Tokyo 108-8558 Japan

April 4, 2004

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

Platform: NEC Express5800/1320Xd c/s
Operating system: SUSE LINUX Enterprise Server 9
Database Manager: Oracle Database 10g Enterprise Edition
Transaction Manager: BEA Tuxedo 8.1

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: NEC Express5800/1320Xd				
32 x Itanium2 (1.5 GHz)	512 GB (6 MB Cache/cpu)	1 x 36 GB int. 1178 x 36 GB ext.	0.27 Seconds	609,467.8
Forty (40) Clients: NEC Express5800/120Rf-2 (Specification for each)				
2 x Xeon (2.0 GHz)	4 GB (512 KB cache/cpu)	1 x 36 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC requirements for the benchmark.

1373 North Franklin Street • Colorado Springs, CO 80903-2527 • **Office**: 719/473-7555 • **Fax**: 719/473-7554

Copyright © 2004 NEC Corporation

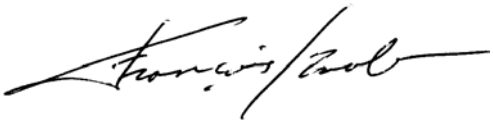
The following verification items were given special attention:

- The transactions were correctly implemented
- The database records were the proper size
- The database was properly scaled and populated
- The ACID properties were met
- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured.
- At least 90% of all delivery transactions met the 80 Second completion time limit
- All 90% response times were under the specified maximums
- The measurement interval was representative of steady state conditions
- The reported measurement interval was 120 minutes
- Four checkpoints were taken during the measurement interval
- The 60 day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

None.

Respectfully Yours,



François Raab, President

1373 North Franklin Street • Colorado Springs, CO 80903-2527 • **Office**: 719/473-7555 • **Fax**: 719/473-7554

Copyright © 2004 NEC Corporation

Appendix A : Application Source Code

A1. Front-End Source Code, Scripts

tpcc.c

```
/*==========================================================
======+
 |      Copyright (c) 1997  Oracle Corp, Redwood Shores, CA      |
 |                    All Rights Reserved                        |

+==========================================================
======+
 | FILE:  TPCC.C                                          |
 | DESCRIPTION: Main module for TPCC.DLL                  |
 | Master created: 15 Apr 97                              |

*==========================================================
==== */
#ifdef TOPEND
#define TP_MT_SOURCE
#define STRICT
#endif



#define LOCAL_ALLOC 1   /* force local alloc to be true */

#define APACHE                          /* convert this source to apache module */

//#include <windows.h>
//#include <process.h>
#include <stdio.h>
#include <stdarg.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys/timeb.h>
//#include <io.h>
#include <assert.h>
//#include <tchar.h>

#include "tpccerr.h"
#include "tpcc_info.h"
//#include "httpext.h"


//#include "getval.h"

#ifdef TOPEND
#include <tp_csi.h>
BOOL TP_InitDone = FALSE;
#define DllExport _declspec(dllexport)
DllExport LONG tp_ChangeToGroup(LPTSTR,DWORD,DWORD);

tp_dif_structs_t *client_dif;

/*
 * Topend structure offset.
 */
#define TOPEND_STRUCT_OFFSET 4124

#endif

#ifdef TUX
#include <tmenv.h>
#include <xa.h>
#include <atmi.h>
#include <Uunix.h>
#endif


//static void UtilStrCpy(char * pDest, char * pSrc, int n);

#include "mod_tpcc.h"
#include "tpccapi.h"

static TPINIT   *tpinf;
//static int  *    TLSIsTpInitedKey;

static DWORD    TLSNewOrderKey;
static DWORD    TLSPaymentKey;
static DWORD    TLSOrderStatusKey;
```


Copyright © 2004 NEC Corporation

```
static DWORD    TLSDeliveryKey;
static DWORD    TLSStockLevelKey;
static int              ThrTpInit();

char    szServer[32]    = { 0 };        //global variables used with this DLL
char    szUser[32]      = { 0 };
char    szPassword[32]  = { 0 };
char    szDatabase[32]  = "tpcc";
char    szIID[16][8]    =
{"IID00*","IID01","IID02","IID03","IID04","IID05","IID06","IID07","IID08","IID09
","IID10","IID11","IID12","IID13","IID14","IID15"};
char    szSP[16][8]     =
{"SP00*","SP01","SP02","SP03","SP04","SP05","SP06","SP07","SP08","SP09","SP1
0","SP11","SP12","SP13","SP14","SP15"};
char    szQty[16][8]    =
{"Qty00*","Qty01","Qty02","Qty03","Qty04","Qty05","Qty06","Qty07","Qty08","Qt
y09","Qty10","Qty11","Qty12","Qty13","Qty14","Qty15"};

BOOL    bLog            = FALSE;
BOOL    dLog            = FALSE;



#ifdef TOPEND
int             wait_time;
int     inactivity_time = 1500;
#endif

BOOL    bGeneric        = FALSE;
/* add structure to get delivery time stamp */
#ifdef WIN32
struct _timeb  timebuffer;
#else
struct timeb  timebuffer;
#endif

int             iThreads                = 5;
int             iMaxWareHouses  = 625;
int             iDelayMs                = 100;
short   iDeadlockRetry  = (short)3;
short   iMaxConnections = (short)625;
int             iErrVal                 = 0;

//allowable client command strings i.e. CMD=command
```

```
char *szCmds[] =
{
        "..NewOrder..", "..Payment..", "..Delivery..", "..Order-Status..", "..Stock-
Level..", "..Exit..",
        "Submit", "Begin", "Process", "Menu", "Clear", ""
};

//defined command string functions, called via CMD=command http string from html
client.

void (*DoCmd[])(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId) =
{
        NewOrderForm,
        PaymentForm,
        DeliveryForm,
        OrderStatusForm,
        StockLevelForm,
        Exitcmd,
        SubmitCmd,
        BeginCmd,
        ProcessCmd,
        MenuCmd,
        ClearCmd
};


//Terminal client id structure and interface defination
//TERM  Term = { 0, 0, 0, FALSE, NULL, TermInit, TermAllocate, TermRestore,
TermAdd, TermDelete };

TERM *Term=NULL;

apr_status_t tpcc_shm_create(tpcc_shm_t* m, apr_pool_t* p)
{
        apr_status_t s= apr_shm_create(&(m->shm),m->size, m->name, p);
//      if (s != APR_SUCCESS)
        if (dLog)
                {
                        FILE *fp;
                        fp = fopen("/tmp/errlog", "ab");
                        fprintf(fp, "create_shm *init Moudle pid=%d,
thread_id=%d, name=%s\n", getpid(),GetCurrentThreadId(),m->name);
                        fclose(fp);
```


Copyright © 2004 NEC Corporation

```
		}
	return s;
}


/* change shm segment size and preserve contents
 * the old shm segment must already attached before this call
 */
void* realloc_shm(tpcc_shm_t*m,apr_size_t new_size, apr_pool_t* p)
{
	apr_size_t old_size=m->size;
	apr_shm_t* new_shm;
	void* old, *new,*temp;
	if (dLog)
		{FILE *fp;	fp = fopen("/tmp/errlog", "ab");
		fprintf(fp, "* In realloc before malloc, size=%d\n",m->size);
		fclose(fp);}
	temp = malloc(old_size);
	if (temp)
		if (dLog)
		{FILE *fp;	fp = fopen("/tmp/errlog", "ab");
		fprintf(fp, "* In realloc malloc done\n");
		fclose(fp);}
	old=apr_shm_baseaddr_get(m->shm);
		if (dLog)
		{FILE *fp;	fp = fopen("/tmp/errlog", "ab");
		fprintf(fp, "* In realloc old_base_addr, shm=%x\n",m->shm);
		fclose(fp);}
	memcpy(temp, old, m->size);

//	apr_shm_detach(m->shm);
	apr_shm_destroy(m->shm);
	unlink(m->name);

	m->size=new_size;

	apr_shm_create(&(m->shm),m->size, m->name,p);

	new= apr_shm_baseaddr_get(m->shm);
	if (old_size>new_size)
		memcpy(new,temp,m->size);
	else
		memcpy(new,temp,old_size);
	free(temp);
		if (dLog)
		{FILE *fp;	fp = fopen("/tmp/errlog", "ab");
		fprintf(fp, "* In realloc new size=%d, shm=%x\n",m->size,m->shm);

		fclose(fp);}
	m->addr=new;
	return new;
}

char* _strupr(char* str)
{
	while(*str)
	{
		*str=toupper(*str);
		str++;
	}
	return str;
}

char *_strtime(char* str)
{
	time_t now;
	struct tm *local;
	time(&now);
	local=localtime(&now);
	strftime(str, 9, "%T",local);
}


//for debug
void debugdll(char *filename, char *format, ...)
{
	FILE *fp;
	va_list args;
	char buf[4096];
	int len;
	va_start( args, format );
	_strtime( buf );
	strcat( buf," ");
	len = strlen( buf );
#ifdef WIN32
	(void)_vsnprintf( buf+ len, sizeof( buf) - len - 1, format, args);
#else
	(void)vsnprintf( buf+ len, sizeof( buf) - len - 1, format, args);
```


Copyright © 2004 NEC Corporation

```
#endif
        buf[sizeof( buf )- 1]= '\0';
        va_end( args );

        if (fp = fopen(filename, "ab"))
          {
                        fprintf(fp,"%s.\n", buf);
                        fflush(fp);
                        fclose(fp);
        }
}

#ifdef WIN32
void WriteMessageToEventLog(LPTSTR lpszMsg)
                //XYZ//
{
  TCHAR   szMsg[256];
  HANDLE  hEventSource;
  LPTSTR  lpszStrings[2];

  // Use event logging to log the error.
  //
  hEventSource = RegisterEventSource(NULL, TEXT("TPCC.DLL"));

  _stprintf(szMsg, TEXT("Error in TPCC.DLL: "));
  lpszStrings[0] = szMsg;
  lpszStrings[1] = lpszMsg;

  if (hEventSource != NULL)
        {
    ReportEvent(hEventSource, // handle of event source
      EVENTLOG_ERROR_TYPE,  // event type
      0,              // event category
      0,              // event ID
      NULL,               // current user's SID
      2,              // strings in lpszStrings
      0,              // no bytes of raw data
      (LPCTSTR *)lpszStrings,    // array of error strings
      NULL);              // no raw data

    (VOID) DeregisterEventSource(hEventSource);
  }
}
#else
void write_err_log(request_rec *r, const char * szMsg)
{
  ap_log_rerror(APLOG_MARK, APLOG_ERR, 0, r, "%s",szMsg);
}
#endif


//welcome to tpc-c html form buffer, this is first form client sees.
static char     *szWelcomeForm =        "<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=\"ERROR\" 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=6><BR>"
                                                                "District ID
<INPUT NAME=\"d_id\" SIZE=2><BR>"
                                                                "<HR>"
                                                                "<INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Submit\">"
                                                                "
"
                                                                "
"
                                                                "
"
                                                                "
"
                                                                "
"
                                                                "
"
```


Copyright © 2004 NEC Corporation

```
                                                  "
"
                                                  "
"
                                                  "
"
                                                  "
"
                                                  "
"
                                                  "
"
                                                  "
"
                                                  "
"
                                                  "
"
                                                  "
"
                                                  "
"
                                                  "
"
                                                  "
"
                                                  "
"
        "</FORM><BODY>"
                                                  "</HTML>";

char szTpccLogPath[256]; //path to html log file if logging turned on in registry.
char szErrorLogPath[256];     //path to error log file.


static  CRITICAL_SECTION        CriticalSection;
static  CRITICAL_SECTION      ErrorLogCriticalSection;

int     bTpccExit;      //exit delivery disconnect loop as dll exiting.

#ifdef TOPEND
typedef struct DTEntry {
  HANDLE      hSync;
  struct DTEntry *next;
  tp_dif_structs_t *dif_struct;
  char        *message_buffer;
  long        *usr_dlg_ptr;
  long        usr_dlg_storage;
  long        buffer_length;
  int         rc;
} DialogTableEntry_t;

//a place to put free DTEntry cells
static DialogTableEntry_t DTFreeList={NULL,NULL,NULL,NULL,NULL,0,0,0};

//a place to put active dialogs
static DialogTableEntry_t DTActiveList={NULL,NULL,NULL,NULL,NULL,0,0,0};

//can use for tuning, keep high
#define RECEIVE_THREAD_PRIORITY THREAD_PRIORITY_TIME_CRITICAL
#define MT_LOG_ERROR {\
char buf[160];\
sprintf(buf,"Problem at %d in %s\n",__LINE__,__FILE__);\
tp_system_log_text("TEthreads",buf);\
}

static void TEReceiveThread(void *ptr);
DialogTableEntry_t *GetDTE();
DialogTableEntry_t *FindOnList(long tp_user_dialog_id,DialogTableEntry_t *head);
void RemoveFromList (DialogTableEntry_t *DTE, DialogTableEntry_t *head);
void AddToList (DialogTableEntry_t *DTE, DialogTableEntry_t *head);
//prototypes for calls used in the application

int tp_mt_initialize(tp_application_info_t *application_info,
                     tp_funct_struct_t   function_array[],
                     long member_functions);

int tp_mt_client_signon (tp_dialogue_info_t *info,
                         tp_dialogue_user_t *client,
                         long inactivity_time,
                         tp_service_name_t *service,
                         tp_input_format_t *input_format,
                         long   message_length,
                         char   *message_text);

int tp_mt_client_send (tp_dialogue_info_t *info,
                       tp_service_name_t *service,
```


Copyright © 2004 NEC Corporation

```
                                tp_input_format_t *input_format,
                                long    message_length,
                                char    *message_text);

int tp_mt_client_receive (tp_dialogue_info_t *info,
                                long        wait_time,
                                tp_output_format_t *output_format,
                                tp_service_name_t *service,
                                tp_location_t *location,
                                long        *buffer_length,
                                char        *message_buffer);

#endif


/* FUNCTION: BOOL APIENTRY DllMain(HANDLE hModule, DWORD
ul_reason_for_call,
 *          LPVOID lpReserved)
 *
 * PURPOSE: This function is the entry point for the DLL. This implementation
 *          is based on the fact that DLL_PROCESS_ATTACH is only called from
 *          the inet service once.
 *          Connections are sent to this function as thread attachments.
 *
 * ARGUMENTS:   HANDLE  hModule                module handle
 *              DWORD   ul_reason_for_call     reason for call
 *              LPVOID  lpReserved             reserved for future use
 *
 * RETURNS:     BOOL    FALSE                  errors occured in initialization
 *                      TRUE                   DLL successfully initialized
 *
 * COMMENTS: None
 *
 */
/*
BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call,
LPVOID lpReserved)
{
    int      i;
    static SECURITY_ATTRIBUTES       sa;
    static PSECURITY_DESCRIPTOR pSD;

#ifdef DEBUG_ENTRY
    DebugBreak( );
#endif


    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:
                if ( ReadRegistrySettings() )
                {
                  MessageBox(NULL, "Cannot Find TPCC Key in registry (run
install.exe).",
                                "Init", MB_OK | MB_ICONSTOP);
                  return FALSE;
                }

                InitializeCriticalSection(&CriticalSection);
                InitializeCriticalSection(&ErrorLogCriticalSection);


                TermInit(EXTENSION_CONTROL_BLOCK* pECB);
                if ( !TermAllocate(EXTENSION_CONTROL_BLOCK* pECB) )
                {
                  MessageBox(NULL, "Error Trm.Allocate().", "Init", MB_OK |
                  MB_ICONSTOP);
                  return FALSE;
                }

                for(i=Term->iNext; i<Term->iAvailable; i++)
                  Term->pClientData[i].inUse = 0;
                Term->pClientData[0].inUse = 1;

                TLSIsTpInitedKey = TlsAlloc(); // check for failure later


                TLSNewOrderKey = TlsAlloc();
                TLSPaymentKey = TlsAlloc();
                TLSOrderStatusKey = TlsAlloc();
                TLSDeliveryKey = TlsAlloc();
                TLSStockLevelKey = TlsAlloc();
                // assumption:value inited to 0


                break;

        case DLL_THREAD_ATTACH:
```


Copyright © 2004 NEC Corporation

```
                break;

        case DLL_THREAD_DETACH:
                break;

        case DLL_PROCESS_DETACH:


                if ( pSD )
                        free( pSD );

                bTpccExit = TRUE;

                TermRestore(EXTENSION_CONTROL_BLOCK* pECB);

                DeleteCriticalSection(&CriticalSection);
                DeleteCriticalSection(&ErrorLogCriticalSection);

                TlsFree(TLSIsTpInitedKey);

                TlsFree(TLSNewOrderKey);
                TlsFree(TLSPaymentKey);
                TlsFree(TLSOrderStatusKey);
                TlsFree(TLSDeliveryKey);
                TlsFree(TLSStockLevelKey);


                break;
    }
    return TRUE;
}
*/

/* FUNCTION: BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO
*pVer)
 *
 * PURPOSE: This function is called by the inet service when the DLL is first
 *          loaded.
 * ARGUMENTS:   HSE_VERSION_INFO       *pVer   passed in structure in which
to
 *                                  place expected version number.
 *
 * RETURNS:     TRUE                   inet service expected return value.
 *
 * COMMENTS:    None
 *
 */
#ifdef WIN32
_declspec (dllexport)

BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO *pVer)
{
    pVer->dwExtensionVersion = MAKELONG(HSE_VERSION_MINOR,
HSE_VERSION_MAJOR);
    lstrcpyn(pVer->lpszExtensionDesc, "TPC-C Server.",
HSE_MAX_EXT_DLL_NAME_LEN);
    return TRUE;
}
#endif

/* FUNCTION: DWORD WINAPI
HttpExtensionProc(EXTENSION_CONTROL_BLOCK *pECB)
 *
 * PURPOSE: This function is the main entry point for the TPCC DLL. The
 *          internet service calls this function passing in the http string.
 *
 * ARGUMENTS:   EXTENSION_CONTROL_BLOCK *pECB    structure pointer to
passed in
 *                                      internet service information.
 *
 * RETURNS:     DWORD   HSE_STATUS_SUCCESS    connection can be dropped
if error
 *                    HSE_STATUS_SUCCESS_AND_KEEP_CONN keep connect valid
comment sent
 *
 * COMMENTS:   None
 *
 */
/*
_declspec (dllexport)
DWORD WINAPI HttpExtensionProc(EXTENSION_CONTROL_BLOCK *pECB)

                    ////////////////////////////
*/
DWORD           HttpExtensionProc(EXTENSION_CONTROL_BLOCK *pECB)
{
    int iCmd, FormId, TermId, iSyncId;
    FILE  *fp;
```


Copyright © 2004 NEC Corporation

```
    static BOOL bReadRegistry = FALSE;


    request_rec *r = pECB->r;
//     SET_TERM_AND_CDATA

    if ( iMaxConnections == -1 )
    {
        ErrorMessage(pECB, ERR_CAN_NOT_SET_MAX_CONNECTIONS,
                    ERR_TYPE_WEBDLL, NULL, -1, -1);
        return HSE_STATUS_SUCCESS;
    }


    //if registry setting is for html logging then show http string passed in.
        if ( bLog )
        {

#ifdef WIN32
                SYSTEMTIME      systemTime;

                fp = fopen(szTpccLogPath, "ab");

                GetLocalTime(&systemTime);

                fprintf(fp, "* QUERY
* %2.2d/%2.2d/%2.2d %2.2d:%2.2d:%2.2d\r\n\r\n%s\r\n\r\n",
                        systemTime.tm_year, systemTime.tm_mon,
systemTime.tm_mday,
                        systemTime.tm_hour, systemTime.tm_min,
systemTime.tm_sec,
                        pECB->lpszQueryString);
                fclose(fp);
#else
                struct tm* systemTime;
                time_t    now;

                fp = fopen(szTpccLogPath, "ab");
                time(&now);
                systemTime = localtime(&now );

                fprintf(fp, "* QUERY
* %2.2d/%2.2d/%2.2d %2.2d:%2.2d:%2.2d\r\n\r\n%s\r\n\r\n",
                        systemTime->tm_year, systemTime->tm_mon,
systemTime->tm_mday,
                        systemTime->tm_hour, systemTime->tm_min,
systemTime->tm_sec,
                        pECB->lpszQueryString);
                fclose(fp);
#endif
        }

    //process http query
   if ( !ProcessQueryString(pECB, &iCmd, &FormId, &TermId, &iSyncId) )
   {
                if ( TermId < 0 ){


                        WriteMessageToEventLog(TEXT("ProcessQueryString is
failed. Invalid TermID"));

                    ErrorMessage(pECB, ERR_INVALID_TERMID,
ERR_TYPE_WEBDLL,
                            NULL, TermId, iSyncId);
                }
        else

                WriteMessageToEventLog(TEXT("Cannot get TermID and
SyncID"));

                ErrorMessage(pECB, ERR_COMMAND_UNDEFINED,
                        ERR_TYPE_WEBDLL, NULL, TermId, iSyncId);
            return 0;//HSE_STATUS_SUCCESS_AND_KEEP_CONN;
   }
    if ( TermId != 0 )
    {

                if ( !IsValidTermId(TermId) )
                {

#ifdef FURLOG
                                                        {       FILE
*fur;
                                                        fur =
fopen("C:\\temp\\fur.txt", "ab");
                                                        fprintf(fur,
"non0 Invalid termid = %d \n", TermId);
```


Copyright © 2004 NEC Corporation

```
                                                                fclose(fur);
                                                                        }
#endif // FURLOG


                WriteMessageToEventLog(TEXT("Invalid TermID"));

                ErrorMessage(pECB, ERR_INVALID_TERMID,
ERR_TYPE_WEBDLL,
                NULL, TermId, iSyncId);
                return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
                }

        //must have a valid syncid here since termid is valid
                if ( iSyncId < 1 || iSyncId != Term->pClientData[TermId].iSyncId )
                {


                        WriteMessageToEventLog(TEXT("Invalid SyncID"));

                        ErrorMessage(pECB,
ERR_INVALID_SYNC_CONNECTION,
                                ERR_TYPE_WEBDLL, NULL, TermId,
iSyncId);
                        return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
                }

    }

        //set use time
        Term->pClientData[TermId].iTickCount = GetTickCount();

        //go execute http: command

        (*DoCmd[iCmd])(pECB, FormId, TermId, iSyncId);
        //finish up and keep connection
        return HSE_STATUS_SUCCESS_AND_KEEP_CONN;

}


/* FUNCTION: static BOOL IsValidTermId(int TermId)
 *
 * PURPOSE:  This function checks to see of the passed in terminal id is valid.
 *
 * ARGUMENTS:   int     TermId          client terminal id
 *
 * RETURNS:     BOOL    FALSE           Terminal ID Invalid
 *                      TRUE            Terminal ID valid
 *
 * COMMENTS:    None
 *
 */

static BOOL IsValidTermId(int TermId)
{
    return (BOOL) ( TermId > 0 && TermId <= Term->iAvailable && Term-
>pClientData[TermId].inUse );
}


/* FUNCTION: BOOL ProcessQueryString(EXTENSION_CONTROL_BLOCK
*pECB, int *pCmd,
 *                              int *pFormId, int *pTermId, int *pSyncId)
 *
 * PURPOSE: This function extracts the relevent information out of the http
 *          command passed in from the browser.
 *
 * ARGUMENTS:   EXTENSION_CONTROL_BLOCK *pECB    structure pointer to
passed in
 *                                      internet service information.
 *            int                 *pCmd    returned command id
 *            int                 *pFormId returned active form client
                                                                browser is on
 *            int                 *pTermId returned client terminal id
 *
 * RETURNS:     BOOL    FALSE   success
 *                      TRUE    command passed in is invalid
 *
 * COMMENTS: If this is the initial connection i.e. client is at welcome screen
 *          then there will not be a terminal id or current form id.
 *          If this is the case then the pTermid and pFormid return values are
 *           undefined.
 */

BOOL ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd,
int *pFormId,
                                int *pTermId, int *pSyncId)
```


Copyright © 2004 NEC Corporation

```
{

	char *ptr;
	char szBuffer[25];
	char szTmp[25];
	char *dest = szBuffer;
	int i;

//		WR_LOG("enter pqs");
//		WR_LOG(pECB->lpszQueryString);
	if ( (ptr = strstr(pECB->lpszQueryString, "FORMID=")) )
		*pFormId = *(ptr+7) & 0x0F;

	if ( (ptr = strstr(pECB->lpszQueryString, "TERMID=")) )
	{
			*pTermId = atoi((ptr+7));

			if ( *pTermId == 0 )		//terminal id 0 used internally
				*pTermId = -1;
			if ( *pTermId == -2 )		//login screen
				*pTermId = 0;
	}
	else
			*pTermId = 0;

	if ( (ptr = strstr(pECB->lpszQueryString, "SYNCID=")) )
			*pSyncId = atoi((ptr+7));
	else
			*pSyncId = 0;

	if ( !(ptr = strstr(pECB->lpszQueryString, "CMD=")) )
	{
			ptr = szBuffer;

			if ( !stricmp(szBuffer, "Default") )
				strcpy(szBuffer, "CMD=Begin");

			switch( *pFormId )
			{
				case WELCOME_FORM:
					strcpy(szBuffer, "CMD=Submit");
					break;
				case MAIN_MENU_FORM:
					strcpy(szBuffer, "CMD=NewOrder");
					break;
				case NEW_ORDER_FORM:
				case PAYMENT_FORM:
				case DELIVERY_FORM:
				case ORDER_STATUS_FORM:
				case STOCK_LEVEL_FORM:
					if ( !(*pTermId) )
						return FALSE;
					if ( GetKeyValue(pECB->lpszQueryString,"PI*",
szTmp, sizeof(szTmp)) )
						strcpy(szBuffer, "CMD=Process");
					else
					{
						strcpy(szBuffer, "CMD=");
						strcat(szBuffer, szCmds[*pFormId -
NEW_ORDER_FORM]);
					}
					break;
				default:

					return FALSE;


			}
	}
	ptr += 4;

	while( *ptr && *ptr != '&' )
			*dest++ = *ptr++;
	*dest = 0;
//		WR_LOG(szBuffer);

	for(i=0; szCmds[i][0]; i++)
	{
			if ( !strcmp(szCmds[i], szBuffer) )
			{
				*pCmd = i;

			return TRUE;
			}
	}
	return FALSE;
}
```


Copyright © 2004 NEC Corporation

```
/* FUNCTION: void NewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId,
 *                        int iTermId, intiSyncId)
 *
 * PURPOSE:  This function wraps the functionality needed for the TPC-C New
 *          Order Form.
 *
 * ARGUMENTS:   int     iFormId        unused
 *            int     iTermId        id of calling browser, i.e. TERMID=from http
 *                                                    command line
 *            EXTENSION_CONTROL_BLOCK *pECB          structure pointer to
passed in
 *                                    internet service information
 *
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */

void NewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
{
    WriteZString(pECB, MakeNewOrderForm(iTermId, iSyncId, TRUE, FALSE,""));
    UNUSEDPARAM(iFormId);
    return;
}


/* FUNCTION: void PaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId, int
 *                        iTermId, int iSyncId)
 *
 * PURPOSE: This function wraps the functionality for the TPC-C Payment Form.
 *
 * ARGUMENTS:   int     iFormId        unused
 *            int     iTermId        id of calling browser, i.e. TERMID=
 *                                 from http command line
 *            int     iSyncId        sync id of calling browser
 *
 *            EXTENSION_CONTROL_BLOCK *pECB   structure pointer to passed in
 *                                  internet service information.
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */

void PaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
{
    WriteZString(pECB, MakePaymentForm(iTermId, iSyncId, TRUE) );
    UNUSEDPARAM(iFormId);
    return;

}


/* FUNCTION: void DeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId, int
 *                        iTermId, int iSyncId)
 *
 * PURPOSE: This function wraps the functionality for the TPC-C Delivery Form.
 *
 * ARGUMENTS:   int     iFormId        unused
 *            int     iTermId        id of calling browser, i.e. TERMID=
 *                                 from http command line
 *            int     iSyncId        sync id of calling browser
 *            EXTENSION_CONTROL_BLOCK *pECB   structure pointer to passed in
 *                                  internet service information.
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */

void DeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
{

    WriteZString(pECB, MakeDeliveryForm(iTermId, iSyncId, TRUE, "") );
    UNUSEDPARAM(iFormId);

    return;

}
```


Copyright © 2004 NEC Corporation

```
/* FUNCTION: void OrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB,
int iFormId,
 *                          int iTermId, int iSyncId)
 *
 * PURPOSE: This function wraps the function of the TPC-C Order Status Form.
 *
 * ARGUMENTS:   int     iFormId        unused
 *              int     iTermId        id of calling browser, i.e. TERMID=
 *                                     from http command line
 *              int     iSyncId        sync id of calling browser
 *              EXTENSION_CONTROL_BLOCK *pECB   structure pointer to passed in
 *                                     internet service information.
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */

void OrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
{

    WriteZString(pECB, MakeOrderStatusForm(iTermId, iSyncId, TRUE) );
    UNUSEDPARAM(iFormId);

    return;

}


/* FUNCTION: void StockLevelForm(EXTENSION_CONTROL_BLOCK *pECB,
int iFormId,
c *                          int iTermId, int iSyncId)
 *
 * PURPOSE:  This function wraps the functions of the TPC-C Stock Level Form.
 *
 * ARGUMENTS:   int     iFormId        unused
 *              int     iTermId        id of calling browser, i.e. TERMID=
 *                                     from http command line
 *              int     iSyncId        sync id of calling browser
 *              EXTENSION_CONTROL_BLOCK *pECB   structure pointer to passed in
 *                                     internet service information.
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */

void StockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
{

    WriteZString(pECB, MakeStockLevelForm(iTermId, iSyncId, TRUE) );

    return;
}


/* FUNCTION: void Exitcmd(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId, int iTermId, int iSyncId)
 *
 * PURPOSE: This function removes a terminal id from use, the allocated
 *          structure however remains valid so the next request for a new client
 *          will not require a new memory allocation.
 *
 * ARGUMENTS:   int     iFormId        unused
 *              int     iTermId        id of calling browser, i.e. TERMID=
 *                                     from http command line
 *              int     iSyncId        sync id of calling browser
 *              EXTENSION_CONTROL_BLOCK *pECB   structure pointer to passed in
 *                                     internet service information.
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */

void Exitcmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId)
{
    TermDelete(pECB, iTermId);
    WriteZString(pECB, MakeWelcomeForm() );
    UNUSEDPARAM(iFormId);
    UNUSEDPARAM(iSyncId);

    return;
}
```


Copyright © 2004 NEC Corporation

```
/* FUNCTION: void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId, int
 *                    iTermId, int iSyncId)
 *
 * PURPOSE: This function allocates a new terminal id in the Term struct array.
 *
 * ARGUMENTS:   int     iFormId        unused
 *            int     iTermId        id of calling browser, i.e. TERMID=
 *                                   from http command line
 *            int     iSyncId        sync id of calling browser
 *     EXTENSION_CONTROL_BLOCK        *pECB   structure pointer to passed in
 *                                   internet service information.
 * RETURNS:     None
 *
 * COMMENTS:  A terminal id can be allocated but still be invalid if the
 *          requested warehouse number is outside the range specified in the
 *          registry. This then will force the client id to be invalid and an
 *          error message sent to the users browser.
 */

void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
{
    int iCurrent;
    if ( (iCurrent = TermAdd(pECB, pECB->lpszQueryString)) < 0 )
    {
        ErrorMessage(pECB, ERR_CANNOT_INIT_TERMINAL,
ERR_TYPE_WEBDLL,NULL, iCurrent, iSyncId);
        return;
    }

    if ( Term->pClientData[iCurrent].w_id > iMaxWareHouses || Term-
>pClientData[iCurrent].w_id < 1 )
    {
        ErrorMessage(pECB, ERR_W_ID_INVALID, ERR_TYPE_WEBDLL,
NULL, iCurrent, iSyncId);
        TermDelete(pECB, iCurrent);
        return;
    }
    if ( Term->pClientData[iCurrent].d_id < 1 || Term->pClientData[iCurrent].d_id >
10 )
    {
        ErrorMessage(pECB, ERR_D_ID_INVALID, ERR_TYPE_WEBDLL,
NULL, iCurrent, iSyncId);
        TermDelete(pECB, iCurrent);
        return;
    }
    WriteZString(pECB, MakeMainMenuForm(iCurrent, Term-
>pClientData[iCurrent].iSyncId) );
    return;
}


/* FUNCTION: void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId, int iTermId, int iSyncId)
 *
 * PURPOSE:  This function is the first command executed. It is executed with
 *        the command:   CMD=Begin?Server=xxx from the http command line.
 *
 * ARGUMENTS:   int     iFormId        unused
 *            int     iTermId        id of calling browser, i.e. TERMID=
 *                                   from http command line
 *            int     iSyncId        sync id of calling browser
 *     EXTENSION_CONTROL_BLOCK        *pECB   structure pointer to passed in
 *                                   internet service information.
 *
 * RETURNS:     None
 *
 * COMMENTS:    SQL server must be specified, however the user and password
 *            parameters are optional.  The complete command line is
 *            CMD=Begin&Server=server&User=sa&Psw=&. The & are used
 *            to separate parameters which is internet browser standard.
 */

void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId)
{
    LPSTR pQueryString;
    pQueryString = pECB->lpszQueryString;

    WriteZString(pECB, MakeWelcomeForm() );

    UNUSEDPARAM(iFormId);
    return;
}
```


Copyright © 2004 NEC Corporation

```
/* FUNCTION: void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId, int
 *                    iTermId, int iSyncId)
 *
 * PURPOSE:    This function process the passed in http command
 *
 * ARGUMENTS:  int    iFormId       unused
 *             int    iTermId       id of calling browser, i.e. TERMID=
 *                                  from http command line
 *             int    iSyncId       sync id of calling browser
 *     EXTENSION_CONTROL_BLOCK       *pECB   structure pointer to passed in
 *                                   internet service information.
 * RETURNS:    None
 *
 * COMMENTS:   None
 *
 */
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
{
    switch( iFormId )
    {
        case WELCOME_FORM:
                return;
        case MAIN_MENU_FORM:
                return;
        }

    switch( iFormId )
    {
    case NEW_ORDER_FORM:
                ProcessNewOrderForm(pECB, iTermId, iSyncId);
                break;
    case PAYMENT_FORM:
                ProcessPaymentForm(pECB, iTermId, iSyncId);
                break;
    case DELIVERY_FORM:
                ProcessDeliveryForm(pECB, iTermId, iSyncId);
                break;
    case ORDER_STATUS_FORM:
                ProcessOrderStatusForm(pECB, iTermId, iSyncId);
                break;
    case STOCK_LEVEL_FORM:
                ProcessStockLevelForm(pECB, iTermId, iSyncId);
                break;
    }
}


/* FUNCTION: void ClearCmd(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId, int
 *                    iTermId, int iSyncId)
 *
 * PURPOSE:    This function frees all currently logged in terminal ids.
 *
 * ARGUMENTS:  int    iFormId       unused
 *             int    iTermId       id of calling browser, i.e. TERMID=
 *                                  from http command line
 *             int    iSyncId       sync id of calling browser
 *     EXTENSION_CONTROL_BLOCK       *pECB   structure pointer to passed in
 *                                   internet service information.
 * RETURNS:    None
 *
 * COMMENTS:   Use this function with caution, it may cause unpredictable
 *             results if existing browsers attempt to use the web client
 *             without beginning at the login screen for each client.
 */

void ClearCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId)
{
    int i;
    EnterCriticalSection(&CriticalSection);
    for(i=0; i<Term->iAvailable; i++)
    {
        if ( Term->pClientData[i].inUse )
                TermDelete(pECB, i);
    }
    Term->iNext             = 0;
    Term->iAvailable        = 0;
    Term->iMasterSyncId     = 1;

    if ( Term->pClientData )
        free(Term->pClientData);

    Term->pClientData       = NULL;
    Term->bInit             = FALSE;
```


Copyright © 2004 NEC Corporation

```
    TermInit(pECB->r->server);

    if ( !TermAllocate(pECB->r->server) )
    {
        ErrorMessage(pECB, ERR_MAX_CONNECT_PARAM,
ERR_TYPE_WEBDLL,NULL, iTermId, iSyncId);
        return;
    }

    for(i=Term->iNext; i<Term->iAvailable; i++)
        Term->pClientData[i].inUse = 0;

    Term->pClientData[0].inUse = 1;
    LeaveCriticalSection(&CriticalSection);

    WriteZString(pECB, MakeWelcomeForm() );
    return;
}


/* FUNCTION: void MenuCmd(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId,
 *                  int iTermId, int iSyncId)
 *
 * PURPOSE: This function causes an exit to the main menu
 *
 * ARGUMENTS:   int     iFormId        unused
 *          int     iTermId        id of calling browser, i.e. TERMID=
 *                              from http command line
 *          int     iSyncId        sync id of calling browser
 *     EXTENSION_CONTROL_BLOCK        *pECB   structure pointer to passed in
 *                              internet service information.
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */

void MenuCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId)
{
    WriteZString(pECB, MakeMainMenuForm(iTermId, iSyncId) );
    return;
}


/* FUNCTION: void WriteZString(EXTENSION_CONTROL_BLOCK *pECB, char
*szStr)
 *
 * PURPOSE:  This function is the low level output function. It writes a string
 * of text back to the client browser.
 *
 * ARGUMENTS: EXTENSION_CONTROL_BLOCK   *pECB   passed in structure
pointer from
 *                              inetsrv.
 *          char                *szStr  string to display in the client
 *                              browser.
 *
 * RETURNS:     None
 *
 * COMMENTS:    This function assumes that the string to written to the client
 *          browser has been formatted in an HTML manner.
 */

static void WriteZString(EXTENSION_CONTROL_BLOCK *pECB, char *szStr)

{
    FILE     *fp;
    int      lpbSize;
    int      iSize;
    char     szHeader[128];
    char     szHeader1[128];

    lpbSize = strlen(szStr)+1;

    if ( bLog )
    {
#ifdef WIN32
                SYSTEMTIME      systemTime;

                fp = fopen(szTpccLogPath, "ab");

                GetLocalTime(&systemTime);

                fprintf(fp, "* HTML PAGE
* %2.2d/%2.2d/%2.2d %2.2d:%2.2d:%2.2d\r\n\r\n%s\r\n\r\n",
                        systemTime.wYear, systemTime.wMonth,
```


Copyright © 2004 NEC Corporation

```
systemTime.wDay,
                                systemTime.wHour, systemTime.wMinute,
systemTime.wSecond,
                                szStr);
#else
                        struct tm* systemTime;
                        time_t    now;
                        fp = fopen(szTpccLogPath, "ab");
                        time(&now);
                        systemTime = localtime(&now );

                        fprintf(fp, "* QUERY
* %2.2d/%2.2d/%2.2d %2.2d:%2.2d:%2.2d\r\n\r\n%s\r\n\r\n%s\r\n",
                                systemTime->tm_year, systemTime->tm_mon,
systemTime->tm_mday,
                                systemTime->tm_hour, systemTime->tm_min,
systemTime->tm_sec,
                                pECB->lpszQueryString,szStr);
#endif
                        fclose(fp);
    }

#ifdef PERFORMIX

            (*pECB->ServerSupportFunction)(pECB->ConnID,
HSE_REQ_DONE_WITH_SESSION, NULL, 0, 0);
            (*pECB->WriteClient)(pECB->ConnID, szStr, &lpbSize, 0);

#else
#ifdef WIN32
            iSize = sprintf(szHeader, "200 Ok");

            sprintf(szHeader1, "Connection: keep-alive\r\nContent-type:
text/html\r\nContent-Length: %d\r\n\r\n", lpbSize);//change 2003/7/11
            (*pECB->ServerSupportFunction)(pECB->ConnID,
HSE_REQ_SEND_RESPONSE_HEADER, szHeader, &iSize,
(LPDWORD)szHeader1);
            (*pECB->WriteClient)(pECB->ConnID, szStr, &lpbSize, 0);
#else
            //ap_set_keepalive(pECB->r);
            //ap_set_content_length(pECB->r,lpbSize);
            ap_set_content_type(pECB->r,"text/html");
       ap_rputs(szStr, pECB->r);
#endif
#endif

     return;
}


/* FUNCTION: void h_printf(EXTENSION_CONTROL_BLOCK *pECB, char
*format, ...)
 *
 * PURPOSE:  This function forms a high level printf for an HTML browser
 *
 * ARGUMENTS:   EXTENSION_CONTROL_BLOCK *pECB   passed in structure
pointer from
 *                                  inetsrv.
 *            char                  *format printf style format string
 *            ...                   other arguments as required by printf
 *                                  style format string.
 *
 * RETURNS:     None
 *
 * COMMENTS: This function is mainly used for developmental support.
 */

static void h_printf(EXTENSION_CONTROL_BLOCK *pECB, char *format, ...)
{
     int lpbSize;
     char szBuff[512];
     char szTmp[512];

     va_list marker;
     va_start( marker, format );
     vsprintf(szTmp, format, marker);
     va_end( marker );

     lpbSize = wsprintf(szBuff, "<HTML>%s</HTML>", szTmp) + 1;
#ifdef WIN32
     (*pECB->WriteClient)(pECB->ConnID, szBuff, &lpbSize, 0);
#else
       ap_rputs(szBuff, pECB->r);
#endif

     return;
}
```


Copyright © 2004 NEC Corporation

```
/* FUNCTION: void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int
iError, int
 *                        iErrorType, char *szMsg)
 *
 * PURPOSE:     This function displays an error message in the client browser.
 *
 * ARGUMENTS:   EXTENSION_CONTROL_BLOCK *pECB   passed in structure
pointer from
 *                                inetsrv.
 *          int                iError  id of error message
 *          int          iErrorType      error type, ERR_TYPE_SQL,
 *                                ERR_TYPE_OCI, or
 *                                ERR_TYPE_WEBDLL
 *          int          iTermId         terminal id from browser
 *          int          iSyncid         sync id from browser
 *          char          *szMsg         optional error message string
 *                            used with ERR_TYPE_SQL and ERR_TYPE_OCI
 *
 * RETURNS:     None
 *
 * COMMENTS:    If the error type is ERR_TYPE_WEBDLL the szmsg parameter
may be
 *          NULL because it is ignored. If the error type is ERR_TYPE_SQL or
 *          ERR_TYPE_OCI then the szMsg parameter contains the text of the
 *          error message, so the szMsg parameter cannot be NULL.
 *
 */

void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int iError, int
iErrorType, char *szMsg, int iTermId, int iSyncId)
{
    int i;
    static SERRORMSG errorMsgs[] =
    {
        {   ERR_SUCCESS,   "Success, no error." },
        {   ERR_COMMAND_UNDEFINED, "Command undefined." },
        {   ERR_NOT_IMPLEMENTED_YET, "Not Implemented Yet." },
        {   ERR_CANNOT_INIT_TERMINAL, "Cannot initialize client
connection." },
        {   ERR_OUT_OF_MEMORY, "insufficient memory." },
        {   ERR_NEW_ORDER_NOT_PROCESSED, "Cannot process new Order
form." },
        {   ERR_PAYMENT_NOT_PROCESSED, "Cannot process payment
form." },
        {   ERR_NO_SERVER_SPECIFIED, "No Server name specified." },
        {   ERR_ORDER_STATUS_NOT_PROCESSED, "Cannot process order
status form." },
        {   ERR_W_ID_INVALID, "Invalid Warehouse ID." },
        {   ERR_CAN_NOT_SET_MAX_CONNECTIONS, "Insufficient memory
to allocate # connections." },
        {   ERR_NOSUCH_CUSTOMER, "No such customer." },
        {   ERR_D_ID_INVALID, "Invalid District ID Must be 1 to 10." },
        {   ERR_MAX_CONNECT_PARAM, "Max client connections exceeded,
run install to increase." },
        {   ERR_INVALID_SYNC_CONNECTION, "Invalid Terminal Sync
ID." },
        {   ERR_INVALID_TERMID, "Invalid Terminal ID." },
        {   ERR_PAYMENT_INVALID_CUSTOMER, "Payment Form, No such
Customer." },
        {   ERR_SQL_OPEN_CONNECTION, "SQLOpenConnection API
Failed." },
        {   ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY, "Stock Level
missing Threshold key \"TT*\"." },
        {   ERR_STOCKLEVEL_THRESHOLD_INVALID, "Stock Level
Threshold invalid data type range = 1 - 99." },
        {   ERR_STOCKLEVEL_THRESHOLD_RANGE, "Stock Level
Threshold out of range, range must be 1 - 99." },
        {   ERR_STOCKLEVEL_NOT_PROCESSED,  "Stock Level not
processed." },
        {   ERR_NEWORDER_FORM_MISSING_DID,  "New Order missing
District key \"DID*\"." },
        {   ERR_NEWORDER_DISTRICT_INVALID, "New Order District ID
Invalid range 1 - 10." },
        {   ERR_NEWORDER_DISTRICT_RANGE, "New Order District ID out
of Range. Range = 1 - 10." },
        {   ERR_NEWORDER_CUSTOMER_KEY,"New Order missing
Customer key \"CID*\"." },
        {   ERR_NEWORDER_CUSTOMER_INVALID, "New Order customer id
invalid data type, range = 1 to 3000." },
        {   ERR_NEWORDER_CUSTOMER_RANGE, "New Order customer id
out of range, range = 1 to 3000." },
        {   ERR_NEWORDER_MISSING_IID_KEY, "New Order missing Item Id
key \"IID*\"." },
        {   ERR_NEWORDER_ITEM_BLANK_LINES, "New Order blank order
lines all orders must be continuous." },
        {   ERR_NEWORDER_ITEMID_INVALID, "New Order Item Id is wrong
data type, must be numeric." },
```


Copyright © 2004 NEC Corporation

```
{	ERR_NEWORDER_MISSING_SUPPW_KEY, "New Order missing Supp_W key \"SP##*\"." },
{	ERR_NEWORDER_SUPPW_INVALID, "New Order Supp_W invalid data type must be numeric." },
{	ERR_NEWORDER_MISSING_QTY_KEY, "New Order Missing Qty key \"Qty##*\"." },
{	ERR_NEWORDER_QTY_INVALID, "New Order Qty invalid must be numeric range 1 - 99." },
{	ERR_NEWORDER_SUPPW_RANGE, "New Order Supp_W value out of range - 1 to MaxWarehouses." },
{	ERR_NEWORDER_ITEMID_RANGE, "New Order Item Id is out of range. Range = 1 to 999999." },
{	ERR_NEWORDER_QTY_RANGE, "New Order Qty is out of range. Range = 1 to 99." },
{	ERR_PAYMENT_DISTRICT_INVALID, "Payment District ID is invalid must be 1 - 10." },
{	ERR_NEWORDER_SUPPW_WITHOUT_ITEMID, "New Order Supp_W field entered without a Item_Id." },
{	ERR_NEWORDER_QTY_WITHOUT_ITEMID, "New Order Qty entered without a corresponding Item_Id." },
{	ERR_NEWORDER_NOITEMS_ENTERED, "New Order Blank Items between items, items must be continuous." },
{	ERR_PAYMENT_MISSING_DID_KEY,"Payment missing district Key \"DID*\"." },
{	ERR_PAYMENT_DISTRICT_RANGE,"Payment District Out of range, range = 1 - 10." },
{	ERR_PAYMENT_MISSING_CID_KEY,"Payment missing Customer Key \"CID*\"." },
{	ERR_PAYMENT_CUSTOMER_INVALID,"Payment Customer data type invalid, must be numeric." },
{	ERR_PAYMENT_MISSING_CLT,"Payment missing Customer Last Name Key \"CLT*\"." },
{	ERR_PAYMENT_LAST_NAME_TO_LONG,"Payment Customer last name longer than 16 characters." },
{	ERR_PAYMENT_CUSTOMER_RANGE,"Payment Customer ID out of range, must be 1 to 3000." },
{	ERR_PAYMENT_CID_AND_CLT, "Payment Customer ID and Last Name entered must be one or other." },
{	ERR_PAYMENT_MISSING_CDI_KEY,"Payment missing Customer district key \"CDI*\"." },
{	ERR_PAYMENT_CDI_INVALID,"Payment Customer district invalid must be numeric." },
{	ERR_PAYMENT_CDI_RANGE,"Payment Customer district out of range must be 1 - 10." },
{	ERR_PAYMENT_MISSING_CWI_KEY,"Payment missing Customer Warehouse key \"CWI*\"." },
{	ERR_PAYMENT_CWI_INVALID,"Payment Customer Warehouse invalid must be numeric." },
{	ERR_PAYMENT_CWI_RANGE,"Payment Customer Warehouse out of range, 1 to Max Warehouses." },
{	ERR_PAYMENT_MISSING_HAM_KEY,"Payment missing Amount key \"HAM*\"." },
{	ERR_PAYMENT_HAM_INVALID,"Payment Amount invalid data type must be numeric." },
{	ERR_PAYMENT_HAM_RANGE,"Payment Amount out of range, 0 - 9999.99." },
{	ERR_ORDERSTATUS_MISSING_DID_KEY,"Order Status missing District key \"DID*\"." },
{	ERR_ORDERSTATUS_DID_INVALID,"Order Status District invalid, value must be numeric 1 - 10." },
{	ERR_ORDERSTATUS_DID_RANGE,"Order Status District out of range must be 1 - 10." },
{	ERR_ORDERSTATUS_MISSING_CID_KEY,"Order Status missing Customer key \"CID*\"." },
{	ERR_ORDERSTATUS_MISSING_CLT_KEY, "Order Status missing Customer Last Name key \"CLT*\"." },
{	ERR_ORDERSTATUS_CLT_RANGE, "Order Status Customer last name longer than 16 characters." },
{	ERR_ORDERSTATUS_CID_INVALID, "Order Status Customer ID invalid, range must be numeric 1 - 3000." },
{	ERR_ORDERSTATUS_CID_RANGE, "Order Status Customer ID out of range must be 1 - 3000." },
{	ERR_ORDERSTATUS_CID_AND_CLT, "Order Status Customer ID and LastName entered must be only one." },
{	ERR_DELIVERY_MISSING_OCD_KEY, "Delivery missing Carrier ID key \"OCD*\"." },
{	ERR_DELIVERY_CARRIER_INVALID, "Delivery Carrier ID invalid must be numeric 1 - 10." },
{	ERR_DELIVERY_CARRIER_ID_RANGE, "Delivery Carrier ID out of range must be 1 - 10." },
{	ERR_PAYMENT_MISSING_CLT_KEY, "Payment missing Customer Last Name key \"CLT*\"." },
{	ERR_TPINIT_BAD,		"Tuxedo tpinit Failed."		},
{	ERR_TPALLOC_BAD,		"Tuxedo tpalloc Failed."		},
{	ERR_TPCALL_BAD,		"Tuxedo tpcall Failed."		},
```


Copyright © 2004 NEC Corporation

```
        {       TOPEND_SEND_ERROR,      "TOPEND client send error."},
        {       TOPEND_RECEIVE_ERROR,   "TOPEND client receive error."},
        {       0,              "" }
    };

    static char szNoMsg[] = "";
    char        *szForm;

    if ( !szMsg  )
        szMsg = szNoMsg;

    if ( iTermId > 0 && IsValidTermId(iTermId) )
        szForm = Term->pClientData[iTermId].szBuffer; //if termid valid use
common terminal   static buffer.
    else
        szForm = Term->pClientData[0].szBuffer;  //else term id invalid so use
common terminal static buffer.


    switch(iErrorType)
    {
        case ERR_TYPE_WEBDLL:
          for(i=0; errorMsgs[i].szMsg[0]; i++)
          {
                if ( iError == errorMsgs[i].iError )
                break;
          }

          if ( !errorMsgs[i].szMsg[0] )
                i = 1;

          strcpy(szForm,  "<HTML><HEAD><TITLE>Welcome To TPC-
C</TITLE></HEAD><BODY><FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">");
          wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\"NAME=\"STATUSID\" VALUE=\"%d\">", iError);
          wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"%d\">", iErrVal);
          wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\"NAME=\"TERMID\" VALUE=\"%d\">", iTermId);
          wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\"NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);
          wsprintf(szForm+strlen(szForm), "Error: TPCCWEB(%d): %s",
iError,errorMsgs[i].szMsg);
          strcat(szForm, "</FORM><BODY></HTML>");
          WriteZString(pECB, szForm);
          break;

        case ERR_TYPE_SQL:
          strcpy(szForm,"<HTML><HEAD><TITLE>Welcome To TPC-
C</TITLE></HEAD><BODY><FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">");
          wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\"NAME=\"STATUSID\" VALUE=\"%d\">", iError);
          wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"%d\">", iErrVal);
          wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\"NAME=\"TERMID\" VALUE=\"%d\">", iTermId);
          wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\"NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);
          wsprintf(szForm+strlen(szForm), "Error: Oracle(%d): %s", iError, szMsg);
          strcat(szForm, "</FORM><BODY></HTML>");
          WriteZString(pECB, szForm);
          break;

        case ERR_TYPE_OCI:
          strcpy(szForm,"<HTML><HEAD><TITLE>Welcome To TPC-
C</TITLE></HEAD><BODY><FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">");
          wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\"NAME=\"STATUSID\" VALUE=\"%d\">", iError);
          wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"%d\">", iErrVal);
          wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\"NAME=\"TERMID\" VALUE=\"%d\">", iTermId);
          wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\"NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);
          wsprintf(szForm+strlen(szForm), "Error: OCI(%d): %s", iError, szMsg);
          strcat(szForm, "</FORM><BODY></HTML>");
          WriteZString(pECB, szForm);
          break;

        case ERR_TYPE_TUXEDO:
                strcpy(szForm,  "<HTML><HEAD><TITLE>Welcome To
TPC-C</TITLE></HEAD><BODY><FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">");
                wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">", iError);
```


Copyright © 2004 NEC Corporation

```
                                wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"%d\">", iErrVal);
                                wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">", iTermId);
                                wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);
                                wsprintf(szForm+strlen(szForm), "Error: Tuxedo: %d %s",
iError, szMsg);
                                strcat(szForm, "</FORM><BODY></HTML>");
                                WriteZString(pECB, szForm);
                                break;
    }
    return;
}


/* FUNCTION: BOOL GetKeyValue(char *pQueryString, char *pKey, char *pValue,
 *                        int iMax, char **pBeyondKey)
 *
 * PURPOSE: This function parses http formatted string for specific key values.
 *
 * ARGUMENTS:   char    *pQueryString   http string from client browser
 *              char    *pKey           key value to look for
 *              char    *pValue  character array into which to place key's value
 *              int     iMax            maximum length of key value array.
 *              char    **pBeyondKey    Pointer to location beyond key value
 *
 * RETURNS:     BOOL    FALSE           key value not found
 *                      TRUE            key value found
 *
 * COMMENTS:    http keys are formatted either KEY=value& or KEY=value\0. This
 * DLL formats TPC-C input fields in such a manner that the keys can be
 * extracted in the above manner.
 */

static BOOL GetKeyValue(char *pQueryString, char *pKey, char *pValue, int iMax)

{
    char *ptr;

    if ( !(ptr=strstr(pQueryString, pKey)) )
            return FALSE;
    if ( !(ptr=strchr(ptr, '=')) )
            return FALSE;
    ptr++;
    iMax--;
    while( *ptr && *ptr != '&' && iMax)
    {
            *pValue++ = *ptr++;
            iMax--;
    }
    *pValue = 0;
    return TRUE;
}


/* FUNCTION: void TermInit(void)
 *
 * PURPOSE: This function initializes the client ternimal structure it is
 *          called when the TPCC.DLL is first loaded by the inet service.
 *
 * ARGUMENTS:   none
 *
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */

void TermInit(server_rec *s)
{
    tpcc_mod_conf* mc = myModConfig(s);
    TERM* Term= apr_shm_baseaddr_get(mc->shmTerm.shm);

    if ( Term->bInit )
            return;
    Term->iNext = 0;
    Term->iMasterSyncId = 1;
    Term->iAvailable = 0;
    Term->pClientData = NULL;
    Term->bInit = TRUE;
    return;
}


/* FUNCTION: void TermRestore(void)
 *
 * PURPOSE: This function frees allocated resources associated with the
```


Copyright © 2004 NEC Corporation

```
 * terminal structure.
 *
 * ARGUMENTS:   none
 *
 * RETURNS:     None
 *
 * COMMENTS:    This function is called only with the inet service unloads the
 *              TPCC.DLL
 *
 */

static void TermRestore(EXTENSION_CONTROL_BLOCK* pECB)
{
    Term->iNext = 0;
    Term->iAvailable = 0;
    Term->iMasterSyncId = 0;
    if ( Term->pClientData )
        free(Term->pClientData);
    Term->pClientData = NULL;
    Term->bInit = FALSE;
    return;
}

/* FUNCTION: int TermAllocate(void)
 *
 * PURPOSE: This funct allocates terminal array entries in the Term structure.
 *
 * ARGUMENTS:   None
 *
 * RETURNS:     int TRUE or 1 if sucessfull
 *              int FALSE or 0 if terminal id cannot be allocated.
 *
 * COMMENTS:    None
 *
 */

int TermAllocate(server_rec *s)
{
/*
    Term->iAvailable += 32;

    if ( !Term->pClientData )
        Term->pClientData = (PCLIENTDATA)malloc(Term->iAvailable *
sizeof(CLIENTDATA));
    else
        term->pClientData = (PCLIENTDATA)realloc(Term->pClientData, Term-
>iAvailable * sizeof(CLIENTDATA));
*/

        TERM* Term;
        apr_size_t new_size;
        tpcc_mod_conf* mc = (tpcc_mod_conf*)myModConfig(s);

        Term = apr_shm_baseaddr_get(mc->shmTerm.shm);
    if (dLog)
        {FILE *fp;      fp = fopen("/tmp/errlog", "ab");
            fprintf(fp, "* In realloc before iAvail=%d\n",Term->iAvailable);

            fclose(fp);
        }

        Term->iAvailable +=iMaxConnections;

        new_size=Term->iAvailable* sizeof (CLIENTDATA);
        Term->pClientData = (PCLIENTDATA)realloc_shm(&(mc-
>shmCData),new_size, mc->pool);
        if (dLog)
            {FILE *fp;      fp = fopen("/tmp/errlog", "ab");
            fprintf(fp, "* In realloc after iAvail=%d, shmCData=%x\n",Term-
>iAvailable,mc->shmCData.shm);
            fclose(fp);
        }

    return ( Term->pClientData ) ? 1 : 0;

        return 1;
}

/* FUNCTION: int TermAdd(EXTENSION_CONTROL_BLOCK *pECB, char
*pQueryString, int iTermId)
 *
 * PURPOSE: This function assigns a terminal id which is used to identify a
 *      client browser.
 *
 * ARGUMENTS: EXTENSION_CONTROL_BLOCK *pECB     passed instructure
pointer
 *                              from inetsrv.
 *      char            *pQueryString   http query string passed to
```


Copyright © 2004 NEC Corporation

```
 *                                  this DLL.
 *
 *          int          iTermId        terminal id from browser
 * RETURNS:    int             assigned terminal id
 *                      -1      cannot assign id error occured.
 *
 *
 * COMMENTS: if the terminal id cannot be assigned it is because of
 *         insufficient memory or the SQL connection cannot be allocated.
 *
 */

static int TermAdd(EXTENSION_CONTROL_BLOCK *pECB, char *pQueryString)
{
    char      szTmp[32];

    int       i, iCurrent, iTotalConnections, iTickCount;
    EnterCriticalSection(&CriticalSection);
    for(i=0, iTotalConnections = 0; i<Term->iAvailable; i++)
    {
          if ( Term->pClientData[i].inUse )
          iTotalConnections++;
    }

    if ( iTotalConnections >= iMaxConnections )
    {
          for(iCurrent = 1, i=1, iTickCount = 0x7FFFFFFF; i<iMaxConnections; i++)
          {
            if ( iTickCount > Term->pClientData[i].iTickCount )
            {
                    iTickCount = Term->pClientData[i].iTickCount;
                    iCurrent = i;
            }
          }
    }
    else
    {
          for(i=0; i<Term->iAvailable; i++)
          {
            if ( !Term->pClientData[i].inUse )
            break;
          }

          iCurrent = i;
    }
    if ( i == Term->iAvailable )
    {
          Term->iNext = Term->iAvailable;

          if ( !TermAllocate(pECB->r->server) )
            goto TermAddErr1;

          for(i=Term->iNext; i<Term->iAvailable; i++)
            Term->pClientData[i].inUse = 0;

          iCurrent = Term->iNext;
    }

    Term->pClientData[iCurrent].inUse = 1;
    if ( !GetKeyValue(pQueryString, "w_id", szTmp, sizeof(szTmp)) )
          goto TermAddErr1;

    Term->pClientData[iCurrent].w_id = (int)atoi(szTmp); //change 2003/07/19
    if ( !GetKeyValue(pQueryString, "d_id", szTmp, sizeof(szTmp)) )
          goto TermAddErr1;

    Term->pClientData[iCurrent].d_id = atoi(szTmp);
    Term->pClientData[iCurrent].iTickCount = GetTickCount();
    Term->pClientData[iCurrent].iSyncId = Term->iMasterSyncId++;
    if ( Init(pECB, iCurrent, Term->pClientData[iCurrent].iSyncId, szServer, szUser,
szPassword, szDatabase) )
    {
           //   WR_LOG("term init failed");
          TermDelete(pECB, iCurrent);
          goto TermAddErr1;
    }

    LeaveCriticalSection(&CriticalSection);

    return iCurrent;

TermAddErr1:
    LeaveCriticalSection(&CriticalSection);
    return -1;   //terminal unsuccessfully added
}


/* FUNCTION: void TermDelete(EXTENSION_CONTROL_BLOCK *pECB, int id)
```


Copyright © 2004 NEC Corporation

```
 *
 * PURPOSE:  This function makes a terminal entry in the Term array available
 *           for reuse.
 *
 * ARGUMENTS: EXTENSION_CONTROL_BLOCK   *pECB passed in structure
pointer from
 *                                inetsrv.
 *     int            id             Terminal id of client exiting
 *
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */

static void TermDelete(EXTENSION_CONTROL_BLOCK *pECB, int id)
{
    if ( id >= 0 && id < Term->iAvailable )
    {
        //TPCC_LOG("Term Delete");
        Close(pECB, id, -1);
        Term->pClientData[id].inUse = 0;

#ifndef LOCAL_ALLOC
      tpfree((char *)Term->pClientData[id].newOrderDataPtr);
      tpfree((char *)Term->pClientData[id].paymentDataPtr);
      tpfree((char *)Term->pClientData[id].orderStatusDataPtr);
      tpfree((char *)Term->pClientData[id].deliveryDataPtr);
      tpfree((char *)Term->pClientData[id].stockLevelDataPtr);
#endif

    }


    return;
}


/* FUNCTION: BOOL Init(EXTENSION_CONTROL_BLOCK *pECB, int iTermId,
int iSyncId,
 *                  char *szServer, char *szUser, char *szPassword,
 *                  char *szDatabase)
 *
 * PURPOSE: This function initializes the sql connection for use.
 *
 * ARGUMENTS:   EXTENSION_CONTROL_BLOCK *pECB   passed in structure
pointer
 *                                  from inetsrv.
 *     int    iTermId       id of browser client that this connection is for.
 *     int    iSyncId       sync id for this client session
 *     char   *szServer      sql server name
 *     char   *szUser        user name
 *     char   *szPassword    user password
 *     char   *szDatabase    database to use
 *
 * RETURNS:     BOOL    FALSE   if successfull
 *              TRUE    if an error occurs and connection cannot be established.
 *
 * COMMENTS:    None
 *
 */
BOOL Init(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId, char
*szServer, char *szUser, char*szPassword, char *szDatabase)
{
    char     szApp[32];
#ifndef LOCAL_ALLOC
    char     buf[40];
    int      iRc;
#endif

#ifdef TOPEND

    int      i;
    void     space_fill();
    tp_dif_structs_t  *tclient_dif;
    char     ascii_pid[10];
    int      msglen;
    FILE     *fp1;
#else
    sprintf(szApp, "TPCC:%ld", (int)iTermId);
  /*   Term->pClientData[iTermId].dbContext = NULL;     - do not keep context */
#endif


#ifdef TOPEND
    EnterCriticalSection(&CriticalSection);
    if (TP_InitDone == FALSE)
    {
```


Copyright © 2004 NEC Corporation

```
        TP_InitDone = TRUE;
        LeaveCriticalSection (&CriticalSection);

        if((i=(int)tp_ChangeToGroup((LPSTR)getenv("TP_SYSTEM"),
              LOGON32_LOGON_INTERACTIVE,
              LOGON32_PROVIDER_DEFAULT))!=0)
        {
          if (bLog)
          {
                fp1 = fopen(szTpccLogPath,"ab");
                fprintf(fp1,"Bad status from tp_ChangeToGroup,status=%d**\n",i);
                fclose(fp1);
          }
          MessageBox(NULL, "Failed in tp_ChangeToGroup.",
                          "Init", MB_OK | MB_ICONSTOP);
                return TRUE; //changed DMA
        }
        //initialize TOPEND

        if ((i=tp_mt_initialize(NULL,NULL,0L))!=TP_OK)
        {
          if (bLog)
          {
                fp1 = fopen(szTpccLogPath,"ab");
                fprintf(fp1,"Bad status from tp_mt_initialize,status=%d**\n",i);
                fclose(fp1);
          }

          return TRUE; //changed DMA

        }

    } //if init
    else
    LeaveCriticalSection(&CriticalSection);

    sprintf(szApp,"TPCC:%1d",(int)iTermId);
//     Term->pClientData[iTermId].dbproc=NULL;

    if((i=client_dif= tp_csi_alloc(TP_DIF_ALL))== NULL)
    {
        if (bLog)
        {
         //  SYSTEMTIME systemTime;
          fp1 = fopen(szTpccLogPath,"ab");
          fprintf(fp1,"Bad status from tp_csi_alloc,status=%d**\n",i);
          fclose(fp1);
        }

        ErrorMessage(pECB,TOPEND_SEND_ERROR,ERR_TYPE_WEBDLL,
                    NULL,iTermId,iSyncId);

        return TRUE;
    }

    (tp_dif_structs_t*)Term->pClientData[iTermId].tp_structadr = client_dif;
    tclient_dif = (tp_dif_structs_t*)Term->pClientData[iTermId].tp_structadr;

    sprintf(ascii_pid,"%d",iTermId);

    tclient_dif->info->tp_user_dialogue_id = iTermId;
    tclient_dif->info->tp_user_message_id = 0L;
    tclient_dif->info->tp_system_dialogue_id.tp_sys_dialogue = 0L;
    tclient_dif->info->tp_flags = TP_NOFLAGS;

    space_fill (tclient_dif->client->tp_userid,"tpcc cli",TP_USERID_LEN);
    space_fill (tclient_dif->client->tp_endpoint,ascii_pid,TP_ENDPOINT_LEN);
    space_fill (tclient_dif->client->tp_password,"",TP_PASSWORD_LEN);
    space_fill
        (tclient_dif->service->tp_product_name,"tpcc",TP_PROD_NAME_LEN);
    space_fill
        (tclient_dif->output_format->tp_format_name,"",TP_FMT_NAME_LEN);
    space_fill
        (tclient_dif->service->tp_function_name,"tpcc1",TP_FUNC_NAME_LEN);

    tclient_dif->service->tp_function_qualifier = 0;
    wait_time = TP_BLOCK;
    msglen = 0L;

    if ((i = tp_mt_client_signon(tclient_dif->info,tclient_dif->client,
                    inactivity_time,NULL,NULL,0,NULL)) != TP_OK)
    {
        if (bLog)
        {
          fp1 = fopen(szTpccLogPath,"ab");
          fprintf(fp1,"client signon failed,status=%d,iTermId=%d,iSyncId=%d\n",
                    i,iTermId,iSyncId);
          fclose(fp1);
```


Copyright © 2004 NEC Corporation

```
        }

        ErrorMessage(pECB,TOPEND_SEND_ERROR,ERR_TYPE_WEBDLL,
                    NULL,iTermId,iSyncId);

        return TRUE;
    }

    if ((i = tp_mt_client_receive(tclient_dif->info,wait_time,
                tclient_dif->output_format,tclient_dif->service,
                tclient_dif->location,&msglen,NULL)) != TP_OK)
    {
        if (bLog)
        {
          fp1 = fopen(szTpccLogPath,"ab");
          fprintf(fp1,"client receive from signon failed,status=%d\n",i);
          fclose(fp1);
        }

        ErrorMessage(pECB,TOPEND_SEND_ERROR,ERR_TYPE_WEBDLL,
                    NULL,iTermId,iSyncId);

        return TRUE;
    }

    if (bLog)
    {
        fp1 = fopen(szTpccLogPath,"ab");
        fprintf(fp1,"client %d signed on.\n",iTermId);
        fclose(fp1);
    }

#else
  #ifdef TUX
    #ifndef LOCAL_ALLOC    /* only do if not doing local alloc of tuxedo data
structures */
//     Add initialization of Tuxedo Structures

//          NEWORDER

                if ((Term->pClientData[iTermId].newOrderDataPtr =
(NewOrderData *)tpalloc("CARRAY", NULL, sizeof(NewOrderData))) == NULL)
                {
                        iRc = tperrno;
                        sprintf(buf, "Tpcalloc %d", iRc);
                        ErrorMessage(pECB, ERR_TPALLOC_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
                        {
                                FILE *fp;

                                fp = fopen(szTpccLogPath, "ab");
                                fprintf(fp, ">>>> Init Thread %d : NewOrder
tpalloc failed: iRc = %d \r\n",
                                        GetCurrentThreadId(), iRc);
                                fclose(fp);
                        }
                        sprintf(buf, "Tpcalloc %d", iRc);
                        ErrorMessage(pECB, ERR_TPALLOC_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
                        return TRUE;
                }

                if ( dLog )
                {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* Init Thread %d iTermId %d *
NewOrderDataPtr: %x \r\n",
                                GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].newOrderDataPtr);
                        fclose(fp);
                }

//          PAYMENT

                if ((Term->pClientData[iTermId].paymentDataPtr = (PaymentData
*)tpalloc("CARRAY", NULL, sizeof(PaymentData))) == NULL)
                {
                        iRc = tperrno;
                        {
                                FILE *fp;

                                fp = fopen(szTpccLogPath, "ab");
                                fprintf(fp, ">>>> Init Thread %d : Payment tpalloc
failed: iRc = %d \r\n",
                                        GetCurrentThreadId(), iRc);
                                fclose(fp);
```


Copyright © 2004 NEC Corporation

```
                }
                sprintf(buf, "Tpcalloc %d", iRc);
                ErrorMessage(pECB, ERR_TPALLOC_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
                return TRUE;
        }

        if ( dLog )
        {
                FILE *fp;

                fp = fopen(szTpccLogPath, "ab");
                fprintf(fp, "* Init Thread %d iTermId %d *
PaymentDataPtr: %x \r\n",
                        GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].paymentDataPtr);
                fclose(fp);
        }

//      ORDERSTATUS

        if ((Term->pClientData[iTermId].orderStatusDataPtr =
(OrderStatusData *)tpalloc("CARRAY", NULL, sizeof(OrderStatusData))) == NULL)
        {
                iRc = tperrno;
                {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, ">>>> Init Thread %d : OrderStatus
tpalloc failed: iRc = %d \r\n",
                                GetCurrentThreadId(), iRc);
                        fclose(fp);
                }
                sprintf(buf, "Tpcalloc %d", iRc);
                ErrorMessage(pECB, ERR_TPALLOC_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
                return TRUE;
        }

        if ( dLog )
        {
                FILE *fp;

                fp = fopen(szTpccLogPath, "ab");
                fprintf(fp, "* Init Thread %d iTermId %d *
OrderStatusDataPtr: %x \r\n",
                        GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].orderStatusDataPtr);
                fclose(fp);
        }

//      DELIVERY

        if ((Term->pClientData[iTermId].deliveryDataPtr = (DeliveryData
*)tpalloc("CARRAY", NULL, sizeof(DeliveryData))) == NULL)
        {
                iRc = tperrno;
                {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, ">>>> Init Thread %d : Delivery tpalloc
failed: iRc = %d \r\n",
                                GetCurrentThreadId(), iRc);
                        fclose(fp);
                }
                sprintf(buf, "Tpcalloc %d", iRc);
                ErrorMessage(pECB, ERR_TPALLOC_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
                return TRUE;
        }

        if ( dLog )
        {
                FILE *fp;

                fp = fopen(szTpccLogPath, "ab");

                fprintf(fp, "* Init Thread %d iTermId %d *
DeliveryDataPtr: %x \r\n",
                        GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].deliveryDataPtr);
                fclose(fp);
        }

//      STOCKLEVEL
```


Copyright © 2004 NEC Corporation

```
                if ((Term->pClientData[iTermId].stockLevelDataPtr =
(StockLevelData *)tpalloc("CARRAY", NULL, sizeof(StockLevelData))) == NULL)
                {
                        iRc = tperrno;
                        {
                                FILE *fp;

                                fp = fopen(szTpccLogPath, "ab");
                                fprintf(fp, ">>>> Init Thread %d : StockLevel
tpalloc failed: iRc = %d \r\n",
                                        GetCurrentThreadId(), iRc);
                                fclose(fp);
                        }
                        sprintf(buf, "Tpcalloc %d", iRc);
                        ErrorMessage(pECB, ERR_TPALLOC_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
                        return TRUE;
                }

                if ( dLog )
                {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");

                        fprintf(fp, "* Init Thread %d iTermId %d *
StockLevelDataPtr: %x \r\n",
                                GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].stockLevelDataPtr);
                        fclose(fp);
                }
    #endif
  #endif
#endif
    return FALSE;
}


/* FUNCTION: BOOL Close(EXTENSION_CONTROL_BLOCK*pECB, int
iTermId, int iSyncId)
 *
 * PURPOSE:     This function closes the sql connection for use.
 *
 * ARGUMENTS:   EXTENSION_CONTROL_BLOCK *pECB   passed in structure
pointer from
 *                                inetsrv.
 *     int     iTermId             id of browser client that this cxn is for.
 *     int     iSyncId                     sync id of client browser
 *
 * RETURNS:     BOOL    FALSE   if successfull
 *                  TRUE    if an error occurs and connection cannot be
 *                          terminated.
 *
 * COMMENTS:    None
 *
 */

static BOOL Close(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId)
{

#ifdef TOPEND

    tp_dif_structs_t *tclient_dif;
    tclient_dif = (tp_dif_structs_t *)Term->pClientData[iTermId].tp_structadr;
            tclient_dif->info->tp_flags = TP_SIGNOFF_IMMED;


    tp_client_signoff(tclient_dif->info);
    tp_csi_free (tclient_dif);
    return TRUE;

#else  /* do any database special closing, nothing at this time */
    return 0;
#endif
    UNUSEDPARAM(iSyncId);

}

/* FUNCTION: void FormatString(char *szDest, char *szPic, char *szSrc)
 *
 * PURPOSE:     This function formats a character string for inclusion in the
 *              HTML formatted page being constructed.
 *
 * ARGUMENTS:   char    *szDest Destination buffer where formatted string is to
 *                      be placed
 *              char    *szPic  picture string which describes how character
 *                      value is to be formatted.
```


Copyright © 2004 NEC Corporation

```
 *            char    *szSrc  character string value.
 *
 * RETURNS:     None
 *
 * COMMENTS: This function is used to format TPC-C phone and zip value strings.
 *
 */
static void FormatString(char *szDest, char *szPic, char *szSrc)
{
    while( *szPic )
    {
        if ( *szPic == 'X' )
        {
          if ( *szSrc )
                *szDest++ = *szSrc++;
          else
                *szDest++ = ' ';
        }
        else
          *szDest++ = *szPic;

        szPic++;
    }

    *szDest = 0;
    return;
}


/* FUNCTION: char *MakeStockLevelForm(int iTermId, int iSyncId, BOOL bInput)
 *
 * PURPOSE:     This function constructs the Stock Level HTML page.
 *
 * ARGUMENTS:   int     iTermId         client browser terminal id
 *              int     iSyncId         client browser sync id
 *              BOOL    bInput          TRUE if form is being constructed for
 *                                      input else FALSE
 *
 * RETURNS:     char *                  A pointer to buffer inside client
 *                                      structure where HTML form is built.
 *
 * COMMENTS:    The internal client buffer is created when the terminal id is
 *              assigned and should not be freed except when the client
 *              terminal id is no longer needed.
 */

static char *MakeStockLevelForm(int iTermId, int iSyncId, BOOL bInput)
{
    char      *szForm;
    szForm = (char *)Term->pClientData[iTermId].szBuffer;
    Term->pClientData[iTermId].trans.stocklevelData.stoin.w_id = (int)Term-
>pClientData[iTermId].w_id; //change 2003/7/19
    Term->pClientData[iTermId].trans.stocklevelData.stoin.d_id = (short)Term-
>pClientData[iTermId].d_id;


        if ( bInput)
        {
                strcpy(szForm, "<HTML><HEAD><TITLE>TPC-C Stock
Level</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=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"7\">"); //
STOCK_LEVEL_FORM = 7

                wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">", iTermId,
iSyncId);

                strcat(szForm, "<PRE>                        Stock-Level<BR>");
                wsprintf(szForm+strlen(szForm), "Warehouse: %6.6d
District: %2.2d<BR><BR>",
                Term->pClientData[iTermId].trans.stocklevelData.stoin.w_id,
                Term->pClientData[iTermId].trans.stocklevelData.stoin.d_id);

                strcat(szForm,  "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></HTML>");
                strcat(szForm,
"                                                        "
"                                                        "
"                                                        "
```


Copyright © 2004 NEC Corporation

```
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
                        );
            }
            else // Not bInput
            {

                        strcpy(szForm, "<HTML><HEAD><TITLE>TPC-C Stock
Level</TITLE></HEAD>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"7\">"); //
STOCK_LEVEL_FORM = 7

                        wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">", iTermId,
iSyncId);

                        strcat(szForm, "<PRE>                          Stock-Level<BR>");

                        wsprintf(szForm+strlen(szForm), "Warehouse: %6.6d
District: %2.2d<BR><BR>\
Stock Level Threshold: %2.2d<BR><BR>low stock: %3.3d</PRE><BR><HR>",
                                                Term-
>pClientData[iTermId].trans.stocklevelData.stoin.w_id,
                                                Term-
>pClientData[iTermId].trans.stocklevelData.stoin.d_id,
                                                Term-
>pClientData[iTermId].trans.stocklevelData.stoin.threshold,
                                                Term-
>pClientData[iTermId].trans.stocklevelData.stoout.low_stock);

                        strcat(szForm,  "<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..\">\
</FORM></HTML>");
                        strcat(szForm,
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
"                                                             "
                        );
            }
```


Copyright © 2004 NEC Corporation

```
    return szForm;
}


/* FUNCTION: char *MakeMainMenuForm(int iTermId, int iSyncId)
 *
 * PURPOSE:  This function makes the main menu HTML page
 *
 * ARGUMENTS:   int     iTermId         client browser terminal id
 *              int     iSyncId         client browser sync id
 *
 * RETURNS:     char *                  A pointer to buffer inside client
 *                                  structure where HTML form is built.
 *
 * COMMENTS:    The internal client buffer is created when the terminal id is
 *              assigned and should not be freed except when the client
 *              terminal id is no longer needed.
 */

static char *MakeMainMenuForm(int iTermId, int iSyncId)
{
    char      *szForm;
    szForm = (char *)Term->pClientData[iTermId].szBuffer;

    strcpy(szForm, "<HTML><HEAD><TITLE>TPC-C
MainMenu</TITLE></HEAD><BODY>"
                   "Select Desired Transaction.<BR><HR>"
                   "<FORM ACTION=\"tpcc.dll\"METHOD=\"GET\">");

    strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">");
    strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">");
    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">", iTermId);
    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);
    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">", MAIN_MENU_FORM);
    strcat(szForm, "<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..\">"
        "                                                       "
        "                                                       "
        "                                                       "
        "                                                       "
        "                                                       "
        "                                                       "
        "                                                       "
        "                                                       "
        "                                                       "
        "                                                       "
        "                                                       "
        "                                                       "
        "                                                       "
        "                                                       "
        "                                                       "
        "                                                       "
                "                                               "
                "                                               "
                "                                               "
                "                                               "
                "                                               "
                "                                               "
                "                                               "
    "</FORM>"
    "</HTML>" );
    return szForm;
}

/* FUNCTION: char *MakeWelcomeForm(void)
 *
 * PURPOSE: This function makes the HTML based welcome form
 *
 * ARGUMENTS:   None
 *
 * RETURNS:     char *  A pointer to the static HTML welcome form.
 *
 * COMMENTS:    The welcome form is static.
 */

static char *MakeWelcomeForm(void)
{
    return szWelcomeForm;
```


Copyright © 2004 NEC Corporation

```
}

/* FUNCTION: char *MakeNewOrderForm(int iTermId, BOOL bInput, BOOL
bValid, char *execution_status)
 *
 * PURPOSE:  This function makes/presents the HTML based New Order Form
 *
 * ARGUMENTS:   int     iTermId         client browser terminal id
 *              int     iSyncId         client browser sync id
 *              BOOL    bInput          TRUE if form is being constructed for
 *                                      input else FALSE
 *              BOOL    bValid          TRUE if NeworderData valid, ELSE FALSE
 *                                      effects output only
 *
 * RETURNS:     char *                  A pointer to buffer inside client
 *                                      structure where HTML form is built.
 *
 * COMMENTS:    The internal client buffer is created when the terminal id is
 *              assigned and should not be freed except when the client
 *              terminal id is no longer needed.
 */

static char *MakeNewOrderForm(int iTermId, int iSyncId, BOOL bInput, BOOL
bValid, char *execution_status)
{
    char     *szForm;
//  char     szName[146];
//  char     szCredit[14];
    int      i;

    szForm = (char *)Term->pClientData[iTermId].szBuffer;

    Term->pClientData[iTermId].trans.newOrderData.newin.w_id = Term-
>pClientData[iTermId].w_id;

        if ( bInput)
        {
                strcpy(szForm,"<HTML><HEAD><TITLE>TPC-C New
Order</TITLE>\
</HEAD><BODY><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"PI*\" VALUE=\"\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"3\">"); //
NEW_ORDER_FORM = 3

                sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">\
<PRE>                        New Order<BR>\
Warehouse: %6.6d   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>",
                                                iTermId,
                                                iSyncId,
                                                Term-
>pClientData[iTermId].trans.newOrderData.newin.w_id);


                strcat(szForm,     "<INPUT NAME=\"SP00*\" SIZE=6> <INPUT
NAME=\"IID00*\" SIZE=6>                    <INPUT NAME=\"Qty00*\"
SIZE=1><BR>\
<INPUT NAME=\"SP01*\" SIZE=6> <INPUT NAME=\"IID01*\" SIZE=6>
<INPUT NAME=\"Qty01*\" SIZE=1><BR>\
<INPUT NAME=\"SP02*\" SIZE=6> <INPUT NAME=\"IID02*\" SIZE=6>
<INPUT NAME=\"Qty02*\" SIZE=1><BR>\
<INPUT NAME=\"SP03*\" SIZE=6> <INPUT NAME=\"IID03*\" SIZE=6>
<INPUT NAME=\"Qty03*\" SIZE=1><BR>\
<INPUT NAME=\"SP04*\" SIZE=6> <INPUT NAME=\"IID04*\" SIZE=6>
<INPUT NAME=\"Qty04*\" SIZE=1><BR>\
<INPUT NAME=\"SP05*\" SIZE=6> <INPUT NAME=\"IID05*\" SIZE=6>
<INPUT NAME=\"Qty05*\" SIZE=1><BR>\
<INPUT NAME=\"SP06*\" SIZE=6> <INPUT NAME=\"IID06*\" SIZE=6>
<INPUT NAME=\"Qty06*\" SIZE=1><BR>\
<INPUT NAME=\"SP07*\" SIZE=6> <INPUT NAME=\"IID07*\" SIZE=6>
<INPUT NAME=\"Qty07*\" SIZE=1><BR>\
<INPUT NAME=\"SP08*\" SIZE=6> <INPUT NAME=\"IID08*\" SIZE=6>
<INPUT NAME=\"Qty08*\" SIZE=1><BR>\
<INPUT NAME=\"SP09*\" SIZE=6> <INPUT NAME=\"IID09*\" SIZE=6>
<INPUT NAME=\"Qty09*\" SIZE=1><BR>\
<INPUT NAME=\"SP10*\" SIZE=6> <INPUT NAME=\"IID10*\" SIZE=6>
<INPUT NAME=\"Qty10*\" SIZE=1><BR>\
<INPUT NAME=\"SP11*\" SIZE=6> <INPUT NAME=\"IID11*\" SIZE=6>
<INPUT NAME=\"Qty11*\" SIZE=1><BR>\
```


Copyright © 2004 NEC Corporation

```
<INPUT NAME=\"SP12*\" SIZE=6> <INPUT NAME=\"IID12*\" SIZE=6>
<INPUT NAME=\"Qty12*\" SIZE=1><BR>\
<INPUT NAME=\"SP13*\" SIZE=6> <INPUT NAME=\"IID13*\" SIZE=6>
<INPUT NAME=\"Qty13*\" SIZE=1><BR>\
<INPUT NAME=\"SP14*\" SIZE=6> <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></HTML>" );
                strcat(szForm,""
                );
        }
                else // not bInput
        {
                if (bValid)
                {
//              FormatHTMLString(szName, Term-
>pClientData[iTermId].NewOrderData.c_last, 16),
//              FormatHTMLString(szCredit, Term-
>pClientData[iTermId].NewOrderData.c_credit, 2);

                strcpy(szForm,"<HTML><HEAD><TITLE>TPC-C New
Order</TITLE>\
</HEAD><BODY><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"3\">"); //
NEW_ORDER_FORM = 3

sprintf(szForm+strlen(szForm),"<INPUT TYPE=\"hidden\" NAME=\"TERMID\"
VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">\
<PRE>                        New Order<BR>\
Warehouse: %6.6d   District: %2.2d                Date: %s
<BR>\
Customer:  %4.4d   Name: %s   Credit: %2.2s   %%Disc: %5.2f          <BR>\
Order Number: %8.8d  Number of Lines: %2.2d       W_tax: %5.2f   D_tax: %5.2f
<BR><BR>\
 Supp_W  Item_Id  Item Name              Qty  Stock  B/G  Price    Amount<BR>",
                                iTermId,
                                iSyncId,
                                Term-
>pClientData[iTermId].trans.newOrderData.newin.w_id,
                                Term-
>pClientData[iTermId].trans.newOrderData.newin.d_id,
                                Term-
>pClientData[iTermId].trans.newOrderData.newout.o_entry_d,
                                Term-
>pClientData[iTermId].trans.newOrderData.newin.c_id,
                                Term-
>pClientData[iTermId].trans.newOrderData.newout.c_last,
                                Term-
>pClientData[iTermId].trans.newOrderData.newout.c_credit,
                                Term-
>pClientData[iTermId].trans.newOrderData.newout.c_discount * 100,
                                Term-
>pClientData[iTermId].trans.newOrderData.newout.o_id,
                                Term-
>pClientData[iTermId].trans.newOrderData.newout.o_ol_cnt,
                                Term-
>pClientData[iTermId].trans.newOrderData.newout.w_tax * 100,
                                Term-
>pClientData[iTermId].trans.newOrderData.newout.d_tax * 100);

// We could make 0-4 be hard coded in one sprintf, since we know that there is at least
5
// order lines, or is this a benchmark special??
// etw
                for(i=0; i<Term-
>pClientData[iTermId].trans.newOrderData.newout.o_ol_cnt; i++)
                        {
//                      FormatHTMLString(szName, Term-
>pClientData[iTermId].NewOrderData.Ol[i].ol_i_name, 24);

                                sprintf(szForm+strlen(szForm),
"%6.6d   %6.6d   %-24.24s  %2.2d    %3.3d    %1c   $%6.2f  $%7.2f  <BR>",
                        //              Term-
>pClientData[iTermId].trans.newOrderData.newin.ol_supply_w_id[i],//change
                        //              Term-
>pClientData[iTermId].trans.newOrderData.newin.ol_i_id[i],//change
                                        Term-
>pClientData[iTermId].trans.newOrderData.newout.ol_supply_w_id[i],//change
                                        Term-
>pClientData[iTermId].trans.newOrderData.newout.ol_i_id[i],//change
                                        Term-
>pClientData[iTermId].trans.newOrderData.newout.i_name[i],
                                        Term-
```


Copyright © 2004 NEC Corporation

```
>pClientData[iTermId].trans.newOrderData.newout.ol_quantity[i],//change
                                //                      Term-
>pClientData[iTermId].trans.newOrderData.newin.ol_quantity[i],//change
                                                        Term-
>pClientData[iTermId].trans.newOrderData.newout.s_quantity[i],
                                                        Term-
>pClientData[iTermId].trans.newOrderData.newout.brand_generic[i],
                                                        Term-
>pClientData[iTermId].trans.newOrderData.newout.i_price[i],
                                                        Term-
>pClientData[iTermId].trans.newOrderData.newout.ol_amount[i] );
                                }
                        for(; i<15; i++)
                                strcat(szForm, "<BR>");

                                sprintf(szForm+strlen(szForm), "Execution Status: %24.24s
Total:  $%8.2f  ",
                                                        execution_status,
                                                        Term-
>pClientData[iTermId].trans.newOrderData.newout.total_amount);

                                strcat(szForm, "</PRE><HR><BR>\
<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..\">\
</FORM></HTML>");

                        }
                        else // not bValid
                        {

//              FormatHTMLString(szName, Term-
>pClientData[iTermId].NewOrderData.c_last, 16),
//              FormatHTMLString(szCredit, Term-
>pClientData[iTermId].NewOrderData.c_credit, 2);
// statusid set to 1 for benchcraft - indicator for invalid item
                        strcpy(szForm,"<HTML><HEAD><TITLE>TPC-C New
Order</TITLE>\
</HEAD><BODY><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"1\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"3\">");  //
NEW_ORDER_FORM = 3

                        sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">\
<PRE>                           New Order<BR>\
Warehouse: %6.6d   District: %2.2d                      Date:<BR>\
Customer:  %4.4d   Name: %s   Credit: %s   %Disc:<BR>\
Order Number: %8.8d  Number of Lines:           W_tax:        D_tax:<BR><BR>",
                                                        iTermId,
                                                        iSyncId,
                                                        Term-
>pClientData[iTermId].trans.newOrderData.newin.w_id,
                                                          Term-
>pClientData[iTermId].trans.newOrderData.newin.d_id,
                                                        Term-
>pClientData[iTermId].trans.newOrderData.newin.c_id,
                                                        Term-
>pClientData[iTermId].trans.newOrderData.newout.c_last,
                                                        Term-
>pClientData[iTermId].trans.newOrderData.newout.c_credit,
                                                        Term-
>pClientData[iTermId].trans.newOrderData.newout.o_id
                                                        );

                        strcat(szForm, " Supp_W  Item_Id  Item Name                Qty  Stock
B/G  Price    Amount<BR>\
<BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR>
<BR>");

                        sprintf(szForm+strlen(szForm), "Execution Status: %24.24s
Total:",
                                                        execution_status);

                        strcat(szForm, "</PRE><HR><BR>\
<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..\">\
</FORM></HTML>");
                        strcat(szForm,
```


Copyright © 2004 NEC Corporation

```
"                                                                    "
"                                                                    "
"                                                                    "
"                                                                    "
                    );
                    }
          }

    return szForm;
}

/* FUNCTION: char *MakePaymentForm(int iTermId, int iSyncId, BOOL bInput)
 *
 * PURPOSE:  This function has the Make Payment HTTP form
 *
 * ARGUMENTS:   int     iTermId client browser terminal id
 *              int     iSyncId client browser sync id
 *              BOOL    bInput  TRUE if form is being constructed for input
 *                      else FALSE
 *
 * RETURNS:     char *  A pointer to buffer inside client structure where HTML
 *                      form is built.
 *
 * COMMENTS: The internal client buffer is created when the terminal id is
 *           assigned and should not be freed except when the client terminal
 *           id is no longer needed.
 */
static char *MakePaymentForm(int iTermId, int iSyncId, BOOL bInput)
{
    char     *szForm;
    char     *ptr;
    char     szTmp[64];
    char     szW_Zip[26];
    char     szD_Zip[26];
    char     szC_Zip[26];
    char     szC_Phone[26];
//    char    szTmpStr1[122];
//    char    szTmpStr2[122];
//    char    szTmpStr3[122];
//    char    szTmpStr4[122];
    int      i;
    int      l;
    char     *szZipPic = "XXXXX-XXXX";

    szForm = (char *)Term->pClientData[iTermId].szBuffer;

    Term->pClientData[iTermId].trans.paymentData.payin.w_id = Term-
>pClientData[iTermId].w_id;

          if ( bInput )
          {
                    strcpy(szForm,  "<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"PI*\" VALUE=\"\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"4\">"); //
PAYMENT_FORM = 4

                    sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">\
<PRE>                         Payment<BR>\
Date:<BR><BR>\
Warehouse: %6.6d",
                                                                iTermId,
                                                                iSyncId,
                                                                Term-
>pClientData[iTermId].trans.paymentData.payin.w_id);

                                strcat(szForm,
"                   District: <INPUT NAME=\"DID*\"
SIZE=1><BR><BR><BR><BR><BR>\
Customer: <INPUT NAME=\"CID*\" SIZE=4>\
Cust-Warehouse: <INPUT NAME=\"CWI*\" SIZE=6>  \
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\">\
</BODY></FORM></HTML>");
```


Copyright © 2004 NEC Corporation

```
                strcat(szForm,""
                );
        }
        else // Not bInput
        {

//          FormatHTMLString(szTmpStr1, Term-
>pClientData[iTermId].PaymentData.w_street_1, 20);
//          FormatHTMLString(szTmpStr2, Term-
>pClientData[iTermId].PaymentData.d_street_1, 20);

//          FormatHTMLString(szTmpStr3, Term-
>pClientData[iTermId].PaymentData.w_street_2, 20);
//         FormatHTMLString(szTmpStr4, Term-
>pClientData[iTermId].PaymentData.d_street_2, 20);

                FormatString(szW_Zip, szZipPic, Term-
>pClientData[iTermId].trans.paymentData.payout.w_zip);
                FormatString(szD_Zip, szZipPic, Term-
>pClientData[iTermId].trans.paymentData.payout.d_zip);

//         FormatHTMLString(szTmpStr5, Term-
>pClientData[iTermId].PaymentData.w_city, 20);
//          FormatHTMLString(szTmpStr6, Term-
>pClientData[iTermId].PaymentData.w_state, 2);
//          FormatHTMLString(szTmpStr7, Term-
>pClientData[iTermId].PaymentData.d_city, 20);
//          FormatHTMLString(szTmpStr8, Term-
>pClientData[iTermId].PaymentData.d_state, 2);

//          FormatHTMLString(szTmpStr9, Term-
>pClientData[iTermId].PaymentData.c_first, 16);
//          FormatHTMLString(szTmpStr10, Term-
>pClientData[iTermId].PaymentData.c_middle, 2);
//          FormatHTMLString(szTmpStr11, Term-
>pClientData[iTermId].PaymentData.c_last, 16);

//          FormatHTMLString(szTmpStr12, Term-
>pClientData[iTermId].PaymentData.c_street_1, 20);
//         FormatHTMLString(szTmpStr13, Term-
>pClientData[iTermId].PaymentData.c_credit, 2);
//
//          FormatHTMLString(szTmpStr14, Term-
>pClientData[iTermId].PaymentData.d_street_2, 20);
                FormatString(szC_Zip, szZipPic, Term-
>pClientData[iTermId].trans.paymentData.payout.c_zip);
                FormatString(szC_Phone, "XXXXXX-XXX-XXX-XXXX",Term-
>pClientData[iTermId].trans.paymentData.payout.c_phone);

//          FormatHTMLString(szTmpStr15, Term-
>pClientData[iTermId].PaymentData.c_city, 20);
//          FormatHTMLString(szTmpStr16, Term-
>pClientData[iTermId].PaymentData.c_state, 2);

                        strcpy(szForm,  "<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"4\">"); //
PAYMENT_FORM = 4

                        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">\
<PRE>                        Payment<BR>\
Date: %19s<BR><BR>\
Warehouse: %6.6d                District: %2.2d<BR>\
%-20s                %-20s<BR>\
%-20s                %-20s<BR>\
%-20s %-2s %10.10s      %-20s %-2s %10.10s<BR><BR>\
Customer: %4.4d  Cust-Warehouse: %6.6d  Cust-District: %2.2d<BR>\
Name:   %-16s %-2s %-16s     Since:  %10s<BR>\
        %-20s                Credit: %s<BR>\
        %-20s                %%Disc:  %5.2f<BR>\
        %-20s %-2s %10.10s      Phone:  %-19.19s<BR><BR>\
Amount Paid:        $%7.2f     New Cust Balance: $%14.2f<BR>\
Credit Limit:   $%13.2f<BR><BR>",

                        iTermId,
                        iSyncId,
                        Term-
>pClientData[iTermId].trans.paymentData.payout.h_date,
                        Term-
>pClientData[iTermId].trans.paymentData.payin.w_id,
                       Term->pClientData[iTermId].trans.paymentData.payin.d_id,
                        Term-
```


Copyright © 2004 NEC Corporation

```
>pClientData[iTermId].trans.paymentData.payout.w_street_1,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.d_street_1,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.w_street_2,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.d_street_2,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.w_city,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.w_state,
                                szW_Zip,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.d_city,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.d_state,
                                szD_Zip,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.c_id,
                                Term-
>pClientData[iTermId].trans.paymentData.payin.c_w_id,
                                Term-
>pClientData[iTermId].trans.paymentData.payin.c_d_id,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.c_first,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.c_middle,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.c_last,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.c_since,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.c_street_1,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.c_credit,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.c_street_2,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.c_discount*100,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.c_city,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.c_state,
                                szC_Zip, szC_Phone,
                                (float)(Term-
>pClientData[iTermId].trans.paymentData.payin.h_amount)/100.0,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.c_balance,
                                Term-
>pClientData[iTermId].trans.paymentData.payout.c_credit_lim);

// This part can probably be made a little more efficient
//etw
                        ptr = Term-
>pClientData[iTermId].trans.paymentData.payout.c_credit;
//for dedug katsuf
                                if (bLog){
                                        FILE *fp1;
                                        fp1 = fopen(szTpccLogPath, "ab");
                                        fprintf(fp1, "c_cre= %s", ptr);
                                        fprintf(fp1, "c_da = %s",Term-
>pClientData[iTermId].trans.paymentData.payout.c_data);
                                        fclose(fp1);
                                }
                        if ( *ptr == 'B' && *(ptr+1) == 'C' )
                        {
                                ptr = Term-
>pClientData[iTermId].trans.paymentData.payout.c_data;
                                l = strlen( ptr ) / 50;
                                for(i=0; i<4; i++, ptr += 50)
                                {
                                        if ( i <= l )
                                                UtilStrCpy(szTmp, ptr, 50);
                                        else
                                                szTmp[0] = 0;
                                        if ( !i )
                                        {
//                      FormatHTMLString(szTmpStr1, szTmp, 50);
                                                wsprintf(szForm+strlen(szForm), "Cust-
Data: %-50s<BR>", szTmp);
                                        }
                                        else
                                        {
//                      FormatHTMLString(szTmpStr1, szTmp, 50);
                                                wsprintf(szForm+strlen(szForm),
"       %-50s<BR>", szTmp);
                                        }
                                }
```


Copyright © 2004 NEC Corporation

```
                }
                else {
                        strcat(szForm, "Cust-Data: <BR><BR><BR><BR>");
                }

                strcat(szForm,  "</PRE><HR><BR>\
<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..\">\
</BODY></FORM></HTML>");

        }

        return szForm;
}


/* FUNCTION: char *MakeOrderStatusForm(int iTermId, int iSyncId, BOOL bInput)
 *
 * PURPOSE: This function makes the HTML based Order Status Form
 *
 * ARGUMENTS:   int     iTermId client browser terminal id
 *          int     iSyncId client browser sync id
 *          BOOL    bInput  TRUE if form is being constructed for input
 *                  else FALSE
 *
 * RETURNS:     char *          A pointer to buffer inside client structure
 *                  where HTML form is built.
 *
 * COMMENTS:    The internal client buffer is created when the terminal id is
 *          assigned and should not be freed except when the client
 *          terminal id is no longer needed.
 */
static char *MakeOrderStatusForm(int iTermId, int iSyncId, BOOL bInput)
{
    char     *szForm;
//  char    c_first[98];
//  char    c_middle[14];
//  char    c_last[98];
    int      i;

    szForm = (char *)Term->pClientData[iTermId].szBuffer;

    Term->pClientData[iTermId].trans.orderStatusData.ordin.w_id = Term-
>pClientData[iTermId].w_id;

        if ( bInput )
        {
                strcpy(szForm,  "<HTML><HEAD><TITLE>TPC-C Order-
Status</TITLE></HEAD><BODY>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"PI*\" VALUE=\"\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"6\">"); //
ORDER_STATUS_FORM = 6

                wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">",
                                                iTermId,
                                                iSyncId);

                strcat(szForm,  "<PRE>                  Order-Status<BR>" );
                wsprintf(szForm+strlen(szForm), "Warehouse: %6.6d   ", Term-
>pClientData[iTermId].trans.orderStatusData.ordin.w_id);

                strcat(szForm,  "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\">\
</BODY></FORM></HTML>" );
                strcat(szForm,
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
                );
```


Copyright © 2004 NEC Corporation

```
        }
        else // Not bInput
        {
                strcpy(szForm,  "<HTML><HEAD><TITLE>TPC-C Order-
Status</TITLE></HEAD><BODY>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"6\">"); //
ORDER_STATUS_FORM = 6

                wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">",
                                        iTermId,
                                        iSyncId);
        strcat(szForm,  "<PRE>                  Order-Status<BR>" );

        sprintf(szForm+strlen(szForm), "Warehouse: %6.6d   District: %2.2d<BR>\
Customer: %4.4d   Name: %-16s %-2s %-16s<BR>\
Cust-Balance: $%9.2f<BR><BR>\
Order-Number: %8.8d   Entry-Date: %19s                  Carrier-
Number: %2.2d<BR>",
                        Term-
>pClientData[iTermId].trans.orderStatusData.ordin.w_id,
                        Term-
>pClientData[iTermId].trans.orderStatusData.ordin.d_id,
                        Term-
>pClientData[iTermId].trans.orderStatusData.ordout.c_id,
                        Term-
>pClientData[iTermId].trans.orderStatusData.ordout.c_first,
                        Term-
>pClientData[iTermId].trans.orderStatusData.ordout.c_middle,
                        Term-
>pClientData[iTermId].trans.orderStatusData.ordout.c_last,
                        Term-
>pClientData[iTermId].trans.orderStatusData.ordout.c_balance,
                        Term-
>pClientData[iTermId].trans.orderStatusData.ordout.o_id,
                        Term-
>pClientData[iTermId].trans.orderStatusData.ordout.o_entry_d,
                        Term-
>pClientData[iTermId].trans.orderStatusData.ordout.o_carrier_id);

                strcat(szForm+strlen(szForm), "Supply-W     Item-Id    Qty
Amount      Delivery-Date<BR>");

                for(i=0; i<Term-
>pClientData[iTermId].trans.orderStatusData.ordout.o_ol_cnt; i++)
                {
                        sprintf(szForm+strlen(szForm), "  %6.6d     %6.6d     %2.2d
$%8.2f     %10s           <BR>",
                                Term-
>pClientData[iTermId].trans.orderStatusData.ordout.ol_supply_w_id[i],
                                Term-
>pClientData[iTermId].trans.orderStatusData.ordout.ol_i_id[i],
                                Term-
>pClientData[iTermId].trans.orderStatusData.ordout.ol_quantity[i],
                                Term-
>pClientData[iTermId].trans.orderStatusData.ordout.ol_amount[i],
                                Term-
>pClientData[iTermId].trans.orderStatusData.ordout.ol_delivery_d[i]);

                }

                strcat(szform,  "<BR></PRE><HR><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..\">\
</BODY></FORM></HTML>" );
                        strcat(szform,
"                                                       "
"                                                       "
"                                                       "
                );
        }

        return szform;
}

/* FUNCTION: char *MakeDeliveryForm(int iTermId, int iSyncId, BOOL bInput,
char *execution_status)
 *
 * PURPOSE:     This function puts out the HTML for the delivery form
 *
```


Copyright © 2004 NEC Corporation

```
 * ARGUMENTS:   int     iTermId client browser terminal id
 *              int     iSyncId client browser sync id
 *              BOOL    bInput  TRUE if form is being constructed for input
 *                      else FALSE
 *
 * RETURNS:     char *  A pointer to buffer inside client structure where HTML
 *                      form is built.
 *
 * COMMENTS: The internal client buffer is created when the terminal id is
 *           assigned and should not be freed except when the client terminal
 *           id is no longer needed.
 */
static char *MakeDeliveryForm(int iTermId, int iSyncId, BOOL bInput, char
*execution_status)
{

    char      *szForm;


    szForm = (char *)Term->pClientData[iTermId].szBuffer;
    Term->pClientData[iTermId].trans.deliveryData.delin.w_id = Term-
>pClientData[iTermId].w_id;

        if ( bInput )
        {
                strcpy( szForm, "<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD><BODY>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"PI*\" VALUE=\"\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"5\">"); //
DELIVERY_FORM = 5

                wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">",
                                                        iTermId,
                                                        iSyncId);

                strcat(szform,  "<PRE>                  Delivery<BR>" );

                wsprintf(szForm+strlen(szForm), "Warehouse: %6.6d<BR><BR>",
Term->pClientData[iTermId].trans.deliveryData.delin.w_id);
                strcat( szForm, "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\">\
</BODY></FORM></HTML>" );
                strcat(szForm,
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
                );

        }
        else // Not bInput
        {
                strcpy( szForm, "<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD><BODY>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"5\">"); //
DELIVERY_FORM = 5

                wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">",
                                                        iTermId,
                                                        iSyncId);
```


Copyright © 2004 NEC Corporation

```
        strcat(szForm,  "<PRE>                          Delivery<BR>" );

        wsprintf(szForm+strlen(szForm), "Warehouse: %6.6d<BR><BR>", Term-
>pClientData[iTermId].trans.deliveryData.delin.w_id);


                wsprintf(szForm+strlen(szForm), "Carrier
Number: %2.2d<BR><BR>\
Execution Status: %25.25s<BR></PRE>",
                                                        Term-
>pClientData[iTermId].trans.deliveryData.delin.o_carrier_id,
                                                        execution_status);

                strcat(szForm,  "<HR><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..\">\
</BODY></FORM></HTML>" );

                strcat(szForm,
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
"                                                       "
                );
        }

        return szForm;
}
```

```
/* FUNCTION: void UtilStrCpy(char * pDest, char * pSrc, int n)
 *
 * PURPOSE: This function copies n characters from string pSrc to pDst and
 *          places a null character at the end of the destination string.
 *
 * ARGUMENTS:   char    *pDest  destination string pointer
 *              char    *pSrc   source string pointer
 *              int     n       number of characters to copy
 *
 * RETURNS:     None
 *
 * COMMENTS: Unlike strncpy this function ensures that the result string is
 *           always null terminated.
 *
 */

static void UtilStrCpy(char * pDest, char * pSrc, int n)
{
   strncpy(pDest, pSrc, n);
   pDest[n] = '\0';

   return;
}


/* FUNCTION: void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK
*pECB, int
 *                              iTermId, int iSyncId)
 *
 * PURPOSE: This function gets and validates the input data from the new order
 * form filling in the required input variables. it then calls the SQLNewOrder
 * transaction, constructs the output form and writes it back to client browser.
 *
 * ARGUMENTS:   EXTENSION_CONTROL_BLOCK *pECB   passed in structure
pointer
 *                                  from inetsrv.
 *      int     iTermId client browser terminal id
 *      int     iSyncId client browser sync id
 *
 * RETURNS:     None
 *
```


Copyright © 2004 NEC Corporation

```
 * COMMENTS:    None
 *
 */

static void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
#ifdef TOPEND
     int                 i,iRc;
     int msglen;
     tp_dif_structs_t       *tclient_dif;
     void space_fill();
#else  // assume tuxedo at this point
          int                 iRc, iError;
          long ilen, *olen;
          char buf[40];
#endif

#ifdef LOCAL_ALLOC
     NewOrderData *newOrderDataPtr;
#endif


#ifdef TUX

    // write_err_log(pECB->r,"in new order");

    if((iRc = ThrTpInit(pECB)) <0)
    {
          // This is bad
                    {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessNewOrder Thread %d iTermId %d
Failed ThrTpInit \r\n",
                                        GetCurrentThreadId(), iTermId );
                              fclose(fp);
                    }
                    sprintf(buf, "NewOrder ThrTpInit < 0 >> Thread %d",
GetCurrentThreadId());
          ErrorMessage(pECB, ERR_TPINIT_BAD, ERR_TYPE_TUXEDO, buf, 0,
0);
          return;//change 2003/7/11
    }

#endif

     memset(&Term->pClientData[iTermId].trans.newOrderData, 0,
sizeof(NewOrderData));
     Term->pClientData[iTermId].trans.newOrderData.newin.w_id = Term-
>pClientData[iTermId].w_id;

     if ( (iError=GetNewOrderData(pECB->lpszQueryString, &Term-
>pClientData[iTermId].trans.newOrderData)) != ERR_SUCCESS )
          {
                    ErrorMessage(pECB, iError, ERR_TYPE_WEBDLL, NULL,
iTermId, iSyncId);
                    return;
          }

#ifdef TOPEND

   tclient_dif = (tp_dif_structs_t *)Term->pClientData[iTermId].tp_structadr;

  // Input top end neworder data
  /* Initialize Top End for tpcc transactions */
     tclient_dif->info->tp_user_dialogue_id = iTermId;
     if (bGeneric)
          space_fill(tclient_dif->service->tp_function_name,
"all_txn",TP_FUNC_NAME_LEN);
     else
          space_fill(tclient_dif->service->tp_function_name,
"neworder",TP_FUNC_NAME_LEN);

     space_fill(tclient_dif->service->tp_product_name, "tpcc",
TP_PROD_NAME_LEN);
     space_fill(tclient_dif->input_format->tp_format_name, "",
TP_FMT_NAME_LEN);
     space_fill(tclient_dif->input_format->tp_format_language, "", TP_LANG_LEN);
     space_fill(tclient_dif->input_format->tp_format_type, "", TP_FMTTYPE_LEN);
     tclient_dif->service->tp_function_qualifier = 0;
     msglen = sizeof(CLIENTDATA) - TOPEND_STRUCT_OFFSET;

     Term->pClientData[iTermId].type = 'N';
     if ((i = tp_mt_client_send(tclient_dif->info, tclient_dif->service, tclient_dif-
>input_format, msglen,  (char *)&Term->pClientData[iTermId].type)) != TP_OK)
     {
```


Copyright © 2004 NEC Corporation

```
        if (bLog)
            {
             FILE *fp1;
             fp1 = fopen(szTpccLogPath, "ab");
             fprintf(fp1, "tp_mt_client_send returned error status = %d\n, i");
             fclose(fp1);
            }

        ErrorMessage(pECB, TOPEND_SEND_ERROR, ERR_TYPE_WEBDLL,
NULL, iTermId, iSyncId);
        return;
    }

    msglen = sizeof(CLIENTDATA) - TOPEND_STRUCT_OFFSET;

    if ((i = tp_mt_client_receive(tclient_dif->info, wait_time, tclient_dif-
>output_format, tclient_dif->service, tclient_dif->location, &msglen, (char *)&Term-
>pClientData[iTermId].type)) != TP_OK)
    {
        if (bLog)
            {
             FILE *fp1;
             fp1 = fopen(szTpccLogPath, "ab");
             fprintf(fp1, "Received error status = %d from client receive for neworder
transaction\n", i);
             fclose(fp1);
            }

        ErrorMessage(pECB, TOPEND_RECEIVE_ERROR,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
            return;
    }


    iRc = Term->pClientData[iTermId].retval


#else    // assume we are running tuxedo at this point
  #ifdef LOCAL_ALLOC    // using local alloc method for tuxedo


        if ((newOrderDataPtr = (NewOrderData *) tpalloc("CARRAY", NULL,
sizeof(NewOrderData))) == NULL)
        {
                iRc = tperrno;
                {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* ProcessNewOrder Thread %d iTermId %d
tpalloc tperrno %d \r\n",
                                GetCurrentThreadId(), iTermId, iRc);
                        fclose(fp);
                }
                sprintf(buf, "NewOrder Tpalloc %d", iRc);
                ErrorMessage(pECB, ERR_TPALLOC_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
                tpfree((char *)newOrderDataPtr);
                return;
        }

        *newOrderDataPtr = Term->pClientData[iTermId].trans.newOrderData;
         ilen = sizeof(NewOrderData);
         olen = &ilen;

    if ( dLog )
        {
                FILE *fp;

                fp = fopen(szTpccLogPath, "ab");
                fprintf(fp, "* ProcessNewOrder Thread %d iTermId %d
newOrderDataPtr: %x \r\n",
                        GetCurrentThreadId(), iTermId, &newOrderDataPtr);
                fclose(fp);
        }

    if (tpcall("OPSTUXSERVER", (char *)newOrderDataPtr, ilen, (char
**)&newOrderDataPtr, (long *)olen, TPSIGRSTRT) == -1)
        {
      iRc = tperrno;
                {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* ProcessNewOrder Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                GetCurrentThreadId(), iTermId, iRc);
```


Copyright © 2004 NEC Corporation

```
                        fclose(fp);
                }
                sprintf(buf, "Neworder tpcall %d", iRc);
        ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf,
iTermId, iSyncId);
                tpfree((char *)newOrderDataPtr);
    return;
  }

        if ( dLog )
        {
                FILE *fp;

                fp = fopen(szTpccLogPath, "ab");
                fprintf(fp, "** ProcessNewOrder Thread %d iTermId %d
newOrderDataPtr: %x \r\n",
                        GetCurrentThreadId(), iTermId, &newOrderDataPtr);
                fclose(fp);
        }


  Term->pClientData[iTermId].trans.newOrderData = *newOrderDataPtr;

  iRc = newOrderDataPtr->retval;

  tpfree((char *)newOrderDataPtr);


 #else          // using global alloc method for tuxedo

  *Term->pClientData[iTermId].newOrderDataPtr = Term-
>pClientData[iTermId].trans.newOrderData;

  ilen = sizeof(NewOrderData);
  olen = &ilen;

        if ( dLog )
        {
                FILE *fp;

                fp = fopen(szTpccLogPath, "ab");
                fprintf(fp, "* ProcessNewOrder Thread %d iTermId %d
NewOrderDataPtr: %x \r\n",
                        GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].newOrderDataPtr);
                fclose(fp);
        }

  if (tpcall("NEWORDER", (char *)Term->pClientData[iTermId].newOrderDataPtr,
ilen,
                (char **)&Term->pClientData[iTermId].newOrderDataPtr, (long
*)olen, TPSIGRSTRT) == -1)
        {
        iRc = tperrno;
                {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* ProcessNewOrder Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                GetCurrentThreadId(), iTermId, iRc);
                        fclose(fp);
                }
                sprintf(buf, "Neworder tpcall %d", iRc);
        ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf,
iTermId, iSyncId);
//              tpfree((char *)newOrderDataPtr);
        return;
  }

        if ( dLog )
        {
                FILE *fp;

                fp = fopen(szTpccLogPath, "ab");
                fprintf(fp, "** ProcessNewOrder Thread %d iTermId %d
NewOrderDataPtr: %x \r\n",
                        GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].newOrderDataPtr);
                fclose(fp);
        }

  Term->pClientData[iTermId].trans.newOrderData = *Term-
>pClientData[iTermId].newOrderDataPtr;

  iRc = Term->pClientData[iTermId].newOrderDataPtr->retval;

 #endif
```


Copyright © 2004 NEC Corporation

```
#endif

        if(bLog){
                FILE *fp1;
                fp1 = fopen(szTpccLogPath, "ab");
                fprintf(fp1, ">> ProcessNewOrder iRc = %d \r\n", iRc);
                fclose(fp1);
        }

        if ( iRc < 0)
   {
        ErrorMessage(pECB, ERR_NEW_ORDER_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
   }


        if (iRc == 0)
        WriteZString(pECB, MakeNewOrderForm(iTermId, iSyncId, FALSE,
FALSE, "Item number is not valid"));
        else

       WriteZString(pECB, MakeNewOrderForm(iTermId, iSyncId, FALSE, TRUE,
"transaction committed"));

        return;

}


/* FUNCTION: void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId,
 *                          int iSyncId)
 *
 * PURPOSE: This function gets and validates the input data from the payment
 *        form filling in the required input variables. It then calls the
 *        SQLPayment transaction, constructs the output form and writes it
 *        back to client browser.
 *
 * ARGUMENTS:   EXTENSION_CONTROL_BLOCK *pECB   passed in structure
pointer
 *                              from inetsrv.
 *     int    iTermId                  client browser terminal id
 *     int    iSyncId                  client browser sync id
 *
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */

static void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
#ifdef TOPEND
    int                 i,iRc;
    tp_dif_structs_t       *tclient_dif;
    int msglen;
#else // assume tuxedo at this point
            int                 iRc, iError;
            long           ilen, *olen;
//      PECBINFO       pEcbInfo;
            char buf[40];
#endif

#ifdef LOCAL_ALLOC
            PaymentData *paymentDataPtr;
#endif


#ifdef TUX
    if((iRc = ThrTpInit(pECB)) <0)
    {
            // This is bad
                    {
                            FILE *fp;

                            fp = fopen(szTpccLogPath, "ab");
                            fprintf(fp, "* ProcessPayment Thread %d iTermId %d
Failed ThrTpInit \r\n",
                                    GetCurrentThreadId(), iTermId );
                            fclose(fp);
                    }
                    sprintf(buf, "Payment ThrTpInit < 0 >> Thread %d",
GetCurrentThreadId());
            ErrorMessage(pECB, ERR_TPINIT_BAD, ERR_TYPE_TUXEDO, buf, 0,
0);
            return;//change 2003/7/11
```


Copyright © 2004 NEC Corporation

```
    }
#endif


    memset(&Term->pClientData[iTermId].trans.paymentData, 0,
sizeof(PaymentData));
    Term->pClientData[iTermId].trans.paymentData.payin.w_id = Term-
>pClientData[iTermId].w_id;

    if ( (iError=GetPaymentData(pECB->lpszQueryString, &Term-
>pClientData[iTermId].trans.paymentData)) !=ERR_SUCCESS )
    {
        ErrorMessage(pECB, iError, ERR_TYPE_WEBDLL, NULL, iTermId,
iSyncId);
        return;
    }

   //  TPCC_LOG("after get payment data");
#ifdef TOPEND

    tclient_dif = (tp_dif_structs_t *)Term->pClientData[iTermId].tp_structadr;

    // Input topend information for call to process payment form

    tclient_dif->info->tp_user_dialogue_id = iTermId;

    if (bGeneric)
        space_fill(tclient_dif->service->tp_function_name,
"all_txn",TP_FUNC_NAME_LEN);
    else
        space_fill(tclient_dif->service->tp_function_name,
"payment",TP_FUNC_NAME_LEN);

    space_fill(tclient_dif->service->tp_product_name, "tpcc",
TP_PROD_NAME_LEN);
    space_fill(tclient_dif->input_format->tp_format_name, "",
TP_FMT_NAME_LEN);
    space_fill(tclient_dif->input_format->tp_format_language, "", TP_LANG_LEN);
    space_fill(tclient_dif->input_format->tp_format_type, "", TP_FMTTYPE_LEN);
    tclient_dif->service->tp_function_qualifier = 0;
    msglen = sizeof(CLIENTDATA) - TOPEND_STRUCT_OFFSET;

    Term->pClientData[iTermId].type = 'P';
        // tp_system_log_text("ProcessPayment:","before tp_mt_client_send");

    if ((i = tp_mt_client_send(tclient_dif->info, tclient_dif->service,
        tclient_dif->input_format, msglen, (char *)&Term-
>pClientData[iTermId].type)) != TP_OK)
    {

        ErrorMessage(pECB, TOPEND_SEND_ERROR, ERR_TYPE_WEBDLL,
NULL, iTermId, iSyncId);

        if (bLog)
        {
          FILE *fp1;
          fp1 = fopen(szTpccLogPath, "ab");
          fprintf(fp1, "Received error status = %d on payment send fnt\n", i);
          fclose(fp1);
        }
        return;
    }


    msglen = sizeof(CLIENTDATA) - TOPEND_STRUCT_OFFSET;
    // tp_system_log_text("ProcessPayment:","before tp_mt_client_receive");
    if ((i = tp_mt_client_receive(tclient_dif->info, wait_time,
        tclient_dif->output_format, tclient_dif->service,
        tclient_dif->location, &msglen, (char *)&Term-
>pClientData[iTermId].type)) != TP_OK)
    {
                //tp_system_log_text("ProcessPayment:","Error on
tp_mt_client_receive");
        if (bLog)
        {
          FILE *fp1;
          fp1 = fopen(szTpccLogPath, "ab");
          fprintf(fp1, "Received error status = %d from client receive for payment
transaction\n", i);
          fclose(fp1);
        }

        ErrorMessage(pECB, TOPEND_RECEIVE_ERROR,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }
```


Copyright © 2004 NEC Corporation

```
    iRc = Term->pClientData[iTermId].retval;


#else  // assume tuxedo running here

 #ifdef LOCAL_ALLOC

            if ((paymentDataPtr = (PaymentData *) tpalloc("CARRAY", NULL,
sizeof(PaymentData))) == NULL)
            {
                        iRc = tperrno;
                        {
                                    FILE *fp;

                                    fp = fopen(szTpccLogPath, "ab");
                                    fprintf(fp, "* ProcessPayment Thread %d iTermId %d
tpalloc tperrno %d \r\n",
                                    GetCurrentThreadId(), iTermId, iRc);
                                    fclose(fp);
                        }
                        sprintf(buf, "Payment Tpalloc %d", iRc);
                        ErrorMessage(pECB, ERR_TPALLOC_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
                        tpfree((char *)paymentDataPtr);
                        return;
            }

   *paymentDataPtr= Term->pClientData[iTermId].trans.paymentData;

    ilen = sizeof(PaymentData);
    olen = &ilen;

            if ( dLog )
            {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* ProcessPayment Thread %d iTermId %d
paymentDataPtr: %x \r\n",
                                    GetCurrentThreadId(), iTermId, &paymentDataPtr);
                        fclose(fp);
            }

    if (tpcall("OPSTUXSERVER", (char *)paymentDataPtr, ilen, (char
**)&paymentDataPtr, (long *)olen, TPSIGRSTRT) == -1)
            {
        iRc = tperrno;
                        {
                                    FILE *fp;

                                    fp = fopen(szTpccLogPath, "ab");
                                    fprintf(fp, "* ProcessPayment Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                                GetCurrentThreadId(), iTermId, iRc);
                                    fclose(fp);
                        }
                        sprintf(buf, "Payment tpcall %d", iRc);
        ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf,
iTermId, iSyncId);
        tpfree((char *)paymentDataPtr);

        return;
    }

            if ( dLog )
            {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "** ProcessPayment Thread %d iTermId %d
paymentDataPtr: %x \r\n",
                                    GetCurrentThreadId(), iTermId, &paymentDataPtr);
                        fclose(fp);
            }

    Term->pClientData[iTermId].trans.paymentData = *paymentDataPtr;

    iRc = paymentDataPtr->retval;

            tpfree((char *)paymentDataPtr);

 #else

    *Term->pClientData[iTermId].paymentDataPtr = Term-
>pClientData[iTermId].trans.paymentData;
```


Copyright © 2004 NEC Corporation

```
    ilen = sizeof(PaymentData);
    olen = &ilen;

            if ( dLog )
            {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "* ProcessPayment Thread %d iTermId %d
PaymentDataPtr: %x \r\n",
                            GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].paymentDataPtr);
                    fclose(fp);
            }

    if (tpcall("PAYMENT", (char *)Term->pClientData[iTermId].paymentDataPtr, ilen,
                    (char **)&Term->pClientData[iTermId].paymentDataPtr, (long
*)olen, TPSIGRSTRT) == -1)
            {
            iRc = tperrno;
                    {
                            FILE *fp;

                            fp = fopen(szTpccLogPath, "ab");
                            fprintf(fp, "* ProcessPayment Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                    GetCurrentThreadId(), iTermId, iRc);
                            fclose(fp);
                    }
                    sprintf(buf, "Payment tpcall %d", iRc);
            ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf,
iTermId, iSyncId);
//          tpfree((char *)PaymentDataPtr);
            return;
    }

            if ( dLog )
            {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "** ProcessPayment Thread %d iTermId %d
PaymentDataPtr: %x \r\n",
                            GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].paymentDataPtr);
                    fclose(fp);
            }

    Term->pClientData[iTermId].trans.paymentData = *Term-
>pClientData[iTermId].paymentDataPtr;

    iRc = Term->pClientData[iTermId].paymentDataPtr->retval;

  #endif
#endif


    if ( iRc == 0 )
            ErrorMessage(pECB, ERR_PAYMENT_INVALID_CUSTOMER,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
    else if ( iRc < 0 )
            ErrorMessage(pECB, ERR_PAYMENT_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
    else
    {
            WriteZString(pECB, MakePaymentForm(iTermId, iSyncId, FALSE) );
    }

    return;

}


/* FUNCTION: void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK
*pECB, int
 *                          iTermId, int iSyncId)
 *
 * PURPOSE:This function gets and validates the input data from the Order Status
 *      form filling in the required input variables. It then calls the
 *      SQLOrderStatus transaction, constructs the output form and writes it
 *      back to client browser.
 *
 * ARGUMENTS:   EXTENSION_CONTROL_BLOCK *pECB   passed in structure
pointer from
 *                          inetsrv.
 *          int             iTermId client browser terminal id
 *          int             iSyncId client browser sync id
 *
```


Copyright © 2004 NEC Corporation

```
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */
static void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
#ifdef TOPEND
     int                 i,iRc;
     int                 iError;
     void space_fill();
     tp_dif_structs_t       *tclient_dif;
     int msglen;
#else  // assume tuxedo
            int                 iRc, iError;
            long ilen, *olen;
//      PECBINFO       pEcbInfo;
            char buf[40];
#endif

#ifdef LOCAL_ALLOC
            OrderStatusData *orderStatusDataPtr;
#endif

#ifdef TUX

    if((iRc = ThrTpInit(pECB)) <0)
    {
             // This is bad
                        {
                                    FILE *fp;

                                    fp = fopen(szTpccLogPath, "ab");
                                    fprintf(fp, "* ProcessOrderStatus Thread %d iTermId %d
Failed ThrTpInit \r\n",
                                                GetCurrentThreadId(), iTermId );
                                    fclose(fp);
                        }
                        sprintf(buf, "OrderStatus ThrTpInit < 0 >> Thread %d",
GetCurrentThreadId());
            ErrorMessage(pECB, ERR_TPINIT_BAD, ERR_TYPE_TUXEDO, buf, 0,
0);
            return;//change 2003/7/11
    }

#endif

     memset(&Term->pClientData[iTermId].trans.orderStatusData, 0,
sizeof(OrderStatusData));
     Term->pClientData[iTermId].trans.orderStatusData.ordin.w_id = Term-
>pClientData[iTermId].w_id;
     if ( (iError=GetOrderStatusData(pECB->lpszQueryString,
                &Term->pClientData[iTermId].trans.orderStatusData)) !=
ERR_SUCCESS )
     {
            ErrorMessage(pECB, iError, ERR_TYPE_WEBDLL, NULL, iTermId,
iSyncId);
            return;
     }

#ifdef TOPEND

     tclient_dif = (tp_dif_structs_t *)Term->pClientData[iTermId].tp_structadr;

     /* Initialize Top End for tpcc transactions */
     tclient_dif->info->tp_user_dialogue_id = iTermId;

     if (bGeneric)
             space_fill(tclient_dif->service->tp_function_name,
"all_txn",TP_FUNC_NAME_LEN);
     else
             space_fill(tclient_dif->service->tp_function_name,
"ordstat",TP_FUNC_NAME_LEN);

     space_fill(tclient_dif->service->tp_product_name, "tpcc",
TP_PROD_NAME_LEN);
     space_fill(tclient_dif->input_format->tp_format_name, "",
TP_FMT_NAME_LEN);
     space_fill(tclient_dif->input_format->tp_format_language, "", TP_LANG_LEN);
     space_fill(tclient_dif->input_format->tp_format_type, "", TP_FMTTYPE_LEN);
     tclient_dif->service->tp_function_qualifier = 0;
     msglen = sizeof(CLIENTDATA) - TOPEND_STRUCT_OFFSET;

     Term->pClientData[iTermId].type = 'O';

     if ((i = tp_mt_client_send(tclient_dif->info, tclient_dif->service,
             tclient_dif->input_format, msglen, (char *)&Term-
```


Copyright © 2004 NEC Corporation

```
>pClientData[iTermId].type)) != TP_OK)
    {

        if (bLog)
        {
          FILE *fp1;
          fp1 = fopen(szTpccLogPath, "ab");
          fprintf(fp1, "Received error status = %d on client send for order status
transaction\n", i);
          fclose(fp1);
        }

        ErrorMessage(pECB, TOPEND_SEND_ERROR, ERR_TYPE_WEBDLL,
NULL, iTermId, iSyncId);
        return;
    }

    msglen = sizeof(CLIENTDATA) - TOPEND_STRUCT_OFFSET;

    if ((i = tp_mt_client_receive(tclient_dif->info, wait_time, tclient_dif-
>output_format, tclient_dif->service, tclient_dif->location, &msglen, (char *)&Term-
>pClientData[iTermId].type)) != TP_OK)
    {
        if (bLog)
        {
          FILE *fp1;
          fp1 = fopen(szTpccLogPath, "ab");
          fprintf(fp1, "Received error status = %d from client receive for order status
transaction\n", i);
          fclose(fp1);
        }
        fprintf(stderr, "Received error status = %d from client receive for order status
transaction\n", i);
        ErrorMessage(pECB, TOPEND_RECEIVE_ERROR,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    iRc = Term->pClientData[iTermId].retval;


#else

 #ifdef LOCAL_ALLOC

        if ((orderStatusDataPtr = (OrderStatusData *) tpalloc("CARRAY", NULL,
sizeof(OrderStatusData))) == NULL)
        {
                iRc = tperrno;
                {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* ProcessOrderStatus Thread %d iTermId %d
tpalloc tperrno %d \r\n",
                        GetCurrentThreadId(), iTermId, iRc);
                        fclose(fp);
                }
                sprintf(buf, "OrderStatus Tpcalloc %d", iRc);
                ErrorMessage(NULL, ERR_TPALLOC_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
                tpfree((char *)orderStatusDataPtr);
                return;
        }

    *orderStatusDataPtr = Term->pClientData[iTermId].trans.orderStatusData;

    ilen = sizeof(OrderStatusData);
    olen = &ilen;

        if ( dLog )
        {
                FILE *fp;

                fp = fopen(szTpccLogPath, "ab");
                fprintf(fp, "* ProcessOrderStatus Thread %d iTermId %d
orderStatusDataPtr: %x \r\n",
                        GetCurrentThreadId(), iTermId, &orderStatusDataPtr);
                fclose(fp);
        }

    if (tpcall("OPSTUXSERVER", (char *)orderStatusDataPtr, ilen, (char
**)&orderStatusDataPtr, (long *)olen, TPSIGRSTRT) == -1)
        {
      iRc = tperrno;
                {
                        FILE *fp;
```


Copyright © 2004 NEC Corporation

```
                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* ProcessOrderStatus Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                GetCurrentThreadId(), iTermId, iRc);
                        fclose(fp);
                }
                sprintf(buf, "OrderStatus tpcall %d", iRc);
        ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf,
iTermId, iSyncId);
        tpfree((char *)orderStatusDataPtr);

        return;
    }

        if ( dLog )
        {
                FILE *fp;

                fp = fopen(szTpccLogPath, "ab");
                fprintf(fp, "** ProcessOrderStatus Thread %d iTermId %d
orderStatusDataPtr: %x \r\n",
                        GetCurrentThreadId(), iTermId, &orderStatusDataPtr);
                fclose(fp);
        }

    Term->pClientData[iTermId].trans.orderStatusData = *orderStatusDataPtr;

    iRc = orderStatusDataPtr->retval;

        tpfree((char *)orderStatusDataPtr);



  #else

    *Term->pClientData[iTermId].orderStatusDataPtr = Term-
>pClientData[iTermId].trans.orderStatusData;

    ilen = sizeof(OrderStatusData);
    olen = &ilen;

        if ( dLog )
        {
                FILE *fp;

                fp = fopen(szTpccLogPath, "ab");
                fprintf(fp, "* ProcessOrderStatus Thread %d iTermId %d
OrderStatusDataPtr: %x \r\n",
                        GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].orderStatusDataPtr);
                fclose(fp);
        }

    if (tpcall("ORDERSTATUS", (char *)Term-
>pClientData[iTermId].orderStatusDataPtr, ilen,
                (char **)&Term->pClientData[iTermId].orderStatusDataPtr, (long
*)olen, TPSIGRSTRT) == -1)
        {
        iRc = tperrno;
                {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* ProcessOrderStatus Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                GetCurrentThreadId(), iTermId, iRc);
                        fclose(fp);
                }
                sprintf(buf, "OrderStatus tpcall %d", iRc);
        ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf,
iTermId, iSyncId);
//      tpfree((char *)Term->pClientData[iTermId].orderStatusDataPtr);
        return;
    }

        if ( dLog )
        {
                FILE *fp;

                fp = fopen(szTpccLogPath, "ab");
                fprintf(fp, "** ProcessOrderStatus Thread %d iTermId %d
OrderStatusDataPtr: %x \r\n",
                        GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].orderStatusDataPtr);
                fclose(fp);
        }
```


Copyright © 2004 NEC Corporation

```
    Term->pClientData[iTermId].trans.orderStatusData = *Term-
>pClientData[iTermId].orderStatusDataPtr;

    iRc = Term->pClientData[iTermId].orderStatusDataPtr->retval;

  #endif
#endif

    if ( iRc == 0 )
            ErrorMessage(pECB, ERR_NOSUCH_CUSTOMER,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
     else if ( iRc < 0 )
            ErrorMessage(pECB, ERR_ORDER_STATUS_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
     else
            WriteZString(pECB, MakeOrderStatusForm(iTermId, iSyncId, FALSE) );

     return;

}


/* FUNCTION: void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK
*pECB,
 *                              int iTermId, int iSyncId)
 *
 * PURPOSE:This function gets and validates the input data from delivery form
 * filling in the required input variables. It then calls the PostDeliveryInfo
 * Api, The client is then informed that the transaction has been posted.
 *
 * ARGUMENTS:   EXTENSION_CONTROL_BLOCK *pECB   passed in structure
pointer from
 *                                  inetsrv.
 *            int                 iTermId client browser terminal id
 *            int                 iSyncId clinet browser sync id
 *
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */

static void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{

#ifdef TOPEND
            char    szTmp[26];
            int     i;
            void space_fill();
            tp_dif_structs_t *tclient_dif;
            int msglen;
#else
          int           iRc;
//      int           iError;
          char    szTmp[26];
          long ilen, *olen;
//      int           tmp;
          char buf[40];
#endif

#ifdef LOCAL_ALLOC
          DeliveryData *deliveryDataPtr;
#endif

#ifdef TUX


   if((iRc = ThrTpInit(pECB)) <0)
    {
            // This is bad
                    {
                            FILE *fp;

                            fp = fopen(szTpccLogPath, "ab");
                            fprintf(fp, "* ProcessDelivery Thread %d iTermId %d
Failed ThrTpInit \r\n",
                                    GetCurrentThreadId(), iTermId );
                            fclose(fp);
                    }
                    sprintf(buf, "Delivery ThrTpInit < 0 >> Thread %d",
GetCurrentThreadId());
          ErrorMessage(pECB, ERR_TPINIT_BAD, ERR_TYPE_TUXEDO, buf, 0,
0);
          return;//change 2003/7/11
    }
```


Copyright © 2004 NEC Corporation

```
#endif

    memset(&Term->pClientData[iTermId].trans.deliveryData, 0,
sizeof(DeliveryData));
    Term->pClientData[iTermId].trans.deliveryData.delin.w_id = Term-
>pClientData[iTermId].w_id;


    if ( !GetKeyValue(pECB->lpszQueryString, "OCD*", szTmp, sizeof(szTmp)) )

    {
        ErrorMessage(pECB,
ERR_DELIVERY_MISSING_OCD_KEY,ERR_TYPE_WEBDLL, NULL, iTermId,
iSyncId);
        return;
    }

     if ( !IsNumeric (szTmp) )
    {
        ErrorMessage(pECB,
ERR_DELIVERY_CARRIER_INVALID,ERR_TYPE_WEBDLL, NULL, iTermId,
iSyncId);
        return;
    }

    Term->pClientData[iTermId].trans.deliveryData.delin.o_carrier_id = atoi(szTmp);
    if ( Term->pClientData[iTermId].trans.deliveryData.delin.o_carrier_id > 10 ||
        Term->pClientData[iTermId].trans.deliveryData.delin.o_carrier_id < 1 )
    {
    ErrorMessage(pECB,
ERR_DELIVERY_CARRIER_ID_RANGE,ERR_TYPE_WEBDLL, NULL, iTermId,
iSyncId);
    return;
    }

    //post delivery info

#ifdef TOPEND

    tclient_dif = (tp_dif_structs_t *)Term->pClientData[iTermId].tp_structadr;

    /* Initialize Top End for tpcc transactions */
    tclient_dif->info->tp_user_dialogue_id = iTermId;
    tclient_dif->info->tp_flags = TP_NO_RESPONSE;
    space_fill(tclient_dif->service->tp_function_name,
"del_bat",TP_FUNC_NAME_LEN);
    space_fill(tclient_dif->service->tp_product_name, "tpcc",
TP_PROD_NAME_LEN);
    space_fill(tclient_dif->input_format->tp_format_name, "",
TP_FMT_NAME_LEN);
    space_fill(tclient_dif->input_format->tp_format_language, "", TP_LANG_LEN);
    space_fill(tclient_dif->input_format->tp_format_type, "", TP_FMTTYPE_LEN);
    tclient_dif->service->tp_function_qualifier = 0;
    msglen = sizeof(CLIENTDATA) - TOPEND_STRUCT_OFFSET;

    Term->pClientData[iTermId].type = 'D';

    /* get the time the entry is put on the queue */
#ifdef WIN32
    _ftime(&timebuffer);
#else
    ftime(&timebuffer);
#endif
    Term->pClientData[iTermId].trans.deliveryData.delin.qtime = timebuffer.time +
timebuffer.millitm/1000.0;

    if ((i = tp_mt_client_send(tclient_dif->info, tclient_dif->service,
        tclient_dif->input_format, msglen, (char *)&Term-
>pClientData[iTermId].type)) != TP_OK)
    {

        if (bLog)
        {
          FILE *fp1;
          fp1 = fopen(szTpccLogPath, "ab");
          fprintf(fp1, "Received error status = %d on client send for delivery
transaction\n", i);
          fclose(fp1);
        }

        ErrorMessage(pECB, TOPEND_SEND_ERROR, ERR_TYPE_WEBDLL,
NULL, iTermId, iSyncId);
        return;
    }

    /* msglen might need to be 0, as not receiving anything put ack from server */
    msglen = sizeof(CLIENTDATA) - TOPEND_STRUCT_OFFSET;
```


Copyright © 2004 NEC Corporation

```
    if ((i = tp_mt_client_receive(tclient_dif->info, wait_time, tclient_dif-
>output_format, tclient_dif->service, tclient_dif->location, &msglen, (char *)&Term-
>pClientData[iTermId].type)) != TP_OK)
    {
        if (bLog)
        {
          FILE *fp1;
          fp1 = fopen(szTpccLogPath, "ab");
          fprintf(fp1, "Received error status = %d from client receive for order status
transaction\n", i);
          fclose(fp1);
        }

        ErrorMessage(pECB, TOPEND_RECEIVE_ERROR,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }
    /* reset flags for next txn */
    tclient_dif->info->tp_flags = TP_NOFLAGS;
#else

 #ifdef LOCAL_ALLOC

        if ((deliveryDataPtr = (DeliveryData *) tpalloc("CARRAY", NULL,
sizeof(DeliveryData))) == NULL)
        {
                iRc = tperrno;
                {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* ProcessDelivery Thread %d iTermId %d
tpalloc tperrno %d \r\n",
                        GetCurrentThreadId(), iTermId, iRc);
                        fclose(fp);
                }
                sprintf(buf, "Delivery Tpcalloc %d", iRc);
                ErrorMessage(NULL, ERR_TPALLOC_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
                tpfree((char *)deliveryDataPtr);

                return;
        }
```

```
        *deliveryDataPtr = Term->pClientData[iTermId].trans.deliveryData;

    ilen = sizeof(DeliveryData);
    olen = &ilen;

        if ( dLog )
        {
                FILE *fp;

                fp = fopen(szTpccLogPath, "ab");
                fprintf(fp, "* ProcessDelivery Thread %d iTermId %d
deliveryDataPtr: %x \r\n",
                        GetCurrentThreadId(), iTermId, &deliveryDataPtr);
                fclose(fp);
        }


    /* get the time the entry is put on the queue */
    ftime(&timebuffer);
    deliveryDataPtr->delin.qtime = timebuffer.time + timebuffer.millitm/1000.0;


    if (tpacall("OPSTUXSERVER", (char *)deliveryDataPtr, ilen,
TPNOREPLY|TPSIGRSTRT) != 0) {
        iRc = tperrno;
                {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* ProcessDelivery Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                GetCurrentThreadId(), iTermId, iRc);
                        fclose(fp);
                }
                sprintf(buf, "Delivery tpcall %d", iRc);
        ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf,
iTermId, iSyncId);
        tpfree((char *)deliveryDataPtr);

        return;
    }
```


Copyright © 2004 NEC Corporation

```
		if ( dLog )
		{
			FILE *fp;

			fp = fopen(szTpccLogPath, "ab");
			fprintf(fp, "** ProcessDelivery Thread %d iTermId %d
deliveryDataPtr: %x \r\n",
				GetCurrentThreadId(), iTermId, &deliveryDataPtr);
			fclose(fp);
		}

		tpfree((char *)deliveryDataPtr);



	#else
	*Term->pClientData[iTermId].deliveryDataPtr = Term-
>pClientData[iTermId].trans.deliveryData;

	ilen = sizeof(DeliveryData);
	olen = &ilen;

		if ( dLog )
		{
			FILE *fp;

			fp = fopen(szTpccLogPath, "ab");
			fprintf(fp, "* ProcessDelivery Thread %d iTermId %d
DeliveryDataPtr: %x \r\n",
				GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].deliveryDataPtr);
			fclose(fp);
		}

	/* get the time the entry is put on the queue */
	ftime(&timebuffer);
	Term->pClientData[iTermId].deliveryDataPtr->delin.qtime = timebuffer.time +
timebuffer.millitm/1000.0;

	if (tpcall("DELIVERY", (char *)Term->pClientData[iTermId].deliveryDataPtr, ilen,
			(char **)&Term->pClientData[iTermId].deliveryDataPtr, (long
*)olen, TPSIGRSTRT) == -1) {
		iRc = tperrno;
		{
			FILE *fp;

			fp = fopen(szTpccLogPath, "ab");
			fprintf(fp, "* ProcessDelivery Thread %d iTermId %d
tpcall tperrno %d \r\n",
				GetCurrentThreadId(), iTermId, iRc);
			fclose(fp);
		}
		sprintf(buf, "Delivery tpcall %d", iRc);
		ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf,
iTermId, iSyncId);
//		tpfree((char *)Term->pClientData[iTermId].deliveryDataPtr);
		return;
	}

		if ( dLog )
		{
			FILE *fp;

			fp = fopen(szTpccLogPath, "ab");
			fprintf(fp, "** ProcessDelivery Thread %d iTermId %d
DeliveryDataPtr: %x \r\n",
				GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].deliveryDataPtr);
			fclose(fp);
		}

	iRc = Term->pClientData[iTermId].deliveryDataPtr->retval;


	#endif
#endif

	WriteZString(pECB, MakeDeliveryForm(iTermId, iSyncId, FALSE, "Delivery has
been queued.")));

	return;
}


/* FUNCTION: void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK
*pECB, int
 *		iTermId, int iSyncId)
```


Copyright © 2004 NEC Corporation

```
 *
 * PURPOSE: This function gets and validates the input data from the Stock Level
 * form filling in the required input variables. It then calls the
 * SQLStockLevel transaction, constructs the output form and writes it
 * back to client browser.
 *
 * ARGUMENTS: EXTENSION_CONTROL_BLOCK   *pECB   passed in structure
pointer from
 *                                    inetsrv.
 *            int                  iTermId client browser terminal id
 *            int                  iSyncId client browser sync id
 *
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */

static void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
#ifdef TOPEND
     char      szTmp[26];
     int                  i,iRc;
     int msglen;
     void space_fill();
     tp_dif_structs_t       *tclient_dif;
#else
            int            iRc;
//      int            iError;
            char           szTmp[26];
//      BOOL           bRc;
            long ilen, *olen;
//      PECBINFO       pEcbInfo;
            char buf[40];
#endif

#ifdef LOCAL_ALLOC
            StockLevelData *stockLevelDataPtr;
#endif


#ifdef TUX
    if((iRc = ThrTpInit(pECB)) <0)
    {
            // This is bad
                    {
                            FILE *fp;

                            fp = fopen(szTpccLogPath, "ab");
                            fprintf(fp, "* ProcessStockLevel Thread %d iTermId %d
Failed ThrTpInit \r\n",
                                    GetCurrentThreadId(), iTermId );
                            fclose(fp);
                    }
                    sprintf(buf, "StockLevel ThrTpInit < 0 >> Thread %d",
GetCurrentThreadId());
            ErrorMessage(pECB, ERR_TPINIT_BAD, ERR_TYPE_TUXEDO, buf, 0,
0);
            return;//change 2003/7/11
    }
#endif

     memset(&Term->pClientData[iTermId].trans.stocklevelData, 0,
sizeof(StockLevelData));
     Term->pClientData[iTermId].trans.stocklevelData.stoin.w_id = Term-
>pClientData[iTermId].w_id;
     Term->pClientData[iTermId].trans.stocklevelData.stoin.d_id = Term-
>pClientData[iTermId].d_id;
     if ( !GetKeyValue(pECB->lpszQueryString, "TT*", szTmp, sizeof(szTmp)) )
     {
            ErrorMessage(pECB,
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,ERR_TYPE_WEBDLL,
NULL, iTermId, iSyncId);
            return;
     }
     if ( !IsNumeric(szTmp) )
     {
            ErrorMessage(pECB,
ERR_STOCKLEVEL_THRESHOLD_INVALID,ERR_TYPE_WEBDLL, NULL,
iTermId, iSyncId);
            return;
     }
     Term->pClientData[iTermId].trans.stocklevelData.stoin.threshold = atoi(szTmp);
     if ( Term->pClientData[iTermId].trans.stocklevelData.stoin.threshold >= 100 ||
              Term->pClientData[iTermId].trans.stocklevelData.stoin.threshold <= 0 )
     {
            ErrorMessage(pECB,
```


Copyright © 2004 NEC Corporation

```
ERR_STOCKLEVEL_THRESHOLD_RANGE,ERR_TYPE_WEBDLL, NULL,
iTermId, iSyncId);
            return;
    }

#ifdef TOPEND

    tclient_dif = (tp_dif_structs_t *)Term->pClientData[iTermId].tp_structadr;

    /* Initialize Top End for tpcc transactions */
    tclient_dif->info->tp_user_dialogue_id = iTermId;

    if (bGeneric)
            space_fill(tclient_dif->service->tp_function_name,
"all_txn",TP_FUNC_NAME_LEN);
    else
            space_fill(tclient_dif->service->tp_function_name,
"stklevel",TP_FUNC_NAME_LEN);

    space_fill(tclient_dif->service->tp_product_name, "tpcc",
TP_PROD_NAME_LEN);
    space_fill(tclient_dif->input_format->tp_format_name, "",
TP_FMT_NAME_LEN);
    space_fill(tclient_dif->input_format->tp_format_language, "", TP_LANG_LEN);
    space_fill(tclient_dif->input_format->tp_format_type, "", TP_FMTTYPE_LEN);
    tclient_dif->service->tp_function_qualifier = 0;
    msglen = sizeof(CLIENTDATA) - TOPEND_STRUCT_OFFSET;

    Term->pClientData[iTermId].type = 'S';

            if ((i = tp_mt_client_send(tclient_dif->info, tclient_dif->service, tclient_dif-
>input_format, msglen, (char *)&Term->pClientData[iTermId].type)) != TP_OK)
    {

            if (bLog)
            {
              FILE *fp1;
              fp1 = fopen(szTpccLogPath, "ab");
              fprintf(fp1, "Received error status = %d from client send for stock level
transaction\n", i);
              fclose(fp1);
            }

            ErrorMessage(pECB, TOPEND_SEND_ERROR, ERR_TYPE_WEBDLL,
NULL, iTermId, iSyncId);
            return;
    }

    msglen = sizeof(CLIENTDATA) - TOPEND_STRUCT_OFFSET;

    if ((i = tp_mt_client_receive(tclient_dif->info, wait_time, tclient_dif-
>output_format, tclient_dif->service, tclient_dif->location, &msglen, (char *)&Term-
>pClientData[iTermId].type)) != TP_OK)
    {
            if (bLog)
            {
              FILE *fp1;
              fp1 = fopen(szTpccLogPath, "ab");
              fprintf(fp1, "Received error status = %d from client receive for stock level
transaction\n", i);
              fclose(fp1);
            }
            ErrorMessage(pECB, TOPEND_RECEIVE_ERROR,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
            return;
    }

    iRc = Term->pClientData[iTermId].retval;

#else

  #ifdef LOCAL_ALLOC

            if ((stockLevelDataPtr = (StockLevelData *) tpalloc("CARRAY", NULL,
sizeof(StockLevelData))) == NULL)
            {
                    iRc = tperrno;
                    {
                            FILE *fp;

                            fp = fopen(szTpccLogPath, "ab");
                            fprintf(fp, "* ProcessStockLevel Thread %d iTermId %d
tpalloc tperrno %d \r\n",
                            GetCurrentThreadId(), iTermId, iRc);
                            fclose(fp);
                    }
                    sprintf(buf, "Tpcalloc %d", iRc);
                    ErrorMessage(NULL, ERR_TPALLOC_BAD,
```


Copyright © 2004 NEC Corporation

```
ERR_TYPE_TUXEDO, buf, 0, 0);
                tpfree((char *)stockLevelDataPtr);

                return;
        }

    *stockLevelDataPtr = Term->pClientData[iTermId].trans.stocklevelData;

    ilen = sizeof(StockLevelData);
    olen = &ilen;

    if (tpcall("OPSTUXSERVER", (char *) stockLevelDataPtr, ilen, (char **)
&stockLevelDataPtr, (long *) olen, TPSIGRSTRT) == -1) {
        iRc = tperrno;
                {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* ProcessStockLevel Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                GetCurrentThreadId(), iTermId, iRc);
                        fclose(fp);
                }
                sprintf(buf, "StockLevel tpcall %d", iRc);
        ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf,
iTermId, iSyncId);
                tpfree((char *)stockLevelDataPtr);

        return;
    }

        if ( dLog )
        {
                FILE *fp;

                fp = fopen(szTpccLogPath, "ab");
                fprintf(fp, "** ProcessStockLevel Thread %d iTermId %d
stockLevelDataPtr: %x \r\n",
                        GetCurrentThreadId(), iTermId, &stockLevelDataPtr);
                fclose(fp);
        }

    Term->pClientData[iTermId].trans.stocklevelData = *stockLevelDataPtr;

    iRc = stockLevelDataPtr->retval;

            tpfree((char *)stockLevelDataPtr);


  #else

    *Term->pClientData[iTermId].stockLevelDataPtr = Term-
>pClientData[iTermId].trans.stocklevelData;

    ilen = sizeof(StockLevelData);
    olen = &ilen;

            if ( dLog )
            {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "* ProcessStockLevel Thread %d iTermId %d
StockLevelDataPtr: %x \r\n",
                            GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].stockLevelDataPtr);
                    fclose(fp);
            }

    if (tpcall("STOCKLEVEL", (char *)Term-
>pClientData[iTermId].stockLevelDataPtr, ilen,
                    (char **)&Term->pClientData[iTermId].stockLevelDataPtr, (long
*) olen, TPSIGRSTRT) == -1) {
            iRc = tperrno;
                    {
                            FILE *fp;

                            fp = fopen(szTpccLogPath, "ab");
                            fprintf(fp, "* ProcessStockLevel Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                    GetCurrentThreadId(), iTermId, iRc);
                            fclose(fp);
                    }
                    sprintf(buf, "StockLevel tpcall %d", iRc);
            ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf,
iTermId, iSyncId);
//              tpfree((char *)StockLevelDataPtr);
```


Copyright © 2004 NEC Corporation

```
            return;
    }

            if ( dLog )
            {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "** ProcessStockLevel Thread %d iTermId %d
StockLevelDataPtr: %x \r\n",
                            GetCurrentThreadId(), iTermId, &Term-
>pClientData[iTermId].stockLevelDataPtr);
                    fclose(fp);
            }

    Term->pClientData[iTermId].trans.stocklevelData = *Term-
>pClientData[iTermId].stockLevelDataPtr;

    iRc = Term->pClientData[iTermId].stockLevelDataPtr->retval;

  #endif
#endif

    if ( iRc  < 0) {
     ErrorMessage(pECB, ERR_STOCKLEVEL_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
    }
    else {
     WriteZString(pECB, MakeStockLevelForm(iTermId, iSyncId, FALSE) );
    }

     return;
}


/* FUNCTION: int GetNewOrderData(LPSTR lpszQueryString, NewOrderData
*pNewOrderData)
 *
 * PURPOSE: This function extracts and validates the new order form data from
 *          an http command string.
 *
 * ARGUMENTS: LPSTR     lpszQueryString client browser http command string
 *            NewOrderData  *pNewOrderData pointer to new order data structure
 *
 * RETURNS:     int            error code indicating reason for failure
 *              ERR_SUCCESS     new order input data successfully parsed
 *
 *
 * COMMENTS:    None
 *
 */
static int GetNewOrderData(LPSTR lpszQueryString, NewOrderData
*pNewOrderData)
{
            char    szTmp[26];
            int             i,j,skipped;
            short   items;

//          BOOL    bCheck;
//          char *  s;

//          s=lpszQueryString;

            if ( !GetKeyValue(lpszQueryString, "DID*", szTmp, sizeof(szTmp)) )
                    return ERR_NEWORDER_FORM_MISSING_DID;

            if ( !IsNumeric(szTmp) )
                    return ERR_NEWORDER_DISTRICT_INVALID;


            pNewOrderData->newin.d_id = atoi(szTmp);

//          if ( pNewOrderData->newin.d_id > 10 || pNewOrderData->newin.d_id < 1)
//                  return ERR_NEWORDER_DISTRICT_INVALID;

            if ( !GetKeyValue(lpszQueryString, "CID*", szTmp, sizeof(szTmp)) )
                    return ERR_NEWORDER_CUSTOMER_KEY;

            if ( !IsNumeric(szTmp) )
                    return ERR_NEWORDER_CUSTOMER_INVALID;

    pNewOrderData->newin.c_id = atoi(szTmp);

//          if ( pNewOrderData->newin.c_id > 3000 || pNewOrderData->newin.c_id <1)
//                  return ERR_NEWORDER_CUSTOMER_INVALID;
    /*pNewOrderData->o_all_local=1;*/

//    bCheck = FALSE;
```


Copyright © 2004 NEC Corporation

```
            skipped = 0;
    for(i=0, items=0; i<15; i++)
    {

        if ( !GetKeyValue(lpszQueryString, szIID[i], szTmp, sizeof(szTmp)) )
                        return ERR_NEWORDER_MISSING_IID_KEY;

                if ( szTmp[0] )
                {
                        //if blank lines between item ids
//                      if ( bCheck )
//                              return
ERR_NEWORDER_ITEM_BLANK_LINES;
                        if ( !IsNumeric(szTmp) )
                                return ERR_NEWORDER_ITEMID_INVALID;
                        pNewOrderData->newin.ol_i_id[i-skipped] = atoi(szTmp);

                        if ( !GetKeyValue(lpszQueryString, szSP[i], szTmp,
sizeof(szTmp)) )
                                return
ERR_NEWORDER_MISSING_SUPPW_KEY;
                        if ( !IsNumeric(szTmp) )
                                return ERR_NEWORDER_SUPPW_INVALID;
                        pNewOrderData->newin.ol_supply_w_id[i-skipped] =
(int)atoi(szTmp); //2003/07/19
/*              if ( pNewOrderData->o_all_local && pNewOrderData-
>o_ol[i].ol_supply_w_id != pNewOrderData->w_id )
                          pNewOrderData->o_all_local = 0; */


                        if ( !GetKeyValue(lpszQueryString, szQty[i], szTmp,
sizeof(szTmp)) )
                                return
ERR_NEWORDER_MISSING_QTY_KEY;

                        if ( !IsNumeric(szTmp) )
                                return ERR_NEWORDER_QTY_INVALID;

                        pNewOrderData->old_quantity[i-skipped] = atoi(szTmp);
                        pNewOrderData->newin.ol_quantity[i-skipped] =
atoi(szTmp);

                        items++;

                        if ( pNewOrderData->newin.ol_i_id[i-skipped] >= 1000000
|| pNewOrderData->newin.ol_i_id[i-skipped] < 1 )
                                return ERR_NEWORDER_ITEMID_RANGE;
                        if ( pNewOrderData->newin.ol_quantity[i-skipped] >= 100
|| pNewOrderData->newin.ol_quantity[i-skipped] < 1 )
                                return ERR_NEWORDER_QTY_RANGE;


                        /*for (j = i-1-skipped; j >= 0;  j-- ) //bug?
                          if (pNewOrderData->newin.ol_i_id[i-skipped] ==
pNewOrderData->newin.ol_i_id[j]){
                                  pNewOrderData->newin.ol_quantity[i-skipped]
+= pNewOrderData->newin.ol_quantity[j];
                                  j = -1;
                          }
                        */
                //      skipped = 0;
                }
                else
                {
                        if ( !GetKeyValue(lpszQueryString, szSP[i], szTmp,
sizeof(szTmp)) )
                                return
ERR_NEWORDER_MISSING_QTY_KEY;

                        if ( szTmp[0] )
                                return
ERR_NEWORDER_SUPPW_WITHOUT_ITEMID;

                        if ( !GetKeyValue(lpszQueryString, szQty[i], szTmp,
sizeof(szTmp)) )
                                return
ERR_NEWORDER_MISSING_QTY_KEY;

                        if ( szTmp[0] )
                                return
ERR_NEWORDER_QTY_WITHOUT_ITEMID;

//                      bCheck = TRUE;
                        skipped ++;
                }
        }
          skipped = 0;
        if ( items == 0 )
```


Copyright © 2004 NEC Corporation

```
                return ERR_NEWORDER_NOITEMS_ENTERED;

    // pNewOrderData->newout.o_ol_cnt = items;

        return ERR_SUCCESS;
}

/* FUNCTION: int GetPaymentData(LPSTR lpszQueryString, PaymentData
*pPaymentData)
 *
 *
 * PURPOSE: This function extracts and validates the payment form data from an
 *          http command string.
 *
 * ARGUMENTS: LPSTR     lpszQueryString client browser http command string
 *            PaymentData       *pPaymentData   ptr to payment data structure
 *
 * RETURNS:   int              error code indicating reason for failure
 *            ERR_SUCCESS       all input data successfully parsed
 *
 * COMMENTS:    None
 *
 */
static int GetPaymentData(LPSTR lpszQueryString, PaymentData *pPaymentData)
{
        char    szTmp[26];
        char    *ptr;
//      char    *s;
        double fTmp;


//      s=lpszQueryString;
        if(!GetKeyValue(lpszQueryString, "DID*", szTmp, sizeof(szTmp)))
                return ERR_PAYMENT_MISSING_DID_KEY;

        if ( !IsNumeric(szTmp) )
                return ERR_PAYMENT_DISTRICT_INVALID;
        pPaymentData->payin.d_id = atoi(szTmp);

//      if ( pPaymentData->payin.d_id > 10 || pPaymentData->payin.d_id <1 )
//              return ERR_PAYMENT_DISTRICT_INVALID;

        if(!GetKeyValue(lpszQueryString, "CID*", szTmp, sizeof(szTmp)))
                return ERR_PAYMENT_MISSING_CID_KEY;

        if ( szTmp[0] && !IsNumeric(szTmp) )
                return ERR_PAYMENT_CUSTOMER_INVALID;

        pPaymentData->payin.c_id = atoi(szTmp);

//      if ( pPaymentData->payin.c_id > 3000 || pPaymentData->payin.c_id <1 )
//              return ERR_PAYMENT_CUSTOMER_INVALID;

        if (bLog){
                        FILE *fp1;
                        fp1 = fopen(szTpccLogPath, "ab");
                        fprintf(fp1, "cid = %s\n",szTmp);
                        fclose(fp1);
        }

        if ( szTmp[0] == 0 )
        {
                if(!GetKeyValue(lpszQueryString, "CLT*", szTmp, sizeof(szTmp)))
                        return ERR_PAYMENT_MISSING_CLT;
                _strupr( szTmp );

                strcpy(pPaymentData->payin.c_last, szTmp);
                if ( strlen(pPaymentData->payin.c_last) > 16 )
                        return ERR_PAYMENT_LAST_NAME_TO_LONG;
                pPaymentData->payin.bylastname=1;
        }
        else
        {
                if(!GetKeyValue(lpszQueryString, "CLT*", szTmp, sizeof(szTmp)))
                        return ERR_PAYMENT_MISSING_CLT_KEY;
                if ( szTmp[0] )
                        return ERR_PAYMENT_CID_AND_CLT;
        }

//      s=lpszQueryString;

        if(!GetKeyValue(lpszQueryString, "CWI*", szTmp, sizeof(szTmp)))
                return ERR_PAYMENT_MISSING_CWI_KEY;

        if ( !IsNumeric(szTmp) )
                return ERR_PAYMENT_CWI_INVALID;

        pPaymentData->payin.c_w_id = atoi(szTmp);
```


Copyright © 2004 NEC Corporation

```
//	if ( pPaymentData->payin.c_w_id > iMaxWareHouses || pPaymentData-
>payin.c_w_id <1)
//		return ERR_PAYMENT_CWI_INVALID;

	if (bLog){
				FILE *fp1;
				fp1 = fopen(szTpccLogPath, "ab");
				fprintf(fp1, "cid = %s\n",szTmp);
				fclose(fp1);
	}

	if(!GetKeyValue(lpszQueryString, "CDI*", szTmp, sizeof(szTmp)))
		return ERR_PAYMENT_MISSING_CDI_KEY;

	if ( !IsNumeric(szTmp) )
		return ERR_PAYMENT_CDI_INVALID;

	pPaymentData->payin.c_d_id = atoi(szTmp);

//	if ( pPaymentData->payin.c_d_id > 10 || pPaymentData->payin.c_d_id <1)
//		return ERR_PAYMENT_CDI_INVALID;

	if(!GetKeyValue(lpszQueryString, "HAM*", szTmp, sizeof(szTmp)))
		return ERR_PAYMENT_MISSING_HAM_KEY;

	ptr = szTmp;

	while( *ptr )
	{
		if ( *ptr == '.' )
		{
			ptr++;
			if ( !*ptr )
				break;
			if ( *ptr < '0' || *ptr > '9' )
				return ERR_PAYMENT_HAM_INVALID;
			ptr++;
			if ( !*ptr )
				break;
			if ( *ptr < '0' || *ptr > '9' )
				return ERR_PAYMENT_HAM_INVALID;
			if ( !*ptr )
				return ERR_PAYMENT_HAM_INVALID;
		}
		else if ( *ptr < '0' || *ptr > '9' )
				return ERR_PAYMENT_HAM_INVALID;
		ptr++;
	}

	fTmp = atof(szTmp);
	pPaymentData->payin.h_amount = fTmp * 100;
	if ( pPaymentData->payin.h_amount >= 1000000 || pPaymentData-
>payin.h_amount < 0 )
		return ERR_PAYMENT_HAM_RANGE;

	if (bLog){
				FILE *fp1;
				fp1 = fopen(szTpccLogPath, "ab");
				fprintf(fp1, "cid = %s\n",szTmp);
				fclose(fp1);
	}
	if (bLog){
				FILE *fp1;
				fp1 = fopen(szTpccLogPath, "ab");
				fprintf(fp1, "ouma = %s\n",lpszQueryString);
				fclose(fp1);
	}

	return ERR_SUCCESS;
}


/* FUNCTION: int GetOrderStatusData(LPSTR lpszQueryString,
 *		OrderStatusData *pOrderStatusData)
 *
 * PURPOSE: This function extracts and validates the payment form data from an
 *	http command string.
 *
 * ARGUMENTS:LPSTR		lpszQueryString	client browser http command
 *		string
 *	OrderStatusData *pOrderStatusData	pointer to order status
 *		data structure
 *
 * RETURNS:	int		error code indicating reason for failure
 *		ERR_SUCCESS	successfully parsed all required input data
 *
 * COMMENTS:	None
```


Copyright © 2004 NEC Corporation

```
 *
 */
static int GetOrderStatusData(LPSTR lpszQueryString, OrderStatusData
*pOrderStatusData)
{
        char    szTmp[26];
//      char    *s;
//      s=lpszQueryString;

        if(!GetKeyValue(lpszQueryString, "DID*", szTmp, sizeof(szTmp)))
                return ERR_ORDERSTATUS_MISSING_DID_KEY;
        if ( !IsNumeric(szTmp) )
                return ERR_ORDERSTATUS_DID_INVALID;
        pOrderStatusData->ordin.d_id = atoi(szTmp);

        if(!GetKeyValue(lpszQueryString, "CID*", szTmp, sizeof(szTmp)))
                return ERR_ORDERSTATUS_MISSING_CID_KEY;

        if ( szTmp[0] == 0 )
        {
                pOrderStatusData->ordin.c_id = 0;
                if(!GetKeyValue(lpszQueryString, "CLT*", szTmp, sizeof(szTmp)))
                        return ERR_ORDERSTATUS_MISSING_CLT_KEY;
                _strupr( szTmp );
                strcpy(pOrderStatusData->ordin.c_last, szTmp);
                if ( strlen(pOrderStatusData->ordin.c_last) > 16 )
                        return ERR_ORDERSTATUS_CLT_RANGE;
                pOrderStatusData->ordin.bylastname = 1;
        }
        else
        {
                if ( !IsNumeric(szTmp) )
                        return ERR_ORDERSTATUS_CID_INVALID;
                pOrderStatusData->ordin.c_id = atoi(szTmp);
                if(!GetKeyValue(lpszQueryString, "CLT*", szTmp, sizeof(szTmp)))
                        return ERR_ORDERSTATUS_MISSING_CLT_KEY;
                if ( szTmp[0] )
                        return ERR_ORDERSTATUS_CID_AND_CLT;
        }

    return ERR_SUCCESS;
}
```

```
#ifdef WIN32
/* FUNCTION: BOOL ReadRegistrySettings(void)
 *
 * PURPOSE: This function reads the NT registry for startup parameters. There
 *          parameters are under the TPCC key.
 *
 * ARGUMENTS:   None
 *
 * RETURNS:     None
 *
 * COMMENTS:    This function also sets up required operation variables to
 *              their default value so if registry is not setup the default
 *              values will be used.
 *
 */
static BOOL ReadRegistrySettings(void)
{
    HKEY      hKey;
    DWORD     size;
    DWORD     type;
    char      szTmp[256];

    bLog      = FALSE;
    dLog      = FALSE;
    bGeneric  = FALSE;
    iMaxWareHouses    = 625;
    iThreads          = 5;
    iDelayMs          = 100;
    iDeadlockRetry    = (short)3;
    strcpy(szTpccLogPath, "tpcclog.");

    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\\ORACLE\\tpcc", 0, KEY_READ,&hKey) != ERROR_SUCCESS )
            return TRUE;
    size = sizeof(szTmp);

    if ( RegQueryValueEx(hKey, "PATH", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
    {
        strcpy(szTpccLogPath, szTmp);
        strcat(szTpccLogPath, "tpcclog.");
        strcpy(szErrorLogPath, szTmp);
        strcat(szErrorLogPath, "tpccerr.");
    }
```


Copyright © 2004 NEC Corporation

```
    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "LOG", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
    {
        if ( !stricmp(szTmp, "ON") )
        bLog = TRUE;
    }

    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "DEBUG", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
    {
        if ( !stricmp(szTmp, "ON") )
        dLog = TRUE;
    }

    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "MaximumWarehouses", 0, &type, szTmp, &size)
== ERROR_SUCCESS )
    {
        iMaxWareHouses = atoi(szTmp);
        if ( iMaxWareHouses == 0 )
            iMaxWareHouses = 500;
    }

    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "NumberOfDeliveryThreads", 0, &type, szTmp,
&size) == ERROR_SUCCESS )
        iThreads = atoi(szTmp);
    if ( !iThreads )
        iThreads = 5;

    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "BackoffDelay", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
        iDelayMs = atoi(szTmp);
    if ( !iDelayMs )
        iDelayMs = 100;

    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "DeadlockRetry", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
        iDeadlockRetry = (short)atoi(szTmp);
    if ( !iDeadlockRetry )
        iDeadlockRetry = (short)3;

    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "MaxConnections", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
        iMaxConnections = (short)atoi(szTmp);
    if ( !iMaxConnections )
        iMaxConnections = (short)25;

    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "GenSrvc", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
    {
        if ( !stricmp(szTmp, "ON") )
        bGeneric = TRUE;
    }

    RegCloseKey(hKey);
    return FALSE;
}
#endif

/* FUNCTION: BOOL IsNumeric(char *ptr)
 *
 * PURPOSE: This function determines if a string is numeric. It fails if any
 *          characters other than numeric and null terminator are present.
 *
 * ARGUMENTS:   char    *ptr    pointer to string to check.
 *
 * RETURNS:     BOOL    FALSE   if string is not all numeric
 *              TRUE    if string contains all numeric characters i.e. '0' - '9'
 *
 * COMMENTS:    None
 *
 */

static BOOL IsNumeric( char *ptr)
{
    if ( *ptr == 0 )
        return FALSE;
    while( *ptr && isdigit(*ptr) )
        ptr++;
    return ( !*ptr );
```


Copyright © 2004 NEC Corporation

```
}

/* FUNCTION: void FormatHTMLString(char *szBuff, int iLen, char *szStr)
 *
 * PURPOSE:  This function Handles translation of HTML specific character field
 *          data when an HTML output form is generated.
 *
 * ARGUMENTS:   char    *szBuff        Returned string information
 *              char    *szStr         input string to be formatted.
 *              int     iLen           Length of returned string
 *
 * RETURNS:     none
 *
 * COMMENTS:    The length parameter is the absolute length of the returned
 *              string in HTML characters. For example the input string > would
 *              be returned as &gt; which would be counted as 1 character.If
 *              the number of input characters is less than the iLen parameter
 *              spaces are appended to the end of the string to ensure that at
 *              least iLen characters are returned in the szBuff parameter.
 *
 */

static void FormatHTMLString(char *szBuff, char *szStr, int iLen)
{
    while( iLen && *szStr )
    {
        switch( *szStr )
        {
          case '>':
                *szBuff++ = '&';
                *szBuff++ = 'g';
                *szBuff++ = 't';
                *szBuff++ = ';';
                szStr++;
                break;
          case '<':
                *szBuff++ = '&';
                *szBuff++ = 'l';
                *szBuff++ = 't';
                *szBuff++ = ';';
                szStr++;
                break;
          case '&':
                *szBuff++ = '&';
                *szBuff++ = 'a';
                *szBuff++ = 'm';
                *szBuff++ = 'p';
                *szBuff++ = ';';
                szStr++;
                break;
          case '\"':
                *szBuff++ = '&';
                *szBuff++ = 'q';
                *szBuff++ = 'u';
                *szBuff++ = 'o';
                *szBuff++ = 't';
                *szBuff++ = ';';
                szStr++;
                break;
          default:
                *szBuff++ = *szStr++;
                break;
        }
        iLen--;
    }
    while( iLen-- )
    *szBuff++ = ' ';
    *szBuff = 0;
    return;
}

/*static int ThrTpInit(EXTENSION_CONTROL_BLOCK *pECB)
{
    // use CS for tptpalloc ?
    static int num_tpinits=0;
    static int x=1;
    static int once=0;
    static CRITICAL_SECTION     TpCriticalSection;
        int iRc, TpRc, lasterr;
        char buf[40];
        int retry = 0;

        BOOL Success = FALSE;

    if(!TlsGetValue(TLSIsTpInitedKey))
    {
                lasterr = GetLastError();
                if (lasterr != NO_ERROR)
```


Copyright © 2004 NEC Corporation

```
		{
			sprintf(buf,"TlsGetValue %d",lasterr);
			ErrorMessage(pECB, ERR_TPINIT_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
			return -1;
		}
		if (!once)
		{
			InitializeCriticalSection(&TpCriticalSection);
								once=1;
		}


		if ( dLog )
		{
			FILE *fp;

			fp = fopen(szTpccLogPath, "ab");
			fprintf(fp, "* In ThrTpInit Thread %d * \r\n",
GetCurrentThreadId());
			fclose(fp);
		}

		while ( retry < TP_MAX_RETRIES )
		{
			EnterCriticalSection(&TpCriticalSection);

			if(tpinf == NULL)
			{
			  if ((tpinf = ( TPINIT *)tpalloc("TPINIT", NULL,
sizeof(TPINIT))) == NULL)
			  {
				TpRc = tperrno;
				if ( dLog )
				{
					FILE *fp;

					fp = fopen(szTpccLogPath, "ab");
					fprintf(fp, ">>>> ThrTpInit %d : tpalloc
of tpinit failed: %d \r\n",
						GetCurrentThreadId(), TpRc);
					fclose(fp);
```

```
Retry : count %d * \r\n",
				}


				tpinf = NULL;
				retry++;
				LeaveCriticalSection(&TpCriticalSection);
				Sleep(50); // Relinquish thread timeslice
				continue;
			  }

			  tpinf->flags |= TPMULTICONTEXTS;
		}


		// Do the TPINIT
		itoa(++num_tpinits, tpinf->cltname, 10);
		iRc = tpinit(tpinf);

		// check tpinit() ?
		if (iRc < 0 )
		{
			TpRc = tperrno;

			lasterr = GetLastError();
			retry++ ;

			if ( dLog )
			{
				FILE *fp;

				fp = fopen(szTpccLogPath, "ab");
				fprintf(fp, "* ThrTpInit Thread %d

					GetCurrentThreadId(), retry);
				fclose(fp);
			}
		}
		else
		{
			Success = TRUE;
			tpfree( ( char * ) tpinf);
			tpinf=NULL;
			break;
		}
```


Copyright © 2004 NEC Corporation

```
--num_tpinits;
LeaveCriticalSection(&TpCriticalSection);
Sleep(50);          // Relinquish thread timeslice

} // retry the tpinit if it failed the first time

LeaveCriticalSection(&TpCriticalSection);

if ( Success == FALSE )
{
TpRc = tperrno;
{
FILE *fp;

fp = fopen(szTpccLogPath, "ab");
fprintf(fp, ">>>> ThrTpInit Thread %d Failed
tperrno= %d Last Err= %d \r\n",
GetCurrentThreadId(), TpRc, lasterr);
fclose(fp);
}
sprintf(buf,"Thread %d: In ThrTpInit iRc=%d
Tperrno=%d",
GetCurrentThreadId(),iRc, TpRc);
ErrorMessage(pECB, ERR_TPINIT_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);

return -1;
}
if ( Success == TRUE )
{
if ( dLog )
{
FILE *fp;

fp = fopen(szTpccLogPath, "ab");
fprintf(fp, "* ThrTpInit Thread %d Success retry
count %d * \r\n",
GetCurrentThreadId(), retry);
fclose(fp);
}

if ( ( iRc=TlsSetValue(TLSIsTpInitedKey,&x)) == 0)
{
{
FILE *fp;

fp = fopen(szTpccLogPath, "ab");
fprintf(fp, ">>>> ThrTpInit %d :
TlsSetValue Failed iRc: %d \r\n",
GetCurrentThreadId(), iRc);
fclose(fp);
}
sprintf(buf,"TlsSetValue %d",iRc);
ErrorMessage(pECB, ERR_TPINIT_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
return -1;
}
if (dLog)
{
FILE *fp;

fp = fopen(szTpccLogPath, "ab");
fprintf(fp, "*ThrTpInit %d : TlsSetValue
Failed iRc: v=%d \r\n",
GetCurrentThreadId(),
TlsGetValue(TLSIsTpInitedKey));
fclose(fp);
}
}
}
else
{
if ( dLog )
{
FILE *fp;

fp = fopen(szTpccLogPath, "ab");
fprintf(fp, "* ThrTpInit Thread %d already tpinited * \r\n",
GetCurrentThreadId());
fclose(fp);
}
}

return 0;
}*/


static int ThrTpInit(EXTENSION_CONTROL_BLOCK *pECB)
```


Copyright © 2004 NEC Corporation

```
{
    // use CS for tptpalloc ?
    int ret=0;
            static int num_tpinits=0;
    static int x=1;
    static int once=0;
    static apr_os_thread_t tid;
    unsigned long TLSIsTpInit;
    //static CRITICAL_SECTION     TpCriticalSection;
    static apr_thread_mutex_t*     TpCriticalSection=NULL;
            int iRc, TpRc, lasterr;
            char buf[40];
            int retry = 0;

           BOOL Success = FALSE;
           apr_pool_t* p;
           apr_thread_mutex_lock(loaded.lock);

           p = pECB->r->server->process->pool;
           // create critical section for this process
           if (!TpCriticalSection)

           apr_thread_mutex_create(&TpCriticalSection,APR_THREAD_MUTEX_DE
FAULT,p);
           {

                       FILE *fp;
                       tls_t key;
                       key.pid = getpid();
                       key.tid=apr_os_thread_current();
                       TLSIsTpInit = (unsigned long) apr_hash_get(loaded.hash, &key,
sizeof(key));
                       if (dLog)
                       {
                       fp = fopen(szTpccLogPath, "ab");
                       fprintf(fp, "TlsGet pid=%d,tid= %d, count=%d,value=%ld \r\n",
                       getpid(),GetCurrentThreadId(),
apr_hash_count(loaded.hash),TLSIsTpInit);
                       fclose(fp);
                       }
           }
           apr_thread_mutex_unlock(loaded.lock);

    //if(!TlsGetValue(TLSIsTpInitedKey))
    if(!TLSIsTpInit)
    {
                       lasterr = GetLastError();
                       if (lasterr != NO_ERROR)
                       {
                                  sprintf(buf,"TlsGetValue %d", lasterr);
                                  ErrorMessage(pECB, ERR_TPINIT_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
                                  return -1;
                       }
                       if (!once)
                       {
                                  once=1;
                                  if (dLog)
                       {
                                  FILE *fp;

                                  fp = fopen(szTpccLogPath, "ab");
                                  fprintf(fp, "* In ThrTpInit create cs pid=%d, tid= %d *
\r\n",getpid(), GetCurrentThreadId());
                                  fclose(fp);
                       }
                                  // this is done in module_init
                                  //mcInitializeCriticalSection(&(TpCriticalSection));

                       }

                       if ( dLog )
                       {
                                  FILE *fp;

                                  fp = fopen(szTpccLogPath, "ab");
                                  fprintf(fp, "* In ThrTpInit Thread %d * \r\n",
GetCurrentThreadId());
                                  fclose(fp);
                       }

                       for ( retry = 0; retry < TP_MAX_RETRIES; retry++ )
                       {
                                  mcEnterCriticalSection(TpCriticalSection);


                                  if(tpinf == NULL)
                                  {
```


Copyright © 2004 NEC Corporation

```
                    if ((tpinf = ( TPINIT *)tpalloc("TPINIT", NULL,
sizeof(TPINIT))) == NULL)
                    {
                        TpRc = tperrno;
                        lasterr = GetLastError();

                        if ( dLog )
                        {
                            FILE *fp;

                            fp = fopen(szTpccLogPath, "ab");
                            fprintf(fp, ">>>> ThrTpInit %d : tpalloc
of tpinit failed: %d \r\n",
                                GetCurrentThreadId(), TpRc);
                            fclose(fp);


                        }


                        tpinf=NULL;
                        Sleep(50); // Relinquish thread timeslice
                        mcLeaveCriticalSection(TpCriticalSection);
                        continue;
                    }
                }

                // Do the TPINIT
        tpinf->flags |= TPMULTICONTEXTS;
                //itoa(++num_tpinits, tpinf->cltname, 10);
                sprintf(tpinf->cltname,"%d",++num_tpinits);
                iRc = tpinit(tpinf);
                if (iRc < 0 )
                {
                    TpRc = tperrno;
                    lasterr = GetLastError();

                    if ( dLog )
                    {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* ThrTpInit Thread %d
Retry : count %d * \r\n",
                                GetCurrentThreadId(), retry);
                        fclose(fp);
                    }

                    --num_tpinits;
                    Sleep(50);          // Relinquish thread timeslice
                    mcLeaveCriticalSection(TpCriticalSection);
                    continue;
                }
                if (dLog)
                    {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* ThrTpInit
cltname=%s,pid=%d,Thread %d Retry : count %d * \r\n",tpinf->cltname,getpid(),
                                GetCurrentThreadId(), retry);
                        fclose(fp);
}

                Success = TRUE;
                tpfree( ( char * ) tpinf);
                tpinf=NULL;
                Sleep(50);          // Relinquish thread timeslice
                mcLeaveCriticalSection(TpCriticalSection);
                break;

            } // retry the tpinit if it failed the first time

            if ( Success == FALSE )
            {
                if ( dLog )
                {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, ">>>> ThrTpInit Thread %d Failed
tperrno= %d Last Err= %d, oser=%d \r\n",
                        GetCurrentThreadId(), TpRc,
lasterr,Uunixerr);
                    fclose(fp);
                }
                sprintf(buf,"Thread %d: In ThrTpInit iRc=%d
Tperrno=%d",
```


Copyright © 2004 NEC Corporation

```
                                GetCurrentThreadId(),iRc, TpRc);
                        ErrorMessage(pECB, ERR_TPINIT_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);

                //              return -1;
                        ret = -1;
                }
                else
                if ( Success == TRUE )
                {
                        if ( dLog )
                        {
                                FILE *fp;

                                fp = fopen(szTpccLogPath, "ab");
                                fprintf(fp, "* ThrTpInit Thread %d Success retry
count %d * \r\n",
                                        GetCurrentThreadId(), retry);
                                fclose(fp);
                        }

        //              if ( ( iRc=TlsSetValue(pECB->r->server,10)) == 0)

        apr_thread_mutex_lock(loaded.lock);
        {
                FILE *fp;
                unsigned long  v=10;
                tls_t* key;
                key=apr_pcalloc(loaded.pool, sizeof (tls_t));
                key->pid=getpid();
                key->tid=apr_os_thread_current();
                apr_hash_set(loaded.hash,key, sizeof(tls_t),(void*)v);
                v =(unsigned long)apr_hash_get(loaded.hash,key, sizeof(tls_t));
                if (dLog)
                {
                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "TlsSet pid=%d,tid= %d, count=%d,value=%ld
\r\n",
                        getpid(),GetCurrentThreadId(),
apr_hash_count(loaded.hash),v);
                        fclose(fp);
                }
        }
        apr_thread_mutex_unlock(loaded.lock);
                        if (0)
                        {
                                {
                                        FILE *fp;

                                        fp = fopen(szTpccLogPath, "ab");
                                        fprintf(fp, ">>>> ThrTpInit %d :
TlsSetValue Failed iRc: %d \r\n",
                                        GetCurrentThreadId(), iRc);
                                        fclose(fp);
                                }
                                sprintf(buf,"TlsSetValue %d",iRc);
                                ErrorMessage(pECB, ERR_TPINIT_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
                                ret= -1;
                        }
                }
        }
        else
        {
                if ( dLog )
                {
                        FILE *fp;

                        fp = fopen(szTpccLogPath, "ab");
                        fprintf(fp, "* ThrTpInit Thread %d already tpinited * \r\n",
GetCurrentThreadId());
                        fclose(fp);
                }
        }
    return ret;
}


#ifdef TOPEND

void space_fill(s1, s2, i)

char *s1, *s2;
int i;

{
    int ii;
```


Copyright © 2004 NEC Corporation

```
    strncpy(s1,s2,i);

    for(ii=0; ii<i; ii++)
    {
        if (s1[ii] == '\0') s1[ii] = 0x20;
    }
    return;
}

/* This is the function for the new thread that is added to the process. */
/* The error handling needs to be improved.  Almost every problem is fatal. */
static void TEReceiveThread(void *ptr)
{
    tp_dif_structs_t *dif_struct;
    long buffer_length;
    char *message_buffer;
    int rc;
    FILE *fp1;

   if ((rc = (int)tp_ChangeToGroup ((LPSTR)getenv("TP_SYSTEM"),
        LOGON32_LOGON_INTERACTIVE,
        LOGON32_PROVIDER_DEFAULT)) != 0)
    {
        if (bLog)
        {
          fp1 = fopen(szTpccLogPath, "ab");
          fprintf(fp1, "Received bad status from tp_ChangeToGroup, status = %d
***\n", rc);
          fclose(fp1);
        }
                MT_LOG_ERROR;
        return;
    }

    dif_struct = tp_csi_alloc (TP_DIF_DIAL_INFO |
                               TP_DIF_SERVICE_NAME |
                               TP_DIF_OUTPUT_FORMAT |
                               TP_DIF_LOCATION);

    if (dif_struct == NULL)
    {
        MT_LOG_ERROR;
        return;
    }

    if ((message_buffer = (char *) malloc (TP_MAX_BUF_LEN)) == NULL)
    {
        MT_LOG_ERROR;
        return;
    }

    /* receive loop.  Should there be a way to shut this down? */
    for (;;)
    {
        if (tp_WaitForSingleObject (NULL, INFINITE) == WAIT_TIMEOUT)
        {
          MT_LOG_ERROR;
          return;
        }
        dif_struct->info->tp_user_dialogue_id = TP_ANY_DIALOGUE;
        dif_struct->info->tp_flags = TP_NOFLAGS;
        buffer_length = TP_MAX_BUF_LEN;

        rc = tp_client_receive (dif_struct->info,TP_NOBLOCK,
                                dif_struct->output_format,
                                dif_struct->service,
                                dif_struct->location,
                                &buffer_length,
                                message_buffer);
        switch (rc)
        {
          case TP_TIMEOUT:
          {
                break;
          }
          case TP_ADMIN:
          {
        /* This assumes that admin messages aren't used in the application. */
        /* Receive the message and throw it away. */
                long flags, buf_len;
                char buffer[1];

                flags = TP_TRUNCATE;
                buf_len = 1;
                tp_system_admin (&flags, &buf_len, buffer);
                break;
          }
          case TP_DISSOLVED:
```


Copyright © 2004 NEC Corporation

```
        case TP_OK:
        case TP_RESET:
        case TP_SERVICE:
        case TP_SIGNON_INHIBITED:
        case TP_USER:
        {
                DialogTableEntry_t *DTE;
                void *ptr;

                DTE = FindOnList (dif_struct->info->tp_user_dialogue_id,
&DTActiveList);
                if (DTE == NULL)
                {
                /* We received in a dialog without outstanding work???? */
                  MT_LOG_ERROR;
                  return;
                }

                /* trade data with DTE. */
                ptr = DTE->dif_struct->info;
                DTE->dif_struct->info = dif_struct->info;
                dif_struct->info = ptr;

                ptr = DTE->dif_struct->output_format;
                DTE->dif_struct->output_format = dif_struct->output_format;
                dif_struct->output_format = ptr;

                ptr = DTE->dif_struct->service;
                DTE->dif_struct->service = dif_struct->service;
                dif_struct->service = ptr;

                ptr = DTE->dif_struct->location;
                DTE->dif_struct->location = dif_struct->location;
                dif_struct->location = ptr;

                DTE->buffer_length = buffer_length;

                if (DTE->message_buffer == NULL)
                {
                  if ((DTE->message_buffer = malloc (buffer_length)) == NULL)
                  {
                        MT_LOG_ERROR;
                        return;
                  }
                }
                else
                {
                  /* NONPORTABLE */
                  if (_msize (DTE->message_buffer) < (size_t) buffer_length)
                  {
                        if ((DTE->message_buffer = realloc (DTE-
>message_buffer,
                            buffer_length)) == NULL)
                        {
                          MT_LOG_ERROR;
                          return;
                        }
                  }
                }
                memcpy (DTE->message_buffer, message_buffer, buffer_length);
                DTE->rc = rc;
                /* Signal that the data is ready. */
                SetEvent (DTE->hSync);
                break;
        }
        case TP_SHUTDOWN:
        {
                DialogTableEntry_t *DTE;
                /* tell everyone */
                if (WaitForSingleObject (DTActiveList.hSync, INFINITE)  ==
WAIT_TIMEOUT)
                {
                  MT_LOG_ERROR;
                  return;
                }
                DTE = DTActiveList.next;
                while (DTE != NULL)
                {
                  DTE->rc = TP_SHUTDOWN;
                  SetEvent (DTE->hSync);
                  DTE = DTE->next;
                }
                if (!ReleaseMutex(DTActiveList.hSync))
                {
                  MT_LOG_ERROR;
                  return;
                }
                break;
```


Copyright © 2004 NEC Corporation

```
            }
            default:
            {
                    if (bLog)
                    {
                      fp1 = fopen(szTpccLogPath, "ab");
                      fprintf(fp1, "TEReceiveThread received error status = %d\n", rc);
                      fclose(fp1);
                    }
                    MT_LOG_ERROR;
                    return;
            }
          }
    }
}



int tp_mt_initialize(tp_application_info_t *application_info,
                         tp_funct_struct_t     function_array[],
                         long                  number_functions)
{
    static HANDLE hReceiveThread = NULL;
    int rc;

            rc = tp_initialize(application_info, function_array, number_functions);

    if (rc == TP_OK)
    {
    /* Make sure that the global data is initialized. */
          if (DTFreeList.hSync == NULL)
          {
            if ((DTFreeList.hSync = CreateMutex(NULL, FALSE, NULL)) == NULL)
            {
                    MT_LOG_ERROR;
                    tp_terminate();
                    return TP_DIFERR;
            }
          }
          if (DTActiveList.hSync == NULL)
          {
            if ((DTActiveList.hSync = CreateMutex(NULL, FALSE, NULL)) ==
NULL)
            {
                    MT_LOG_ERROR;
                    tp_terminate();
                    return TP_DIFERR;
            }
          }
          /* Make sure that the receive thread is running */
          if (hReceiveThread == NULL)
          {
            if ((hReceiveThread = (HANDLE) _beginthread(TEReceiveThread, 0,
NULL)) == (HANDLE) -1 )
            {
                    MT_LOG_ERROR;
                    tp_terminate();
                    return TP_DIFERR;
            }
            if (!SetThreadPriority (hReceiveThread,
RECEIVE_THREAD_PRIORITY))
            {
                    MT_LOG_ERROR;
            }
          }
    }
    return rc;
}


int tp_mt_client_signon(tp_dialogue_info_t *info,
                            tp_dialogue_user_t *client,
                            long            inactivity_time,
                            tp_service_name_t  *service,
                            tp_input_format_t  *input_format,
                            long            message_length,
                            char            *message_text)
{
    DialogTableEntry_t *DTE = GetDTE();
    int rc;

    if (DTE == NULL)
    {
          MT_LOG_ERROR;
          return TP_DIFERR;
    }
    if (!ResetEvent (DTE->hSync))
    {
```


Copyright © 2004 NEC Corporation

```
        MT_LOG_ERROR;
        return TP_DIFERR;
    }

    /* This is to handle TP_UNIQUE_DIALOGUE */
    DTE->usr_dlg_ptr = &(info->tp_user_dialogue_id);
    AddToList(DTE, &DTActiveList);

    rc = tp_client_signon(info, client, inactivity_time, service,
                                input_format, message_length, message_text);

    switch (rc)
    {
        case TP_DISSOLVING:
        case TP_OK:
        {
          DTE->usr_dlg_storage = info->tp_user_dialogue_id;
          DTE->usr_dlg_ptr = &(DTE->usr_dlg_storage);
          break;
        }
        default:
        {
        /* there is nothing to receive */
          RemoveFromList (DTE, &DTActiveList);
          AddToList (DTE, &DTFreeList);
        }
    }
    return rc;
}


int tp_mt_client_send(tp_dialogue_info_t *info,
                      tp_service_name_t  *service,
                      tp_input_format_t  *input_format,
                      long               message_length,
                      char               *message_text)
{
    DialogTableEntry_t *DTE = GetDTE();
    int rc;

    if (DTE == NULL)
    {
        MT_LOG_ERROR;
        return TP_DIFERR;
    }
    if (!ResetEvent (DTE->hSync))
    {
        MT_LOG_ERROR;
        return TP_DIFERR;
    }
    DTE->usr_dlg_storage = info->tp_user_dialogue_id;
    DTE->usr_dlg_ptr = &(DTE->usr_dlg_storage);
    AddToList(DTE, &DTActiveList);

    rc = tp_client_send(info, service, input_format, message_length, message_text);
    switch (rc)
    {
        case TP_DISSOLVING:
        case TP_OK:
        case TP_RESET:
        {
          break;
        }
        default:
        {
          /* there is nothing to receive */
          RemoveFromList (DTE, &DTActiveList);
          AddToList (DTE, &DTFreeList);
        }
    }
    return rc;
}


int tp_mt_client_receive(tp_dialogue_info_t *info,
                         long               wait_time,
                         tp_output_format_t *output_format,
                         tp_service_name_t  *service,
                         tp_location_t      *location,
                         long               *buffer_length,
                         char               *message_buffer)
{
    DialogTableEntry_t *DTE = FindOnList(info->tp_user_dialogue_id,
&DTActiveList);
    int rc;
    FILE *fp1;
```


Copyright © 2004 NEC Corporation

```
    if (DTE == NULL)
    {
        MT_LOG_ERROR;
        return TP_DIFERR;
    }
    wait_time = TP_BLOCK;

    /* Wait for the receive to actually happen */
    if (WaitForSingleObject(DTE->hSync, INFINITE) == WAIT_TIMEOUT)
    {
        if (bLog)
        {
          fp1 = fopen(szTpccLogPath, "ab");
          fprintf(fp1, "tp_mt_client_receive got a timeout error\n");
          fclose(fp1);
        }
        return TP_TIMEOUT;
    }

    if (DTE->buffer_length > *buffer_length)
    {
        *buffer_length = DTE->buffer_length;
        return TP_BUFFSIZE;
    }

    RemoveFromList (DTE, &DTActiveList);

    memcpy (info, DTE->dif_struct->info, sizeof (tp_dialogue_info_t));
    memcpy (output_format, DTE->dif_struct->output_format, sizeof
(tp_output_format_t));
    memcpy (service, DTE->dif_struct->service, sizeof (tp_service_name_t));
    memcpy (location, DTE->dif_struct->location, sizeof (tp_location_t));
    *buffer_length = DTE->buffer_length;
    memcpy (message_buffer, DTE->message_buffer, *buffer_length);
    rc = DTE->rc;

    AddToList (DTE, &DTFreeList);
    return rc;
}


void AddToList (DialogTableEntry_t *DTE, DialogTableEntry_t *head)
{
    if (WaitForSingleObject (head->hSync, INFINITE) == WAIT_TIMEOUT)
    {
        MT_LOG_ERROR;
        return;
    }
    DTE->next = head->next;
    head->next = DTE;
    if (!ReleaseMutex (head->hSync))
    {
        MT_LOG_ERROR;
    }
    return;
}


void RemoveFromList (DialogTableEntry_t *DTE, DialogTableEntry_t *head)
{
    DialogTableEntry_t *ptr;

    if (WaitForSingleObject (head->hSync, INFINITE) == WAIT_TIMEOUT)
    {
        MT_LOG_ERROR;
        return;
    }
    ptr = head;

    while ((ptr != NULL) && (ptr->next != DTE))
    {
        ptr = ptr->next;
    }
    if (ptr == NULL)
    {
        MT_LOG_ERROR;
        ReleaseMutex (head->hSync);
        return;
    }
    ptr->next = DTE->next;

    if (!ReleaseMutex (head->hSync))
    {
    MT_LOG_ERROR;
    }
    return;
}
```


Copyright © 2004 NEC Corporation

```
DialogTableEntry_t *FindOnList (long tp_user_dialogue_id, DialogTableEntry_t
*head)
{
    DialogTableEntry_t *ptr;

    if (WaitForSingleObject (head->hSync, INFINITE) == WAIT_TIMEOUT)
    {
        MT_LOG_ERROR;
        return NULL;
    }
    ptr = head->next;

    while ((ptr != NULL) && (*(ptr->usr_dlg_ptr) != tp_user_dialogue_id))
    {
        ptr = ptr->next;
    }
    if (ptr == NULL)
    {
        MT_LOG_ERROR;
        ReleaseMutex (head->hSync);
        return NULL;
    }
    if (!ReleaseMutex (head->hSync))
    {
        MT_LOG_ERROR;
    }
    return ptr;
}

DialogTableEntry_t *GetDTE()
{
    DialogTableEntry_t *ptr = NULL;

    if (WaitForSingleObject (DTFreeList.hSync, INFINITE) == WAIT_TIMEOUT)
    {
        MT_LOG_ERROR;
        return NULL;
    }

    if (DTFreeList.next != NULL)
    {
        ptr = DTFreeList.next;
        DTFreeList.next = ptr->next;
    }

    if (!ReleaseMutex (DTFreeList.hSync))
    {
        MT_LOG_ERROR;
    }

    if (ptr == NULL)
    {
        if ((ptr = (DialogTableEntry_t *) malloc (sizeof (DialogTableEntry_t)))
==NULL)
        {
          MT_LOG_ERROR;
          return NULL;
        }

        if ((ptr->dif_struct = tp_csi_alloc (TP_DIF_DIAL_INFO |
                                             TP_DIF_SERVICE_NAME |
                                             TP_DIF_OUTPUT_FORMAT |
                                             TP_DIF_LOCATION)) == NULL)
        {
          MT_LOG_ERROR;
          free(ptr);
          return NULL;
        }
        ptr->buffer_length = 0;
        ptr->message_buffer = NULL;
        ptr->rc = 0;
        if ((ptr->hSync = CreateEvent (NULL, TRUE, FALSE,NULL)) == NULL)
        {
          MT_LOG_ERROR;
          tp_csi_free (ptr->dif_struct);
          free (ptr);
          return NULL;
        }
        ptr->next = NULL;
    }
    return ptr;
}

#endif
```


Copyright © 2004 NEC Corporation

tpccerr.h

```
#ifndef TPCCERR_H
#define TPCCERR_H

/*=================================================================+
 |         Copyright (c) 1997  Oracle Corp, Redwood Shores, CA      |
 |                        All Rights Reserved                       |
 +=================================================================+
 | FILENAME
 |    tpccerr.h
 | DESCRIPTION
 |    Include file for TPC-C benchmark programs.
 +=================================================================*/

//#define LIMIT_TUXQUEUE

#define TP_MAX_RETRIES  1
#define ERR_TYPE_WEBDLL        1
#define ERR_TYPE_SQL   2
#define ERR_TYPE_OCI   3
#define ERR_TYPE_TUXEDO 4
#define ERR_DB_SUCCESS 0
/* Database transaction succeeded */

#define ERR_DB_ERROR          1       /* A database error has occured */
#define ERR_TRANSPORT_ERROR   2       /* A transport error has occured */
#define ERR_DB_INTERFACE      3       /* An error occured setting up */
                                      /* connection to DB */
#define ERR_DB_DEADLOCK_LIMIT 4       /* The DB deadlock retry limit was
*/
                                      /* reached */
#define ERR_DB_NOT_COMMITTED  5       /* Transaction not committed */
#define ERR_DB_DEAD           6       /* Database connection is invalid
*/
#define ERR_SUCCESS           1000    //"Success, no error.
#define ERR_COMMAND_UNDEFINED 1001    //"Command undefined.
#define ERR_NOT_IMPLEMENTED_YET       1002   //"Not Implemented Yet.
#define ERR_CANNOT_INIT_TERMINAL  1003       //"Cannot initialize client
connection.
#define ERR_OUT_OF_MEMORY     1004    //"insufficient memory.
#define ERR_NEW_ORDER_NOT_PROCESSED 1005  //"Cannot process new Order
form.
#define ERR_PAYMENT_NOT_PROCESSED  1006      //"Cannot process payment
form.
#define ERR_NO_SERVER_SPECIFIED       1007  //"No Server name specified.
#define ERR_ORDER_STATUS_NOT_PROCESSED 1008 //"Cannot process order
status form.
#define ERR_W_ID_INVALID      1009    //"Invalid Warehouse ID.
#define ERR_CAN_NOT_SET_MAX_CONNECTIONS      1010   //"Insufficient
memory to
                                      //allocate # connections.
#define ERR_NOSUCH_CUSTOMER   1011    //"No such customer.
#define ERR_D_ID_INVALID      1012    //"Invalid District ID Must be 1 to
10.
#define ERR_MAX_CONNECT_PARAM 1013    //"Max client connections exceeded,
                                      //run install to increase.
#define ERR_INVALID_SYNC_CONNECTION 1014 //"Invalid Terminal Sync ID.
#define ERR_INVALID_TERMID    1015    //"Invalid Terminal ID.
#define ERR_PAYMENT_INVALID_CUSTOMER 1016 //"Payment Form, No such
Customer.
#define ERR_SQL_OPEN_CONNECTION       1017   //"SQLOpenConnection API
Failed.
#define ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY 1018  //"Stock Level missing
                                                   //Threshold key "TT*".
#define ERR_STOCKLEVEL_THRESHOLD_INVALID 1019 //"Stock Level Threshold
invalid
                                                //data type range = 1 - 99.
#define ERR_STOCKLEVEL_THRESHOLD_RANGE 1020   //"Stock Level Threshold
out of
                                              //range, range must be 1 - 99.
#define ERR_STOCKLEVEL_NOT_PROCESSED 1021  //"Stock Level not processed.
#define ERR_NEWORDER_FORM_MISSING_DID 1022 //"NewOrder miss District key
"DID*".
#define ERR_NEWORDER_DISTRICT_INVALID 1023 //"NewOrder District ID
Invalid
                                            //range 1 - 10.
#define ERR_NEWORDER_DISTRICT_RANGE  1024  //"New Order District ID out
of
                                            //Range. Range = 1 - 10.
#define ERR_NEWORDER_CUSTOMER_KEY 1025 //"New Order missing Customer key
"CID*".
#define ERR_NEWORDER_CUSTOMER_INVALID 1026 //"NewOrder customer id
invalid data
                                            //type, range = 1 to 3000.
#define ERR_NEWORDER_CUSTOMER_RANGE 1027   //"New Order customer id out
of
                                            //range, range = 1 to 3000.
#define ERR_NEWORDER_MISSING_IID_KEY 1028  //"NewOrder missng ItemId key
"IID*".
#define ERR_NEWORDER_ITEM_BLANK_LINES 1029 //"NewOrder blank order lines
all
                                            //orders must be continuous.
#define ERR_NEWORDER_ITEMID_INVALID 1030   //"New Order Item Id is wrong
data
                                            //type, must be numeric.
#define ERR_NEWORDER_MISSING_SUPPW_KEY 1031 //"NewOrder missing Supp_W
key
                                             //"SP##*".
#define ERR_NEWORDER_SUPPW_INVALID 1032          //"New Order Supp_W
invalid data
                                            //type must be numeric.
#define ERR_NEWORDER_MISSING_QTY_KEY 1033  //"NewOrder Missing Qty key
"Qty##*".
#define ERR_NEWORDER_QTY_INVALID 1034    //"NewOrder Qty invalid must be
                                            //numeric range 1 - 99.
#define ERR_NEWORDER_SUPPW_RANGE 1035            //"New Order Supp_W
value out of
                                            //range range = 1 - Max
Warehouses.
#define ERR_NEWORDER_ITEMID_RANGE 1036           //"New Order Item Id is
out of range.
                                            //Range = 1 to 999999.
#define ERR_NEWORDER_QTY_RANGE 1037    //"New Order Qty is out of range.
                                       //Range = 1 to 99.
#define ERR_PAYMENT_DISTRICT_INVALID 1038  //"Payment District ID is
invalid
                                            //must be 1 - 10.
#define ERR_NEWORDER_SUPPW_WITHOUT_ITEMID 1039 //"NewOrder Supp_W field
entered
                                              //without a corrosponding
Item_Id.
#define ERR_NEWORDER_QTY_WITHOUT_ITEMID 1040  //"NewOrder Qty entered
without a
```


Copyright © 2004 NEC Corporation

```
                                        //corrosponding Item_Id.
#define ERR_NEWORDER_NOITEMS_ENTERED  1041  //"NewOrder Blank Items
between
                                        //items, items must be
continuous.
#define ERR_PAYMENT_MISSING_DID_KEY 1042 //"Payment missing District Key
"DID*".
#define ERR_PAYMENT_DISTRICT_RANGE 1043       //"Payment District Out of
range,
                                     //range = 1 - 10.
#define ERR_PAYMENT_MISSING_CID_KEY 1044 //"Payment missing Customer Key
"CID*".
#define ERR_PAYMENT_CUSTOMER_INVALID 1045 //"PaymentCustomer data type
invalid,
                                       //must be numeric.
#define ERR_PAYMENT_MISSING_CLT       1046   //"Payment missing Customer
Last Name
                                     //Key "CLT*".
#define ERR_PAYMENT_LAST_NAME_TO_LONG 1047   //"Payment Customer last
name
                                         // longer than 16 characters.
#define ERR_PAYMENT_CUSTOMER_RANGE 1048      //"Payment Customer ID out
of range,
                                     //must be 1 to 3000.
#define ERR_PAYMENT_CID_AND_CLT       1049   //"Payment Customer ID and
Last Name
                                     //entered must be one or other.
#define ERR_PAYMENT_MISSING_CDI_KEY 1050 //"Payment missing Customer
district
                                      //key "CDI*".
#define ERR_PAYMENT_CDI_INVALID       1051   //"Payment Customer
district invalid
                                     //must be numeric.
#define ERR_PAYMENT_CDI_RANGE  1052   //"Payment Customer district out of
                                     //range must be 1 - 10.
#define ERR_PAYMENT_MISSING_CWI_KEY 1053 //"Payment missing Customer
Warehouse
                                      //key "CWI*".
#define ERR_PAYMENT_CWI_INVALID       1054   //"Payment Customer
Warehouse invalid
                                     //must be numeric.
#define ERR_PAYMENT_CWI_RANGE  1055   //"Payment Customer Warehouse out
of
                                     //range, 1 to Max Warehouses.
#define ERR_PAYMENT_MISSING_HAM_KEY 1056  //"Payment missing Amount key
"HAM*".
#define ERR_PAYMENT_HAM_INVALID       1057   //"Payment Amount invalid
data type
                                     //must be numeric.
#define ERR_PAYMENT_HAM_RANGE  1058   //"Payment Amount out of range,
                                     //0 - 9999.99.
#define ERR_ORDERSTATUS_MISSING_DID_KEY      1059 //"OrderStatus missing
District
                                         //key "DID*".
#defineERR_ORDERSTATUS_DID_INVALID 1060 //"Order Status District invalid,
                                      //value must be numeric 1 - 10.
#define ERR_ORDERSTATUS_DID_RANGE 1061       //"Order Status District
out of range
                                     //must be 1 - 10.
#define ERR_ORDERSTATUS_MISSING_CID_KEY 1062 //"OrderStatus missing
Customer
                                         //key "CID*".
#define ERR_ORDERSTATUS_MISSING_CLT_KEY 1063 //"OrderStatus missing
Customer
                                           //Last Name key "CLT*".
#define ERR_ORDERSTATUS_CLT_RANGE 1064       //"Order Status Customer
last name
                                       //longer than 16 characters.
#define ERR_ORDERSTATUS_CID_INVALID 1065 //"Order Status Customer ID
invalid,
                                        //range must be numeric 1 - 3000.
#define ERR_ORDERSTATUS_CID_RANGE 1066       //"Order Status Customer ID
out of
                                       //range must be 1 - 3000.
#define ERR_ORDERSTATUS_CID_AND_CLT 1067 //"Order Status Customer ID and
                                        //LastName entered must be only
one."
#define ERR_DELIVERY_MISSING_OCD_KEY 1068 //"Delivery missing Carrier ID
key
                                         //\"OCD*\".
#define ERR_DELIVERY_CARRIER_INVALID 1069 //"Delivery Carrier ID invalid
must
                                         //be numeric 1 - 10.
#define ERR_DELIVERY_CARRIER_ID_RANGE 1070 //"Delivery Carrier ID out of
range
                                          //must be 1 - 10.
#define ERR_PAYMENT_MISSING_CLT_KEY 1071   //"Payment missing Customer
Last
                                          //Name key "CLT*".
#define TOPEND_SEND_ERROR 1072 //"TOPEND client send error".
#define TOPEND_RECEIVE_ERROR 1073 //"TOPEND client receive error".

#ifdef LIMIT_TUXQUEUE
#define ERR_TUXQUEUE_TIMEOUT 1074
#define ERR_TUXQUEUE_FAILED 1075
#endif // LIMIT_TUXQUEUE

#ifdef LIMIT_TUXQUEUE2
#define ERR_TUXQUEUE_TIMEOUT 1074
#define ERR_TUXQUEUE_FAILED 1075
#endif // LIMIT_TUXQUEUE

#define ERR_TPINIT_BAD 5001  //"Tuxedo tpinit Failed."
#define ERR_TPALLOC_BAD 5002 //"Tuxedo tpalloc Failed."
#define ERR_TPCALL_BAD  5003 //"Tuxedo tpcall Failed."

#endif /* TPCCERR_H */
```

tpcc_info.h

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


Copyright © 2004 NEC Corporation

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

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

#ifndef TPCC_INFO_H
#define TPCC_INFO_H

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

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

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

struct newstruct {
  int    retval;
  int old_quantity[15];
  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;
  float 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];
```


Copyright © 2004 NEC Corporation

```
  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 {
  int retval;
  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 {
  int retval;
  struct ordinstruct ordin;
  struct ordoutstruct ordout;
};


/* Delivery */

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

struct deloutstruct {
  int terror;
  int retry;
};

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


/* Stock level */
```


Copyright © 2004 NEC Corporation

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

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

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

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

#endif
```

mod_tpcc.c

```
/*
==========================================================================
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2000-2003 The Apache Software Foundation.  All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 *    if any, must include the following acknowledgment:
 *       "This product includes software developed by the
 *        Apache Software Foundation (http://www.apache.org/)."
 *    Alternately, this acknowledgment may appear in the software itself,
 *    if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Apache" and "Apache Software Foundation" must
 *    not be used to endorse or promote products derived from this
 *    software without prior written permission. For written
 *    permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache",
 *    nor may "Apache" appear in their name, without prior written
 *    permission of the Apache Software Foundation.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR
IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE
 * DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE
FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF
 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY
WAY OUT
```


Copyright © 2004 NEC Corporation

```
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF
 * SUCH DAMAGE.
 *
==================================================================
=======
 *
 * This software consists of voluntary contributions made by many
 * individuals on behalf of the Apache Software Foundation.  For more
 * information on the Apache Software Foundation, please see
 * <http://www.apache.org/>.
 *
 * Portions of this software are based upon public domain software
 * originally written at the National Center for Supercomputing Applications,
 * University of Illinois, Urbana-Champaign.
 */

/*
 * TPCC Module.  Web server response handler for tpcc transactions
 *
 * <Location /tpcc.dll>
 * SetHandler tpcc
 * </Location>
 *
 *
 *
 * 11.01.2003 Initial Version
 *
 */

#define CORE_PRIVATE

#include "httpd.h"
#include "http_config.h"
#include "http_core.h"
#include "http_log.h"
#include "http_main.h"
#include "http_protocol.h"
#include "http_request.h"
#include "util_script.h"
#include "apr_strings.h"
#include "apr_lib.h"
#define APR_WANT_STRFUNC
#include "apr_want.h"
#include "ap_mpm.h"

#ifdef TUX
#include <tmenv.h>
#include <xa.h>
#include <atmi.h>
#include <Uunix.h>
#endif

#include "mod_tpcc.h"
#include "tpcc_info.h"
#include "tpccapi.h"

extern TERM *Term;
extern char szTpccLogPath[]; //path to html log file if logging turned on in registry.
extern char szErrorLogPath[];        //path to error log file.
extern BOOL    bLog ;
extern BOOL    dLog ;
extern BOOL    bGeneric;
extern int           iThreads ;
extern int           iMaxWareHouses;
extern int           iDelayMs  ;
extern short   iDeadlockRetry ;
extern short   iMaxConnections;
extern int     bTpccExit;      //exit delivery disconnect loop as dll exiting.
#define TPCC_MOD_CONFIG_KEY "tpcc_module"


module AP_MODULE_DECLARE_DATA tpcc_module;

tpcc_shm_t shmTerm ={"/tmp/tpcc_term",NULL,sizeof(TERM)};
tpcc_shm_t shmCData ={"/tmp/tpcc_cdata",NULL,sizeof(CLIENTDATA)*32};

loaded_t loaded;

tpcc_mod_conf* tpcc_config_global_create(server_rec *s)
{
        apr_pool_t *pool= s->process->pool;
        tpcc_mod_conf *mc;
        apr_pool_userdata_get((void **)&mc, TPCC_MOD_CONFIG_KEY, pool);
        if (mc)
        {
                return mc;
        }
```


Copyright © 2004 NEC Corporation

```
        mc = (tpcc_mod_conf*) apr_pcalloc(pool, sizeof(*mc));
        mc->cs = NULL;
        mc->pool =pool;
        // initialize shm data
        memcpy(&(mc->shmTerm),&shmTerm,sizeof(shmTerm));
        memcpy(&(mc->shmCData),&shmCData,sizeof(shmCData));



   // initialize global storage
    bLog      = FALSE;
    dLog      = FALSE;
    bGeneric  = FALSE;
    iMaxWareHouses    = 625;
    iThreads          = 5;
    iDelayMs          = 100;
    iDeadlockRetry    = (short)3;
    strcpy(szTpccLogPath, "tpcclog.");

//      InitializeCriticalSection(&CriticalSection);
//      InitializeCriticalSection(&ErrorLogCriticalSection);

        apr_pool_userdata_set(mc, TPCC_MOD_CONFIG_KEY,
                              apr_pool_cleanup_null,
                              pool);
        return mc;
}

static void *create_tpcc_config(apr_pool_t *p, server_rec *s)
{
   int i;
   info_svr_conf *conf = (info_svr_conf *) apr_pcalloc(p, sizeof(info_svr_conf));

   conf->mc = tpcc_config_global_create(s);

   return conf;
}

static void *merge_tpcc_config(apr_pool_t *p, void *basev, void *overridesv)
{
/*  no merging of configuration
   info_svr_conf *new = (info_svr_conf *) apr_pcalloc(p, sizeof(info_svr_conf));
   info_svr_conf *base = (info_svr_conf *) basev;
   info_svr_conf *overrides = (info_svr_conf *) overridesv;


   return new;
*/
   return basev;
}

static int tpcc_handler(request_rec *r)
{
   ext_ctrl_block* ecb = NULL;
   module *modp = NULL;
   const char *more_info;
   const command_rec *cmd = NULL;
#ifdef NEVERMORE
   const handler_rec *hand = NULL;
#endif
   server_rec *serv = r->server;
   int comma = 0;
   tpcc_mod_conf* mc;

   if (strcmp(r->handler, "tpcc"))
      return DECLINED;

   r->allowed |= (AP_METHOD_BIT << M_GET);
   if (r->method_number != M_GET)
         return DECLINED;


   ecb = (ext_ctrl_block*)apr_palloc ( r->pool, sizeof(ext_ctrl_block));
   if (ecb)
   {  // construct a lpszQueryString for isapi code
         char *aprQueryString;
         int len = strlen(r->uri)+ (r->args ? strlen(r->args)+3:1);

         ecb->r=r;
         aprQueryString = (char*)apr_palloc(r->pool, len);
         if (aprQueryString)
         {
         strcpy(aprQueryString, r->uri);
            if (r->args)
            {
                  strcat(aprQueryString, "?");
                  strcat(aprQueryString, r->args);
```


Copyright © 2004 NEC Corporation

```
            }
            ecb->lpszQueryString = aprQueryString;
        }
        else
        {
                return 0;
        }
    }
    else
    {
            return 0;
    }

    mc = myModConfig(r->server);
    Term = (TERM*) apr_shm_baseaddr_get(mc->shmTerm.shm);
    Term->pClientData = (PCLIENTDATA) apr_shm_baseaddr_get(mc-
>shmCData.shm);

     HttpExtensionProc(ecb);

    /* Done, turn off timeout, close file and return */
    return 0;
}

static const char *set_log_path(cmd_parms *cmd, void *dummy, const char *arg)
{
        strcpy(szTpccLogPath, arg);
                strcat(szTpccLogPath, "tpcclog");
            strcpy(szErrorLogPath, arg);
        strcat(szErrorLogPath, "tpccerr");


    return NULL;
}
static const char *set_shm_path(cmd_parms *cmd, void *dummy, const char *arg)
{

        tpcc_mod_conf* mc = myModConfig(cmd->server);
        mc->shmTerm.name= apr_psprintf(mc->pool,"%stpcc_term.%lu",
                        arg,(unsigned long) getpid());
        mc->shmCData.name= apr_psprintf(mc->pool,"%stpcc_cdata.%lu",
                        arg,(unsigned long) getpid());
        mc->szMutexFile= apr_psprintf(mc->pool, "%stpcc_mtx.%lu",
                        arg,(unsigned long) getpid());
            return NULL;
}

static const char *set_log(cmd_parms *cmd, void *dummy, const char *arg)
{

if ( !stricmp(arg, "ON") )
                bLog = TRUE;
    return NULL;
}

static const char *set_debug(cmd_parms *cmd, void *dummy, const char *arg)
{

if ( !stricmp(arg, "ON") )
                dLog = TRUE;
    return NULL;
}

static const char *set_max_warehouses(cmd_parms *cmd, void *dummy,
                                        const char *arg)
{
        iMaxWareHouses = atoi(arg);
                if ( iMaxWareHouses == 0 )
                        iMaxWareHouses = 500;

                        return NULL;
}
static const char *set_del_threads(cmd_parms *cmd, void *dummy,
                                        const char *arg)
{
        iThreads = atoi(arg);
                if ( !iThreads )
                        iThreads = 5;

                        return NULL;
}
static const char *set_backoff_delay(cmd_parms *cmd, void *dummy,
                                        const char *arg)
{
                iDelayMs = atoi(arg);
                    if ( !iDelayMs )
                            iDelayMs = 100;
```


Copyright © 2004 NEC Corporation

```
                        return NULL;
}

static const char *set_deadlock_retry(cmd_parms *cmd, void *dummy,
                                        const char *arg)
{
                iDeadlockRetry = (short)atoi(arg);
                        if ( !iDeadlockRetry )
                                        iDeadlockRetry = (short)3;

                        return NULL;
}

static const char *set_max_connections(cmd_parms *cmd, void *dummy,
                                        const char *arg)
{
                iMaxConnections = (short)atoi(arg);
                    if ( !iMaxConnections )
                                        iMaxConnections = (short)25;

                        return NULL;
}

static const char *set_gen_srvc(cmd_parms *cmd, void *dummy,
                                        const char *arg)
{
    if ( !stricmp(arg, "ON") )
                    bGeneric = TRUE;

                        return NULL;
}


static const command_rec info_cmds[] =
{
   AP_INIT_TAKE1("Path", set_log_path, NULL, RSRC_CONF,
                        "full path for tpcclog and tpccerr log files"),
   AP_INIT_TAKE1("Shm_Path", set_shm_path, NULL, RSRC_CONF,
                        "full path for share memory file"),
   AP_INIT_TAKE1("Log", set_log, NULL, RSRC_CONF,
                        "Enable error log"),
   AP_INIT_TAKE1("Debug", set_debug, NULL, RSRC_CONF,
                        "Enable debug log"),
   AP_INIT_TAKE1("MaximumWarehouses", set_max_warehouses, NULL,
RSRC_CONF,
                        "Maximum number of warehouses allowed"),
   AP_INIT_TAKE1("NumberOfDeliveryThreads", set_del_threads, NULL,
RSRC_CONF,
                        "Number of delivery threads"),
   AP_INIT_TAKE1("BackoffDelay", set_backoff_delay, NULL, RSRC_CONF,
                        "Backoff delay"),
   AP_INIT_TAKE1("DeadlockRetry", set_deadlock_retry, NULL, RSRC_CONF,
                        "Number deadlock retry"),
   AP_INIT_TAKE1("MaxConnections", set_max_connections, NULL, RSRC_CONF,
                        "Maximum connections"),
   AP_INIT_TAKE1("GenSrvc", set_gen_srvc, NULL, RSRC_CONF,
                        "Generic service (for TOPEND only)"),
   {NULL}
};

apr_status_t tpcc_init_modulekill(void *data)
{
        tpcc_mod_conf *mc;
        server_rec *base_server = (server_rec *)data;
        server_rec *s;

        mc = myModConfig(base_server);

        bTpccExit = TRUE;

        if (mc && mc->cs)
        {
                TERM* Term;
                Term = apr_shm_baseaddr_get(mc->shmTerm.shm);
                if (dLog)
                {
                        FILE *fp;

                        fp = fopen("/tmp/errlog", "ab");
                        fprintf(fp, "modulekill, pid=%d,
thread_id=%d,iAvail=%d\n", getpid(),GetCurrentThreadId(),Term->iAvailable);

                        fclose(fp);
                }

                apr_shm_destroy(mc->shmTerm.shm);
                apr_shm_destroy(mc->shmCData.shm);
```


Copyright © 2004 NEC Corporation

```
		unlink(mc->shmTerm.name);
		unlink(mc->shmCData.name);

		apr_global_mutex_destroy(mc->cs);
		mc->cs=NULL;
	}
		if (dLog)
		{
			FILE *fp;

			fp = fopen("/tmp/errlog", "ab");
			fprintf(fp, "aftermodulekill, pid=%d, thread_id=%d\n",
getpid(),GetCurrentThreadId());

			fclose(fp);
		}
   return APR_SUCCESS;
}

int tpcc_init_module(apr_pool_t *p, apr_pool_t *plog,
		apr_pool_t *ptemp,
				server_rec *base_server)
{
	tpcc_mod_conf* mc = myModConfig(base_server);
		/*
		 *	* Let us cleanup on restarts and exists
		 *	*/
		apr_pool_cleanup_register(p, base_server,
				tpcc_init_modulekill,
					apr_pool_cleanup_null);
		if (dLog)
			{
			FILE *fp;

			fp = fopen("/tmp/errlog", "ab");
			fprintf(fp, "*init Moudle pid=%d, thread_id=%d\n",
getpid(),GetCurrentThreadId());

			fclose(fp);
		}

	if (!mc->cs)
	{
		TERM* Term;
		if (dLog)
		{
			FILE *fp;

			fp = fopen("/tmp/errlog", "ab");
			fprintf(fp, "mccs *init Module pid=%d, thread_id=%d\n",
getpid(),GetCurrentThreadId());

			fclose(fp);
		}
		tpcc_shm_create(&(mc->shmTerm),mc->pool);   // Term
		tpcc_shm_create(&(mc->shmCData),mc->pool);   //ClientData

		apr_global_mutex_create(&(mc->cs),"/tmp/tpcc_mtx"/*mc-
>szMutexFile*/,APR_LOCK_DEFAULT,mc->pool);
		if (dLog)
		{
			FILE *fp;

			fp = fopen("/tmp/errlog", "ab");
			fprintf(fp, "before get  *init Moudle pid=%d,
thread_id=%d\n", getpid(),GetCurrentThreadId());

			fclose(fp);
		}
		Term = (TERM*) apr_shm_baseaddr_get(mc->shmTerm.shm);
		if (Term)
		{

		int i;
		Term->iAvailable=0;
		Term->iNext=0;
		Term->iMasterSyncId=0;
		Term->bInit=FALSE;
		Term->pClientData=NULL;

		TermInit(base_server);
		if ( !TermAllocate(base_server) )
		{

			FILE *fp;

			fp = fopen("/tmp/errlog", "ab");
```


Copyright © 2004 NEC Corporation

```
                        fprintf(fp, "Error Trm.Allocate Init, pid=%d,
thread_id=%d\n", getpid(),GetCurrentThreadId());

                        fclose(fp);
                }
                //Term = (TERM*) apr_shm_baseaddr_get(mc->shmTerm.shm);

                for(i=Term->iNext; i<Term->iAvailable; i++)
                  Term->pClientData[i].inUse = 0;
                Term->pClientData[0].inUse = 1;

                }
        }

        return APR_SUCCESS;

}

static int tpcc_pre_config(apr_pool_t* pconf, apr_pool_t *plog, apr_pool_t *ptemp)
{
        apr_status_t rv;
        apr_pool_sub_make(&loaded.pool, pconf, NULL);
                if (dLog)
                {
                        FILE *fp;

                        fp = fopen("/tmp/errlog", "ab");
                        fprintf(fp, "*init Moudle pid=%d, thread_id=%d\n",
getpid(),GetCurrentThreadId());

                        fclose(fp);
                }

  if (!loaded.pool) {
    ap_log_error(APLOG_MARK, APLOG_ERR, APR_EGENERAL, NULL,
            "tpcc: could not create the configuration pool");
    return APR_EGENERAL;
  }

  loaded.hash = apr_hash_make(loaded.pool);
  if (!loaded.hash) {
    ap_log_error(APLOG_MARK, APLOG_ERR, 0, NULL,
            "tpcc: Failed to create tls cache");
    return APR_EGENERAL;
  }

  rv = apr_thread_mutex_create(&loaded.lock, APR_THREAD_MUTEX_DEFAULT,
                        loaded.pool);
  if (rv != APR_SUCCESS) {
    ap_log_error(APLOG_MARK, rv, 0, NULL,
            "ISAPI: Failed to create module cache lock");
    return rv;
  }
  return OK;
}

static void register_hooks(apr_pool_t *p)
{
  ap_hook_pre_config(tpcc_pre_config, NULL,NULL, APR_HOOK_MIDDLE);
  ap_hook_handler(tpcc_handler, NULL, NULL, APR_HOOK_MIDDLE);
  ap_hook_post_config(tpcc_init_module, NULL,NULL,APR_HOOK_MIDDLE);
}

module AP_MODULE_DECLARE_DATA tpcc_module =
{
  STANDARD20_MODULE_STUFF,
  NULL,                 /* dir config creater */
  NULL,                 /* dir merger --- default is to override */
  create_tpcc_config,       /* server config */
  merge_tpcc_config,        /* merge server config */
  info_cmds,            /* command apr_table_t */
  register_hooks
};
```

mod_tpcc.h

```
#ifndef MOD_TPCC
#define MOD_TPCC

#define CORE_PRIVATE

#include "httpd.h"
#include "http_config.h"
#include "http_core.h"
#include "http_log.h"
#include "http_main.h"
#include "http_protocol.h"
```


Copyright © 2004 NEC Corporation

```
#include "http_request.h"
#include "util_script.h"
#include "apr_strings.h"
#include "apr_lib.h"
#define APR_WANT_STRFUNC
#include "apr_want.h"
#include "ap_mpm.h"

// map WIN32/IIS API types to apache types
typedef unsigned int DWORD;
typedef char*       LPTSTR;

#define GetCurrentThreadId()        apr_os_thread_current()
#define TlsAlloc(x)          (NULL)

#define HSE_STATUS_SUCCESS_AND_KEEP_CONN       1
#define HSE_STATUS_SUCCESS 0
#define TEXT(x)                      x

#define CRITICAL_SECTION             apr_global_mutex_t *

#define mySrvConfig(srv) (info_svr_conf*)ap_get_module_config(srv-
>module_config, &tpcc_module)
#define myModConfig(srv) (mySrvConfig((srv)))->mc

#define DestroyCriticalSection(x)       apr_global_mutex_destroy(*(x))

#define InitializeCriticalSection(x)

#define mcInitializeCriticalSection(x)


#define mcEnterCriticalSection(x)              \
{\
        apr_thread_mutex_lock(x);\
}

#define mcLeaveCriticalSection(x)   \
{\
        apr_thread_mutex_unlock(x);\
}

#define EnterCriticalSection(ax)                \
{\
        tpcc_mod_conf* mc = (tpcc_mod_conf *)myModConfig(pECB->r->server);\
        apr_global_mutex_lock(mc->cs);\
        Term = apr_shm_baseaddr_get(mc->shmTerm.shm);\
        Term->pClientData = (PCLIENTDATA)apr_shm_baseaddr_get(mc-
>shmCData.shm);\
}

#define LeaveCriticalSection(ax)   \
{\
        tpcc_mod_conf* mc = (tpcc_mod_conf *)myModConfig(pECB->r->server);\
        apr_global_mutex_unlock(mc->cs);\
}

#define SET_TERM_AND_CDATA           \
{\
        tpcc_mod_conf* mc = (tpcc_mod_conf *)myModConfig(pECB->r->server);\
        Term = apr_shm_baseaddr_get(mc->shmTerm.shm);\
        Term->pClientData = (PCLIENTDATA)apr_shm_baseaddr_get(mc-
>shmCData.shm);\
}


#define NO_ERROR        0
#define GetLastError()      NO_ERROR

#define wsprintf  sprintf
#define _ftime              ftime
#define stricmp             strcasecmp
#define Sleep(x)  usleep((x)*1000)
#define GetTickCount()      apr_time_as_msec(apr_time_now())

#define WriteMessageToEventLog(x)                    write_err_log(pECB->r, x)

#define WR_LOG(x)       write_err_log(pECB->r,x)
#define TPCC_LOG(x)     {           \
                        FILE *fp;\
                        fp = fopen(szTpccLogPath, "ab");\
                        fprintf(fp, x" Thread %d pid= %d\n", \
                                GetCurrentThreadId(),getpid() );\
                        fclose(fp);\
                }

typedef struct                          /* simulate ECB in apache */
```


Copyright © 2004 NEC Corporation

```
{
 request_rec *r;
 char * lpszQueryString;
} ext_ctrl_block;

typedef struct
{
        pid_t pid;
        apr_os_thread_t tid;
} tls_t;

typedef struct
{
        char *name;
        apr_shm_t *shm;
        apr_size_t size;
        void *addr;
}
tpcc_shm_t;
typedef struct {
  apr_global_mutex_t *cs;
  tpcc_shm_t shmTerm;
  tpcc_shm_t shmCData;
  apr_pool_t*      pool;
  char* szMutexFile;
} tpcc_mod_conf;


typedef struct {
   apr_array_header_t *more_info;
   tpcc_mod_conf* mc;
}  info_svr_conf;

extern module AP_MODULE_DECLARE_DATA tpcc_module;

typedef struct tpcc_global_conf {
   apr_pool_t        *pool;
   apr_thread_mutex_t *lock;
   apr_hash_t        *hash;
} loaded_t;

extern loaded_t loaded;
#endif
```

Makefile.am

```
## This is the shared library to be built
lib_LTLIBRARIES = libmodtpcc.la

## Define the source file for the module
libmodtpcc_la_SOURCES = mod_tpcc.c tpcc.c

## Define that an include directory is required.
INCLUDES = -DTUX -I@apache_dir@/include  -I../tm_src -I${TUXDIR}/include

LDFLAGS = -L@apache_dir@/lib -lapr-0 -L${TUXDIR}/lib -ltux -lcobatmi -
lgw -lnative -ltmib
```

build.sh

```
make
/usr/local/apache2/bin/apxs -i -a -n tpcc libmodtpcc.la
```

configure.in

```
# Required initializer
AC_INIT

# Automake initialization
AM_INIT_AUTOMAKE(mod_tpcc, 1.0)

# Add a test for a compiler.
AC_PROG_CC
AM_PROG_LIBTOOL

# Define a macro that is used to parse a --with-apache parameter
# The macro is named "APACHE_DIR"
AC_DEFUN([APACHE_DIR],[

        AC_ARG_WITH(
                apache,
                [  --with-apache[=DIR]     Apache server directory],
                ,
                [with_apache="no"]
        )

        AC_MSG_CHECKING(for Apache directory)
```


Copyright © 2004 NEC Corporation

```
        if test "$with_apache" = "no"; then
                AC_MSG_ERROR( You need to specify the apache directory using
--with-apache)
        else
                # make sure that a well known include file exists
                if test -e $with_apache/include/httpd.h; then
                        apache_dir=$with_apache
                        AC_MSG_RESULT(APACHE found!)
                else
                        AC_MSG_ERROR( $with_apache not found.  Check the
value you specified with --with-apache)
                fi
        fi

])

# Now call the APACHE_DIR macro that was just specified
APACHE_DIR

# Save the location of apache into the "apache_dir" variable.
# The AC_SUBST macro causes the variable to be saved in config.status
AC_SUBST(apache_dir)

# Write config.status and the Makefile
AC_OUTPUT(Makefile)
```

A2. Back-End Source Code, Scripts

pldel.c

```
#ifdef RCSID
static char *RCSid =
  "$Header: pldel.c 7030100.5 96/06/24 16:26:06 plai Generic<base> $ Copyr (c)
1994 Oracle";
#endif /* RCSID */

/*==================================================================
======+
 |      Copyright (c) 1996  Oracle Corp, Redwood Shores, CA       |
 |                OPEN SYSTEMS PERFORMANCE GROUP                  |
 |                    All Rights Reserved                         |
 +==================================================================
======+
 | FILENAME
 |    pldel.c
 | DESCRIPTION
 |    OCI version of DELIVERY transaction in TPC-C benchmark.

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

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

/*
extern int userlog();
*/

#define DMLRETDEL

#define SQLTXT "BEGIN inittpcc.init_del ; END;"

#define SQLTXT1 "DELETE FROM nord WHERE no_d_id = :d_id \
                AND no_w_id = :w_id and rownum <= 1 \
               RETURNING no_o_id into :o_id "

#define SQLTXT3 "UPDATE ordr 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"

#define SQLTXT4 "UPDATE ordl \
   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 sum(ol_amount) into :ol_amount "

#define SQLTXT6 "UPDATE cust 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
```


Copyright © 2004 NEC Corporation

```
struct delctx {
  sb2 del_o_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];

  ub4 del_o_id_len[NDISTS];
  ub4 c_id_len[NDISTS];
  int oid_ctx;
  int cid_ctx;
  OCIBind *olamt_bp;

  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 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];

  int del_o_id[NDISTS];
  int del_d_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];
  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;
struct pldelctx {
```


Copyright © 2004 NEC Corporation

```
  ub2 del_d_id_len[NDISTS];
  ub2 del_o_id_len[NDISTS];


  ub2 w_id_len;
  ub2 d_id_len[NDISTS];
  ub2 o_c_id_len[NDISTS];
  ub2 sums_len[NDISTS];
  ub2 carrier_id_len;
  ub2 ordcnt_len;
  ub2 del_date_len;

  int del_o_id[NDISTS];
  int del_d_id[NDISTS];
  int o_c_id[NDISTS];
  float sums[NDISTS];
  OCIDate del_date;
  int carrier_id;
  int ordcnt;

  ub4 del_o_id_rcnt;
  ub4 del_d_id_rcnt;
  ub4 o_c_id_rcnt;
  ub4 sums_rcnt;

  int retry;
  OCIStmt *curp1;
  OCIStmt *curp2;
  OCIBind *w_id_bp;
  OCIBind *d_id_bp;
  OCIBind *o_id_bp;
  OCIBind *o_c_id_bp;
  OCIBind *ordcnt_bp;
  OCIBind *sums_bp;
  OCIBind *del_date_bp;
  OCIBind *carrier_id_bp;
  OCIBind *retry_bp;

  int norow;

};
typedef struct pldelctx pldelctx;
```

```
static pldelctx *pldctx;


static delctx *dctx;

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

#endif


#ifdef DMLRETDEL
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);
}


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);
}
```


Copyright © 2004 NEC Corporation

```
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);
}


# ifdef OLD
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=actx->ol_cnt+1;
  *bufpp = &actx->ol_amt[index];
  *indpp= &actx->ol_amt_ind[index];
  actx->ol_amt_len[index]=sizeof(actx->ol_amt[0]);
  *alenp= &actx->ol_amt_len[index];
  *rcodepp = &actx->ol_amt_rcode[index];
  *piecep =OCI_ONE_PIECE;
  if (iter ==1 )
    return (OCI_CONTINUE);
  else
    return (OCI_ERROR);
}
# else
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;
  *bufpp = &actx->ol_amt[index];
  *indpp= &actx->ol_amt_ind[index];
  actx->ol_amt_len[index]=sizeof(actx->ol_amt[0]);
  *alenp= &actx->ol_amt_len[index];
  *rcodepp = &actx->ol_amt_rcode[index];
  *piecep =OCI_ONE_PIECE;
  return (OCI_CONTINUE);
}
# endif

#endif


tkvcdinit (int plsqlflag)

{

  text stmbuf[SQL_BUF_SIZE];

  if (plsqlflag)
  {
    pldctx = (pldelctx *) malloc (sizeof(pldelctx));
    DISCARD memset(pldctx,(char)0,(ub4)sizeof(pldelctx));
    /* Initialize */
    DISCARD OCIHandleAlloc(tpcenv, (dvoid**)&pldctx->curp1,
OCI_HTYPE_STMT, 0,
                    (dvoid**)0);
    DISCARD sprintf ((char *) stmbuf, SQLTXT);
    DISCARD OCIStmtPrepare(pldctx->curp1, errhp, stmbuf,
                    (ub4) strlen((char *)stmbuf),
                OCI_NTV_SYNTAX, OCI_DEFAULT);
    DISCARD OCIERROR(errhp,
      OCIStmtExecute(tpcsvc,pldctx->curp1,errhp,1,0,NULLP(OCISnapshot),
                NULLP(OCISnapshot), OCI_DEFAULT));


    DISCARD OCIHandleAlloc(tpcenv,(dvoid**) &pldctx->curp2,
OCI_HTYPE_STMT,
                0, (dvoid**)0);
#if defined(ISO5) || defined(ISO6) || defined(ISO8)
  #if defined(ISO5)
    sqlfile("../blocks/tkvcpdel_iso5.sql",stmbuf);
  #endif
  #if defined(ISO6)
    sqlfile("../blocks/tkvcpdel_iso6.sql",stmbuf);
  #endif
  #if defined(ISO8)
    sqlfile("../blocks/tkvcpdel_iso8.sql",stmbuf);
```


Copyright © 2004 NEC Corporation

```
  #endif
#else
    sqlfile("../blocks/tkvcpdel.sql",stmbuf);
#endif
    DISCARD OCIStmtPrepare(pldctx->curp2, errhp, stmbuf,
                (ub4)strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT);
    OCIBNDPL(pldctx->curp2, pldctx->w_id_bp , errhp,":w_id",
            ADR(w_id), SIZ(int), SQLT_INT,&pldctx->w_id_len);
    OCIBNDPL(pldctx->curp2, pldctx->ordcnt_bp , errhp,":ordcnt",
            ADR(pldctx->ordcnt), SIZ(int), SQLT_INT,&pldctx->ordcnt_len);
//     OCIBNDPL(pldctx->curp2, pldctx->del_date_bp,errhp,":now",
//             dctx->del_date, SIZ(OCIDate), SQLT_ODT,&pldctx->del_date_len);
//bug??
             OCIBNDPL(pldctx->curp2, pldctx->del_date_bp,errhp,":now",
            ADR(pldctx->del_date), SIZ(OCIDate), SQLT_ODT,&pldctx-
>del_date_len);
    OCIBNDPL(pldctx->curp2, pldctx->carrier_id_bp , errhp,
            ":carrier_id", ADR(o_carrier_id), SIZ(int),
            SQLT_INT, &pldctx->carrier_id_len);

    OCIBNDPLA(pldctx->curp2, pldctx->d_id_bp, errhp,":d_id",
             pldctx->del_d_id, SIZ(int),SQLT_INT, pldctx->del_d_id_len,
             NDISTS, &pldctx->del_d_id_rcnt);
    OCIBNDPLA(pldctx->curp2, pldctx->o_id_bp, errhp,":order_id",
             pldctx->del_o_id,SIZ(int),SQLT_INT, pldctx->del_o_id_len,NDISTS,
             &pldctx->del_o_id_rcnt);
    OCIBNDPLA(pldctx->curp2, pldctx->sums_bp, errhp,":sums",
             pldctx->sums,SIZ(float),SQLT_BFLOAT, pldctx->sums_len,NDISTS,
             &pldctx->sums_rcnt);
    OCIBNDPLA(pldctx->curp2, pldctx->o_c_id_bp, errhp,":o_c_id",
             pldctx->o_c_id,SIZ(int),SQLT_INT, pldctx->o_c_id_len,NDISTS,
             &pldctx->o_c_id_rcnt);
    OCIBND(pldctx->curp2, pldctx->retry_bp , errhp,":retry",
           ADR(pldctx->retry), SIZ(int),SQLT_INT);

   }
   else
   {

    dctx = (delctx *) malloc (sizeof(delctx));
    memset(dctx,(char)0,sizeof(delctx));
    dctx->norow = 0;
    actx = (amtctx *) malloc (sizeof(amtctx));
    memset(actx,(char)0,sizeof(amtctx));
    OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd1), OCI_HTYPE_STMT, 0,
            (dvoid**)0);
    DISCARD sprintf ((char *) stmbuf, "%s",    SQLTXT1);
    DISCARD 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);

   /* open third cursor */

    DISCARD OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd3),
OCI_HTYPE_STMT,
                    0, (dvoid**)0);
    DISCARD sprintf ((char *) stmbuf, SQLTXT3);
    DISCARD 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);

    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);

   /* open fourth cursor */

    DISCARD OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd4),
```


Copyright © 2004 NEC Corporation

```
OCI_HTYPE_STMT, 0,
                (dvoid**)0);
    DISCARD sprintf ((char *) stmbuf, SQLTXT4);
    DISCARD 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);
    OCIBNDRAD(dctx->curd4, dctx->olamt_bp, errhp, ":ol_amount",
           SIZ(int), SQLT_INT,NULL, actx,no_data,amt_data);



  /* open sixth cursor */

    DISCARD OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd6),
OCI_HTYPE_STMT,
                     0, (dvoid**)0);
    DISCARD sprintf ((char *) stmbuf, SQLTXT6);
    DISCARD 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);
  }
  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 plsqlflag)

{

  int i, j;
  int rpc,rcount,count;
  int invalid;

  if (plsqlflag)
  {

    pldctx->w_id_len = sizeof (int);
    pldctx->carrier_id_len = sizeof (int);
    for (i = 0; i < NDISTS; i++)
    {
      pldctx->del_o_id_len[i] = sizeof(int);
      del_o_id[i] = 0;
    }
    pldctx->del_date_len = DEL_DATE_LEN;
    DISCARD memcpy(&pldctx->del_date,&cr_date,sizeof(OCIDate));

    pldctx->retry=0;

    DISCARD OCIERROR(errhp,
      OCIStmtExecute(tpcsvc,pldctx->curp2,errhp,1,0,NULLP(CONST OCISnapshot),
                NULLP(OCISnapshot),OCI_DEFAULT));
```


Copyright © 2004 NEC Corporation

```
    for (i = 0; i < NDISTS; i++)
    {
      del_o_id[i] = 0;
    }
    for (i = 0; i < pldctx->del_o_id_rcnt; i++)
      del_o_id[pldctx->del_d_id[i] - 1] = pldctx->del_o_id[i];
  }
  else
  {


retry:

    invalid = 0;

  /* initialization for array operations */

    for (i = 0; i < NDISTS; i++)
    {
      dctx->del_o_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->del_o_id_len[i] = SIZ(dctx->del_o_id[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->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));
    }

    memset(actx,(char)0,sizeof(amtctx));

  /* array select from new_order and orders tables */

  execstatus=OCIStmtExecute(tpcsvc,dctx->curd1,errhp,NDISTS,0,
          NULLP(CONST OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);
  if((execstatus != OCI_SUCCESS) && (execstatus != OCI_NO_DATA))
  {
    DISCARD 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 */
  DISCARD OCIAttrGet(dctx->curd1,OCI_HTYPE_STMT,&rcount,NULLP(ub4),
          OCI_ATTR_ROW_COUNT,errhp);
  rpc = rcount;
  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);
    }
  }

  execstatus=OCIStmtExecute(tpcsvc,dctx->curd3,errhp,rpc,0,
          NULLP(CONST OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);
```


Copyright © 2004 NEC Corporation

```
    if(execstatus != OCI_SUCCESS)
    {
     DISCARD 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;
     }
    }

    DISCARD OCIAttrGet(dctx->curd3,OCI_HTYPE_STMT,&rcount,NULLP(ub4),
          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
     DISCARD fprintf (stderr,
          "Error in TPC-C server %d: %d rows selected, %d ords updated\n",
          proc_no, rpc, rcount);
#endif
     DISCARD OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     return (-1);
    }

    /* array update of order_line table */
    execstatus=OCIStmtExecute(tpcsvc,dctx->curd4,errhp,rpc,0,
          NULLP(CONST OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);
    if(execstatus != OCI_SUCCESS)
    {
     DISCARD 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;
     }
    }
    DISCARD OCIAttrGet(dctx->curd4,OCI_HTYPE_STMT,&rcount,NULLP(ub4),
          OCI_ATTR_ROW_COUNT,errhp);
/* transfer amounts  */
    for (i=0;i<rpc;i++)
    {
     dctx->amt[i]=0;
      if ( actx->ol_amt_rcode[i] == 0)
      {
       dctx->amt[i] = actx->ol_amt[i];
      }
    }
#ifdef OLD
    if (rcount > rpc) {
      userlog
        ("Error in TPC-C server %d: %d ordnrs updated, %d ordl updated\n",
          proc_no, rpc, rcount);
    }
#endif
```


Copyright © 2004 NEC Corporation

```
    /* array update of customer table */
    execstatus=OCIStmtExecute(tpcsvc,dctx->curd6,errhp,rpc,0,
              NULLP(CONST OCISnapshot),NULLP(OCISnapshot),
              OCI_COMMIT_ON_SUCCESS | OCI_DEFAULT);

    if(execstatus != OCI_SUCCESS)
    {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     errcode = OCIERROR(errhp,execstatus);
     if(errcode == NOT_SERIALIZABLE)
     {
      retries++;
      goto retry;
     }
     else if (errcode == RECOVERR)
     {
      retries++;
      goto retry;
     }
     else if (errcode == SNAPSHOT_TOO_OLD)
     {
      retries++;
      goto retry;
     }
     else
     {
      return -1;
     }
    }

    DISCARD OCIAttrGet(dctx->curd6,OCI_HTYPE_STMT,&rcount,NULLP(ub4),
                 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
      DISCARD fprintf (stderr,
            "Error in TPC-C server %d: %d rows selected, %d cust updated\n",
            proc_no, rpc, rcount);
#endif
      DISCARD OCITransRollback(tpcsvc, errhp, OCI_DEFAULT);
      return (-1);
    }

   /* 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];
   }
   return (0);

}

void tkvcddone (int plsqlflag)

{

   if (plsqlflag)
   {
    if (pldctx)
    {
      DISCARD OCIHandleFree((dvoid *)dctx->curd0,OCI_HTYPE_STMT);
      DISCARD free(pldctx);
    }
   }
   else
   {
    if (dctx)
    {
      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);
      DISCARD free (dctx);
    }
   }

}
```

plnew.c

Copyright © 2004 NEC Corporation

```
#ifdef RCSID
static char *RCSid =
   "$Header: tkvcnew.c 21-apr-98.18:32:59 rdecker Exp $ Copyr (c) 1994
Oracle";
#endif /* RCSID */

/*===================================================================+
 |        Copyright (c) 1996 , 1997, 1998 Oracle Corp, Redwood Shores,
CA       |
 |                    OPEN SYSTEMS PERFORMANCE GROUP                 |
 |                         All Rights Reserved                       |
 +===================================================================+
 | FILENAME
 |    plnew.c
 | DESCRIPTION
 |    OCI version (using PL/SQL stored procedure) of
 |    NEW ORDER transaction in TPC-C benchmark.
 +===================================================================*/

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


#ifndef ORA_TPCC
# define ORA_TPCC
# include "tpcc.h"
#endif

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

#define SQLTXT2 "BEGIN inittpcc.init_no(:idx1arr); END;"

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

struct newctx {

  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 s_dist_info_len[NITEMS];
  ub2 ol_o_id_len[NITEMS];
  ub2 ol_number_len[NITEMS];
  ub2 s_remote_len[NITEMS];
  ub2 s_quant_len[NITEMS];
  ub2 ol_dist_info_len[NITEMS];
  ub2 s_bg_len[NITEMS];

  int ol_o_id[NITEMS];
  int ol_number[NITEMS];

  float s_remote[NITEMS];
  char s_dist_info[NITEMS][25];
  OCIStmt *curn1;
  OCIBind *ol_i_id_bp;
  OCIBind *ol_supply_w_id_bp;
  OCIBind *i_price_bp;
  OCIBind *i_name_bp;
  OCIBind *s_bg_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;
  OCIStmt *curn2;
  OCIBind *ol_quantity_bp;
  OCIBind *s_remote_bp;
  OCIBind *s_quantity_bp;
  OCIBind *w_id_bp;
  OCIBind *d_id_bp;
  OCIBind *c_id_bp;
  OCIBind *o_all_local_bp;
  OCIBind *o_all_cnt_bp;
  OCIBind *w_tax_bp;
  OCIBind *d_tax_bp;
  OCIBind *o_id_bp;
  OCIBind *c_discount_bp;
  OCIBind *c_credit_bp;
  OCIBind *c_last_bp;
  OCIBind *retries_bp;
  OCIBind *cr_date_bp;
  OCIBind *ol_o_id_bp;
  OCIBind *ol_amount_bp;

  sb2 w_id_len;
  ub2 d_id_len;
  ub2 c_id_len;
  ub2 o_all_local_len;
  ub2 o_ol_cnt_len;
  ub2 w_tax_len;
  ub2 d_tax_len;
  ub2 o_id_len;
  ub2 c_discount_len;
  ub2 c_credit_len;
  ub2 c_last_len;
  ub2 retries_len;
  ub2 cr_date_len;
};

typedef struct newctx newctx;

static newctx *nctx;


tkvcninit ()
{

  int i;
  text stmbuf[32*1024];

  nctx = (newctx *) malloc (sizeof(newctx));

  DISCARD memset(nctx,(char)0,sizeof(newctx));
  nctx->w_id_len = sizeof(w_id);
  nctx->d_id_len = sizeof(d_id);
```


Copyright © 2004 NEC Corporation

```
  nctx->c_id_len = sizeof(c_id);
  nctx->o_all_local_len = sizeof(o_all_local);
  nctx->o_ol_cnt_len = sizeof(o_ol_cnt);
  nctx->w_tax_len = 0;
  nctx->d_tax_len = 0;
  nctx->o_id_len = sizeof(o_id);
  nctx->c_discount_len = 0;
  nctx->c_credit_len = 0;
  nctx->c_last_len = 0;
  nctx->retries_len = sizeof(retries);
  nctx->cr_date_len = sizeof(cr_date);

  /* open first cursor */
  DISCARD OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&nctx->curn1),
           OCI_HTYPE_STMT, 0, (dvoid**)0));
#if defined(ISO)
  sqlfile("../blocks/tkvcpnew_iso.sql",stmbuf);
#else
#if defined(ISO7)
  sqlfile("../blocks/tkvcpnew_iso7.sql",stmbuf);
#else
  sqlfile("../blocks/tkvcpnew.sql",stmbuf);


#endif
#endif

  DISCARD OCIERROR(errhp,OCIStmtPrepare(nctx->curn1, errhp, stmbuf,
           strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));

  /* bind variables */

  OCIBNDPL(nctx->curn1, nctx->w_id_bp, errhp,
":w_id",ADR(w_id),SIZ(w_id),
          SQLT_INT, &nctx->w_id_len);


  OCIBNDPL(nctx->curn1, nctx->d_id_bp, errhp,
":d_id",ADR(d_id),SIZ(d_id),
          SQLT_INT, &nctx->d_id_len);

  OCIBNDPL(nctx->curn1, nctx->c_id_bp, errhp,
":c_id",ADR(c_id),SIZ(c_id),
          SQLT_INT, &nctx->c_id_len);


  OCIBNDPL(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_len);


  OCIBNDPL(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_len);

  OCIBNDPL(nctx->curn1, nctx->w_tax_bp, errhp,
":w_tax",ADR(w_tax),SIZ(w_tax),
          SQLT_FLT, &nctx->w_tax_len);

  OCIBNDPL(nctx->curn1, nctx->d_tax_bp, errhp,
":d_tax",ADR(d_tax),SIZ(d_tax),
          SQLT_FLT, &nctx->d_tax_len);

  OCIBNDPL(nctx->curn1, nctx->o_id_bp, errhp,
":o_id",ADR(o_id),SIZ(o_id),
          SQLT_INT, &nctx->o_id_len);

  OCIBNDPL(nctx->curn1, nctx->c_discount_bp, errhp, ":c_discount",
          ADR(c_discount), SIZ(c_discount),SQLT_FLT, &nctx-
>c_discount_len);

  OCIBNDPL(nctx->curn1, nctx->c_credit_bp, errhp, ":c_credit",c_credit,
          SIZ(c_credit),SQLT_CHR, &nctx->c_credit_len);

  OCIBNDPL(nctx->curn1, nctx->c_last_bp, errhp,
":c_last",c_last,SIZ(c_last),
          SQLT_STR, &nctx->c_last_len);

  OCIBNDPL(nctx->curn1, nctx->retries_bp, errhp, ":retry",ADR(retries),
          SIZ(retries),SQLT_INT, &nctx->retries_len);

  OCIBNDPL(nctx->curn1, nctx->cr_date_bp, errhp, ":cr_date",&cr_date,
          SIZ(OCIDate), SQLT_ODT, &nctx->cr_date_len);

  OCIBNDPLA(nctx->curn1, nctx->ol_i_id_bp,errhp,":ol_i_id",nol_i_id,
           SIZ(int), SQLT_INT, nctx->nol_i_id_len,NITEMS,&nctx-
>nol_i_count);

  OCIBNDPLA(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_len,
           NITEMS, &nctx->nol_s_count);


  OCIBNDPLA(nctx->curn1, nctx->ol_quantity_bp,errhp,":ol_quantity",
           nol_quantity, SIZ(float),SQLT_BFLOAT,nctx->nol_quantity_len,
           NITEMS,&nctx->nol_q_count);


  OCIBNDPLA(nctx->curn1, nctx-
>i_price_bp,errhp,":i_price",i_price,SIZ(float),
           SQLT_BFLOAT, nctx->i_price_len, NITEMS, &nctx-
>nol_item_count);

  OCIBNDPLA(nctx->curn1, nctx->i_name_bp,errhp,":i_name",i_name,
           SIZ(i_name[0]),SQLT_STR, nctx->i_name_len,NITEMS,
           &nctx->nol_name_count);

  OCIBNDPLA(nctx->curn1, nctx-
>s_quantity_bp,errhp,":s_quantity",s_quantity,
           SIZ(float), SQLT_BFLOAT,nctx->s_quant_len,NITEMS,&nctx-
>nol_qty_count);

  OCIBNDPLA(nctx->curn1, nctx-
>s_bg_bp,errhp,":brand_generic",brand_generic,
           SIZ(char), SQLT_CHR,nctx->s_bg_len,NITEMS,&nctx-
>nol_bg_count);

  OCIBNDPLA(nctx->curn1, nctx-
>ol_amount_bp,errhp,":ol_amount",nol_amount,
           SIZ(float),SQLT_BFLOAT, nctx->nol_amount_len,NITEMS,&nctx-
>nol_am_count);

  OCIBNDPLA(nctx->curn1, nctx->s_remote_bp,errhp,":s_remote",nctx-
>s_remote,
```


Copyright © 2004 NEC Corporation

```
          SIZ(float),SQLT_BFLOAT, nctx->s_remote_len,NITEMS,&nctx-
>s_remote_count);

  /* open second cursor */
  DISCARD OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)(&nctx->curn2),
                   OCI_HTYPE_STMT, 0, (dvoid**)0));

  DISCARD sprintf ((char *) stmbuf, SQLTXT2);
  DISCARD 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 */
    OCIBNDPLA(nctx->curn2, idx1arr_bp,errhp,":idx1arr",idx1arr,
             SIZ(int), SQLT_INT, idx1arr_len, NITEMS,&idx1arr_count);

     execstatus = OCIStmtExecute(tpcsvc,nctx->curn2,errhp,1,0,
                  NULLP(CONST
OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);


     if(execstatus != OCI_SUCCESS) {


       OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
       errcode = OCIERROR(errhp,execstatus);


       return -1;
     }
   }



  return (0);

}



tkvcn ()
{
```

```
   int i;
   int rcount;

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] = (float)1;
         o_all_local = 0;
      }
      else
         nctx->s_remote[i] = 0;
   }

   nctx->w_id_len = sizeof(w_id);
   nctx->d_id_len = sizeof(d_id);
   nctx->c_id_len = sizeof(c_id);
   nctx->o_all_local_len = sizeof(o_all_local);
   nctx->o_ol_cnt_len = sizeof(o_ol_cnt);
   nctx->w_tax_len = 0;
   nctx->d_tax_len = 0;
   nctx->o_id_len = sizeof(o_id);
   nctx->c_discount_len = 0;
   nctx->c_credit_len = 0;
   nctx->c_last_len = 0;
   nctx->retries_len = sizeof(retries);
   nctx->cr_date_len = sizeof(cr_date);
   /* 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;

   /* initialization for array operations */
   for (i = 0; i < o_ol_cnt; i++) {
      nctx->ol_number[i] = i + 1;
      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_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->s_remote_len[i] = sizeof(int);
      nctx->s_quant_len[i] = sizeof(int);
```


Copyright © 2004 NEC Corporation

```
      nctx->i_name_len[i]=0;
      nctx->s_bg_len[i] = 0;
   }
   for (i = o_ol_cnt; i < NITEMS; i++) {

      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_o_id_len[i] = 0;
      nctx->ol_number_len[i] = 0;
      nctx->ol_dist_info_len[i] = 0;
      nctx->s_remote_len[i] = 0;
      nctx->s_quant_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);


   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;
     }
   }

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


   total_amount = 0;
   for (i = 0; i < o_ol_cnt; i++) total_amount += nol_amount[i];
   total_amount *= ((float)(1.0 - c_discount)) * (float)(1.0 +
((float)(d_tax)) + ((float) (w_tax)));
   total_amount = total_amount/100;

   return (0);

}



void tkvcndone ()

{
   int i;

   if (nctx)
   {
     DISCARD OCIHandleFree((dvoid *)nctx->curn1,OCI_HTYPE_STMT);
     DISCARD OCIHandleFree((dvoid *)nctx->curn2,OCI_HTYPE_STMT);
     free (nctx);
   }
}
```

plord.c

```
/* Copyright (c) 2002, Oracle Corporation.  All rights reserved.  */

/*

   NAME
     tkvcordq.c - OCI version  using queues of ORDER STATUS
                  transaction in TPC-C benchmark.

   DESCRIPTION
     <short description of facility this file declares/defines>

   EXPORT FUNCTION(S)

   INTERNAL FUNCTION(S)
     <other external functions defined - one-line descriptions>

   STATIC FUNCTION(S)
     <static functions defined - one-line descriptions>

   NOTES
     <other useful comments, qualifications, etc.>

   MODIFIED   (MM/DD/YY)
   xnie        06/25/02 - queue open cluster join.
   heri        05/07/02 - Fix error in cursor.
   heri        02/01/02 - Cleanup, remove indicator values and return
codes.
   lwang       07/25/01 - Merged lwang_tpccitrc
   lwang       07/23/01 - fix include
   lwang       07/23/01 - Creation

*/


#include "tpcc.h"

/*---------------------------------------------------------------------
-----
                      PRIVATE TYPES AND CONSTANTS
  ---------------------------------------------------------------------
-----*/

/*---------------------------------------------------------------------
-----
                      STATIC FUNCTION DECLARATIONS
  ---------------------------------------------------------------------
-----*/
```


Copyright © 2004 NEC Corporation

```
#define SQLCUR0 "SELECT rowid FROM cust \
                 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 /*+ USE_NL(cust) INDEX_DESC(ordr iordr2) */  \
                 c_id, c_balance, c_first, c_middle, c_last, \
                 o_id, o_entry_d, o_carrier_id, o_ol_cnt, ordr.rowid \
                 FROM cust, ordr \
                 WHERE cust.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 DESC, o_d_id DESC, o_w_id DESC, o_id
DESC"


#define SQLCUR2 "SELECT /*+ USE_NL(cust)  INDEX_DESC (ordr iordr2) */
\
                 c_balance, c_first, c_middle, c_last, \
                 o_id, o_entry_d, o_carrier_id, o_ol_cnt, ordr.rowid \
                 FROM cust, ordr \
                 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 DESC, o_d_id DESC, o_w_id DESC , o_id
DESC"


#define SQLCUR3 "SELECT  /*+ ORDERED USE_NL(ordl) CLUSTER(ordl) */ \
                ol_i_id, ol_supply_w_id, ol_quantity, ol_amount,
ol_delivery_d \
                FROM ordr, ordl \
                WHERE ordr.rowid = :ordr_rowid \
                  AND o_id = ol_o_id AND  ol_d_id = o_d_id AND ol_w_id =
o_w_id"

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


struct ordctx {

   ub2 c_rowid_len[100];
   ub2 ol_supply_w_id_len[NITEMS];
   ub2 ol_i_id_len[NITEMS];
   ub2 ol_quantity_len[NITEMS];
   ub2 ol_amount_len[NITEMS];
   ub2 ol_delivery_d_len[NITEMS];
   ub2 ol_w_id_len;
   ub2 ol_d_id_len;
   ub2 ol_o_id_len;


   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;
   OCIStmt *curo1;
   OCIStmt *curo2;
   OCIStmt *curo3;
   OCIStmt *curo4;
   OCIBind *c_id_bp;
   OCIBind *w_id_bp[4];
   OCIBind *d_id_bp[4];
   OCIBind *c_last_bp[2];
   OCIBind *o_id_bp;
   OCIBind *c_rowid_bp;
   OCIBind *o_rowid_bp;
   OCIDefine *c_rowid_dp;
   OCIDefine *c_last_dp[2];
   OCIDefine *c_id_dp;
   OCIDefine *c_first_dp[2];
   OCIDefine *c_middle_dp[2];
   OCIDefine *c_balance_dp[2];
   OCIDefine *o_rowid_dp[2];
   OCIDefine *o_id_dp[2];
   OCIDefine *o_entry_d_dp[2];
   OCIDefine *o_cr_id_dp[2];
   OCIDefine *o_ol_cnt_dp[2];
   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 *c_rowid_cust;
   OCIRowid *o_rowid;
   int cs;
   int cust_idx;
   int norow;
   int rcount;
   int somerows;
};

typedef struct ordctx ordctx;

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

static ordctx *octx;

static defctx cbctx;


tkvcoinit ()

{

   int i;
   text stmbuf[SQL_BUF_SIZE];
```


Copyright © 2004 NEC Corporation

```
   octx = (ordctx *) malloc (sizeof(ordctx));
   DISCARD memset(octx,(char)0,sizeof(ordctx));
   octx->cs = 1;
   octx->norow = 0;
   octx->somerows = 10;
   /* get the rowid handles */
   OCIERROR(errhp, OCIDescriptorAlloc((dvoid *)tpcenv,(dvoid **)&octx-
>o_rowid,
                        (ub4)OCI_DTYPE_ROWID, (size_t) 0, (dvoid **)0));
   for(i=0;i<100;i++) {
    DISCARD OCIERROR(errhp, OCIDescriptorAlloc(tpcenv,
                    (dvoid**)&octx->c_rowid_ptr[i],
OCI_DTYPE_ROWID,0,(dvoid**)0));
  }


   DISCARD OCIERROR(errhp,
     OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo0,OCI_HTYPE_STMT,0,(dvoid**)0));
   DISCARD OCIERROR(errhp,
     OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo0,OCI_HTYPE_STMT,0,(dvoid**)0));
   DISCARD OCIERROR(errhp,
     OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo1,OCI_HTYPE_STMT,0,(dvoid**)0));
   DISCARD OCIERROR(errhp,
     OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo2,OCI_HTYPE_STMT,0,(dvoid**)0));
   DISCARD OCIERROR(errhp,
     OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo3,OCI_HTYPE_STMT,0,(dvoid**)0));
   DISCARD OCIERROR(errhp,
     OCIHandleAlloc(tpcenv,(dvoid**)&octx-
>curo4,OCI_HTYPE_STMT,0,(dvoid**)0));

/*  c_id = 0, use find customer by lastname. Get an array or rowid's
back*/
   DISCARD sprintf((char *) stmbuf, SQLCUR0);
   DISCARD OCIERROR(errhp,
      OCIStmtPrepare(octx->curo0,errhp,stmbuf,(ub4)strlen((char
*)stmbuf),
                     OCI_NTV_SYNTAX,OCI_DEFAULT));
   DISCARD OCIERROR(errhp,
      OCIAttrSet(octx->curo0,OCI_HTYPE_STMT,&octx->norow,0,
                 OCI_ATTR_PREFETCH_ROWS,errhp));
/* get order/customer info back based on rowid */
   DISCARD sprintf((char *) stmbuf, SQLCUR1);
   DISCARD OCIERROR(errhp,
      OCIStmtPrepare(octx->curo1,errhp,stmbuf,(ub4)strlen((char
*)stmbuf),
                     OCI_NTV_SYNTAX,OCI_DEFAULT));
   DISCARD OCIERROR(errhp,
      OCIAttrSet(octx->curo1,OCI_HTYPE_STMT,&octx->norow,0,
                 OCI_ATTR_PREFETCH_ROWS,errhp));

/*  c_id == 0, use lastname to find customer */
   DISCARD sprintf((char *) stmbuf, SQLCUR2);
   DISCARD OCIERROR(errhp,
      OCIStmtPrepare(octx->curo2,errhp,stmbuf,(ub4)strlen((char
*)stmbuf),
                     OCI_NTV_SYNTAX,OCI_DEFAULT));
   DISCARD OCIERROR(errhp,
      OCIAttrSet(octx->curo2,OCI_HTYPE_STMT,&octx->norow,0,
                 OCI_ATTR_PREFETCH_ROWS,errhp));

   DISCARD sprintf((char *) stmbuf, SQLCUR3);
   DISCARD OCIERROR(errhp,
      OCIStmtPrepare(octx->curo3,errhp,stmbuf,(ub4)strlen((char
*)stmbuf),
                     OCI_NTV_SYNTAX,OCI_DEFAULT));
   DISCARD OCIERROR(errhp,
      OCIAttrSet(octx->curo3,OCI_HTYPE_STMT,&octx->norow,0,
                 OCI_ATTR_PREFETCH_ROWS,errhp));

   DISCARD sprintf((char *) stmbuf, SQLCUR4);
   DISCARD OCIERROR(errhp,
      OCIStmtPrepare(octx->curo4,errhp,stmbuf,(ub4)strlen((char
*)stmbuf),
                     OCI_NTV_SYNTAX,OCI_DEFAULT));
   DISCARD OCIERROR(errhp,
      OCIAttrSet(octx->curo4,OCI_HTYPE_STMT,&octx->norow,0,
                 OCI_ATTR_PREFETCH_ROWS,errhp));

   for (i = 0; i < NITEMS; i++) {

      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_len = sizeof(int);
   octx->ol_d_id_len = sizeof(int);
   octx->ol_o_id_len = sizeof(int);

   /* bind variables */

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

   OCIBND(octx->curo1,octx->c_rowid_bp,errhp,":cust_rowid", &octx-
>c_rowid_cust,
          sizeof( octx->c_rowid_ptr[0]),SQLT_RDD);
   OCIDEF(octx->curo1,octx-
>c_id_dp,errhp,1,ADR(c_id),SIZ(int),SQLT_INT);
   OCIDEF(octx->curo1,octx->c_balance_dp[0],errhp,2,ADR(c_balance),
           SIZ(double),SQLT_BDOUBLE);
   OCIDEF(octx->curo1,octx->c_first_dp[0],errhp,3,c_first,SIZ(c_first)-1,
           SQLT_CHR);
   OCIDEF(octx->curo1,octx->c_middle_dp[0],errhp,4,c_middle,
           SIZ(c_middle)-1,SQLT_AFC);
```


Copyright © 2004 NEC Corporation

```
   OCIDEF(octx->curo1,octx->c_last_dp[0],errhp,5,c_last,SIZ(c_last)-1,
           SQLT_CHR);
   OCIDEF(octx->curo1,octx-
>o_id_dp[0],errhp,6,ADR(o_id),SIZ(int),SQLT_INT);
   OCIDEF(octx->curo1,octx->o_entry_d_dp[0],errhp,7,
           &o_entry_d_base,SIZ(OCIDate),SQLT_ODT);
   OCIDEF(octx->curo1,octx->o_cr_id_dp[0],errhp,8,ADR(o_carrier_id),
           SIZ(int),SQLT_INT);
   OCIDEF(octx->curo1,octx->o_ol_cnt_dp[0],errhp,9,ADR(o_ol_cnt),
           SIZ(int),SQLT_INT);
   OCIDEF(octx->curo1,octx->o_rowid_dp[0],errhp,10,ADR(octx->o_rowid),
          SIZ(OCIRowid*),SQLT_RDD);


/* Bind for third cursor , no-zero customer id */
   OCIBND(octx->curo2,octx->w_id_bp[1],errhp,":w_id",ADR(w_id),
          SIZ(int),SQLT_INT);
   OCIBND(octx->curo2,octx->d_id_bp[1],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_dp[1],errhp,1,ADR(c_balance),
           SIZ(double),SQLT_BDOUBLE);
   OCIDEF(octx->curo2,octx->c_first_dp[1],errhp,2,c_first,SIZ(c_first)-1,
           SQLT_CHR);
   OCIDEF(octx->curo2,octx->c_middle_dp[1],errhp,3,c_middle,
           SIZ(c_middle)-1,SQLT_AFC);
   OCIDEF(octx->curo2,octx->c_last_dp[1],errhp,4,c_last,SIZ(c_last)-1,
          SQLT_CHR);
   OCIDEF(octx->curo2,octx-
>o_id_dp[1],errhp,5,ADR(o_id),SIZ(int),SQLT_INT);
   OCIDEF(octx->curo2,octx->o_entry_d_dp[1],errhp,6, &o_entry_d_base,
          SIZ(OCIDate),SQLT_ODT);
   OCIDEF(octx->curo2, octx->o_cr_id_dp[1],errhp,7,ADR(o_carrier_id),
           SIZ(int), SQLT_INT);
   OCIDEF(octx->curo2,octx->o_ol_cnt_dp[1],errhp,8,ADR(o_ol_cnt),
           SIZ(int),SQLT_INT);
   OCIDEF(octx->curo2,octx->o_rowid_dp[1],errhp,9,ADR(octx->o_rowid),
          SIZ(OCIRowid*),SQLT_RDD);

/* Bind for last cursor */

/*
   OCIBND(octx->curo3,octx->w_id_bp[2],errhp,":w_id",ADR(w_id),
SIZ(int),SQLT_INT);
   OCIBND(octx->curo3,octx->d_id_bp[2],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);
   OCIBND(octx->curo3,octx->c_id_bp,errhp,":c_id",ADR(c_id),
SIZ(int),SQLT_INT);
 */
   OCIBND(octx->curo3,octx->o_rowid_bp,errhp,":ordr_rowid",
          &octx->o_rowid, SIZ(OCIRowid*),SQLT_RDD);

   OCIDFNRA(octx->curo3, octx->ol_i_id_dp, errhp, 1,
ol_i_id,SIZ(int),SQLT_INT,
           NULL,octx->ol_i_id_len, NULL);
   OCIDFNRA(octx->curo3,octx->ol_supply_w_id_dp,errhp,2, ol_supply_w_id,
            SIZ(int),SQLT_INT, NULL,
            octx->ol_supply_w_id_len, NULL);
   OCIDFNRA(octx->curo3, octx->ol_quantity_dp,errhp,3,
ol_quantity,SIZ(float),
           SQLT_BFLOAT, NULL,octx->ol_quantity_len, NULL);
   OCIDFNRA(octx->curo3,octx->ol_amount_dp,errhp,4,ol_amount,  SIZ(float),
           SQLT_BFLOAT,NULL, octx->ol_amount_len, NULL);
   OCIDFNRA(octx->curo3,octx-
>ol_d_base_dp,errhp,5,ol_d_base,SIZ(OCIDate),
           SQLT_ODT, NULL,octx->ol_delivery_d_len,NULL);

   OCIBND(octx->curo4,octx->w_id_bp[3],errhp,":w_id",ADR(w_id),
          SIZ(int),SQLT_INT);
   OCIBND(octx->curo4,octx->d_id_bp[3],errhp,":d_id",ADR(d_id),
          SIZ(int),SQLT_INT);
   OCIBND(octx->curo4,octx->c_last_bp[1],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);


   return (0);

}



tkvco ()

{

   int i;
   int rcount;

#if defined(ISO9)
   int secondread = 0;
   char sdate[30];
   ub4  datelen;
   sysdate(sdate);
   printf("Order Status started at: %s\n", sdate);
#endif

   for (i = 0; i < NITEMS; i++) {
      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;
retry:
   if(bylastname)
   {
     cbctx.reexec = FALSE;
     execstatus=OCIStmtExecute(tpcsvc,octx->curo0,errhp,100,0,
                NULLP(CONST
OCISnapshot),NULLP(OCISnapshot),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))
```


Copyright © 2004 NEC Corporation

```
      {
        DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
        retries++;
        goto retry;
      } else {
        printf("cursor o0. wid = %d. c_id = %d. ", w_id, d_id);
        if (bylastname) printf("lastname = %s.\n", c_last);
        else printf("c_id = %d.\n", c_id);
        return -1;
      }
    }
    if (execstatus == OCI_NO_DATA) /* there are no more rows */
    {
     /* get rowcount, find middle one */
      DISCARD OCIAttrGet(octx->curo0,OCI_HTYPE_STMT,&rcount,NULL,
                         OCI_ATTR_ROW_COUNT,errhp);
      if (rcount <1)
      {
        userlog("ORDERSTATUS  rcount=%d\n",rcount);
        return (-1);
      }
      octx->cust_idx=(rcount)/2 ;
    }
    else
    {
      /* count the number of rows */
      execstatus=OCIStmtExecute(tpcsvc,octx->curo4,errhp,1,0,
                    NULLP(CONST
OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);
      if ((execstatus != OCI_NO_DATA) && (execstatus != OCI_SUCCESS))
      {
        errcode=OCIERROR(errhp, execstatus);
       if ((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR))
       {
         DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
         retries++;
         goto retry;
       } else {
         printf("cursor o4. wid = %d. c_id = %d. ", w_id, d_id);
         if (bylastname) printf("lastname = %s.\n", c_last);
         else printf("c_id = %d.\n", c_id);
         return -1;
       }
      }
      if (octx->rcount+1 < 2*10 )
         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,NULLP(CONST OCISnapshot),
                                   NULLP(OCISnapshot),OCI_DEFAULT);
         /* will get OCI_NO_DATA if <100 found */
         if (cbctx.count > 0)
         {
            userlog ("did not get all rows ");
            return (-1);
         }

         if ((execstatus != OCI_NO_DATA) && (execstatus !=
OCI_SUCCESS))
         {
           errcode=OCIERROR(errhp, execstatus);
           if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR))
           {
             DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
             retries++;
             goto retry;
           } else {
             printf("cursor o0. wid = %d. c_id = %d. ", w_id, d_id);
             if (bylastname) printf("lastname = %s.\n", c_last);
             else printf("c_id = %d.\n", c_id);
             return -1;
           }
         }
         octx->cust_idx=0 ;
      }
    }

    octx->c_rowid_cust = octx->c_rowid_ptr[octx->cust_idx];
    execstatus=OCIStmtExecute(tpcsvc,octx->curo1,errhp,1,0,
                    NULLP(CONST
OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);
    if (execstatus != OCI_SUCCESS)
    {
       errcode=OCIERROR(errhp,execstatus);
       DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
       if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR)
          || (errcode == SNAPSHOT_TOO_OLD))
       {
          retries++;
          goto retry;
       } else {
          printf("cursor o1. wid = %d. c_id = %d. ", w_id, d_id);
          if (bylastname) printf("lastname = %s.\n", c_last);
          else printf("c_id = %d.\n", c_id);
          return -1;
       }
    }
   }
   else
   {
     execstatus=OCIStmtExecute(tpcsvc,octx->curo2,errhp,1,0,
                         NULLP(CONST OCISnapshot),NULLP(OCISnapshot),
                         OCI_DEFAULT);
     if (execstatus != OCI_SUCCESS)
     {
       errcode=OCIERROR(errhp,execstatus);
       DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
       if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR)
          || (errcode == SNAPSHOT_TOO_OLD))
       {
         retries++;
         goto retry;
       }
       else
       {
         printf("cursor o2. wid = %d. c_id = %d. ", w_id, d_id);
         if (bylastname) printf("lastname = %s.\n", c_last);
         else printf("c_id = %d.\n", c_id);
         return -1;
       }
     }
#ifdef ISO9
     sysdate (sdate);
```


Copyright © 2004 NEC Corporation

```
if (!secondread)
      printf ("---------- FIRST READ RESULT (out) %s ----------\n",
sdate);
else
      printf ("---------- SECOND READ RESULT (out) %s ----------\n",
sdate);

      printf ("c_id = %d\n", c_id);
      printf ("c_last = %s\n", c_last);
      printf ("c_first = %s\n", c_first);
      printf ("c_middle = %s\n", c_middle);
      printf ("c_balance = %7.2f\n", (float)c_balance/100);
      printf ("o_id = %d\n", o_id);
      datelen = sizeof(o_entry_d);

OCIERROR(errhp,OCIDateToText(errhp,&o_entry_d_base,(text*)FULLDATE,SIZ(F
ULLDATE),(text*
)0,0,&datelen,o_entry_d));
      printf ("o_entry_d = %s\n", o_entry_d);
      printf ("o_carrier_id = %d\n", o_carrier_id);
      printf ("o_ol_cnt = %d\n", o_ol_cnt);
      printf ("-----------------------------------------------\n\n",
sdate);

if (!secondread) {
      printf ("Sleep before re-read order at: %s\n", sdate);
      sleep (30);
      sysdate (sdate);
      printf ("Wake up and reread at: %s\n", sdate);
      secondread = 1;
      goto retry;
}
#endif /* ISO9 */
    }
    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,
                            NULLP(CONST OCISnapshot),NULLP(OCISnapshot),
                            OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
    if (execstatus != OCI_SUCCESS )
    {
      errcode=OCIERROR(errhp,execstatus);
      DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
      if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR)
           || (errcode == SNAPSHOT_TOO_OLD))
      {
        retries++;
        goto retry;
      }
      else
      {
        printf("cursor o3. wid = %d. c_id = %d. ", w_id, d_id);
        if (bylastname) printf("lastname = %s.\n", c_last);
        else printf("c_id = %d.\n", c_id);
        return -1;
      }

    }
   /* clean up and convert the delivery dates */
    for (i = 0; i < o_ol_cnt; i++)
    {
      ol_del_len[i]=sizeof(ol_delivery_d[i]);
      DISCARD OCIERROR(errhp,OCIDateToText(errhp,&ol_d_base[i],
         (const text*)SHORTDATE,(ub1)strlen(SHORTDATE),(text*)0,0,
          &ol_del_len[i], ol_delivery_d[i]));
/*
        cvtdmy(ol_d_base[i],ol_delivery_d[i]);
*/
    }

   return (0);

}


void tkvcodone ()

{

   if (octx)
      free (octx);

}

/* end of file tkvcord.c */
```

plpay.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: plpay.c 7030100.1 95/07/19 14:44:59 plai Generic<base>
$ Copyr (c) 1994 Oracle";
#endif /* RCSID */

/*=================================================================+
 |        Copyright (c) 1995  Oracle Corp, Redwood Shores, CA       |
 |                    OPEN SYSTEMS PERFORMANCE GROUP                |
 |                         All Rights Reserved                      |
 +=================================================================+
 | FILENAME
 |    plpay.c
 | DESCRIPTION
 |    OCI version (using PL/SQL stored procedure) of
 |    PAYMENT transaction in TPC-C benchmark.
 +=================================================================*/

#include "tpcc.h"

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


#define SQLTXT_INIT "BEGIN inittpcc.init_pay; END;"

struct payctx {
  OCIStmt *curpi;
  OCIStmt *curp0;
  OCIStmt *curp1;
  OCIBind *w_id_bp[2];
  ub2 w_id_len;
```


Copyright © 2004 NEC Corporation

```
OCIBind *d_id_bp[2];
ub2 d_id_len;

OCIBind *c_w_id_bp[2];
ub2 c_w_id_len;

OCIBind *c_d_id_bp[2];
ub2 c_d_id_len;

OCIBind *c_id_bp[2];
ub2 c_id_len;

OCIBind *h_amount_bp[2];
ub2 h_amount_len;

OCIBind *c_last_bp[2];
ub2 c_last_len;

OCIBind *w_street_1_bp[2];
ub2 w_street_1_len;

OCIBind *w_street_2_bp[2];
ub2 w_street_2_len;

OCIBind *w_city_bp[2];
ub2 w_city_len;

OCIBind *w_state_bp[2];
ub2 w_state_len;

OCIBind *w_zip_bp[2];
ub2 w_zip_len;

OCIBind *d_street_1_bp[2];
ub2 d_street_1_len;

OCIBind *d_street_2_bp[2];
ub2 d_street_2_len;

OCIBind *d_city_bp[2];
ub2 d_city_len;

OCIBind *d_state_bp[2];
ub2 d_state_len;

OCIBind *d_zip_bp[2];
ub2 d_zip_len;

OCIBind *c_first_bp[2];
ub2 c_first_len;

OCIBind *c_middle_bp[2];
ub2 c_middle_len;

OCIBind *c_street_1_bp[2];
ub2 c_street_1_len;

OCIBind *c_street_2_bp[2];
ub2 c_street_2_len;

OCIBind *c_city_bp[2];
ub2 c_city_len;

  OCIBind *c_state_bp[2];
  ub2 c_state_len;

  OCIBind *c_zip_bp[2];
  ub2 c_zip_len;

  OCIBind *c_phone_bp[2];
  ub2 c_phone_len;

  OCIBind *c_since_bp[2];
  ub2 c_since_len;

  OCIBind *c_credit_bp[2];
  ub2 c_credit_len;

  OCIBind *c_credit_lim_bp[2];
  ub2 c_credit_lim_len;

  OCIBind *c_discount_bp[2];
  ub2 c_discount_len;

  OCIBind *c_balance_bp[2];
  ub2 c_balance_len;

  OCIBind *c_data_bp[2];
  ub2 c_data_len;

  OCIBind *h_date_bp[2];
  ub2 h_date_len;

  OCIBind *retries_bp[2];
  ub2 retries_len;

  OCIBind *cr_date_bp[2];
  ub2 cr_date_len;

  OCIBind *byln_bp[2];
  ub2 byln_len;
};

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

   DISCARD OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)(&(pctx-
>curp0)),
```


Copyright © 2004 NEC Corporation

```
            OCI_HTYPE_STMT,0,(dvoid**)0));
   DISCARD 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);
   DISCARD OCIERROR(errhp,OCIStmtPrepare(pctx->curpi, errhp, stmbuf,
            strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));
   DISCARD OCIERROR(errhp, OCIStmtExecute(tpcsvc,pctx->curpi,errhp,1,0,
                     NULLP(CONST
OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT));

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

   sqlfile("../blocks/paynz.sql",stmbuf);
   DISCARD OCIERROR(errhp,OCIStmtPrepare(pctx->curp0, errhp, stmbuf,
            strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));

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

   sqlfile("../blocks/payz.sql",stmbuf); /* sqlfile opens
$O/bench/.../blocks/... */
   DISCARD OCIERROR(errhp,OCIStmtPrepare(pctx->curp1, errhp, stmbuf,
            strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));

  pctx->w_id_len = SIZ(w_id);
  pctx->d_id_len = SIZ(d_id);
  pctx->c_w_id_len = SIZ(c_w_id);
  pctx->c_d_id_len = SIZ(c_d_id);
  pctx->c_id_len = 0;
  pctx->h_amount_len = SIZ(h_amount);
  pctx->c_last_len = 0;
  pctx->w_street_1_len = 0;
  pctx->w_street_2_len = 0;
  pctx->w_city_len = 0;
  pctx->w_state_len = 0;
  pctx->w_zip_len = 0;
  pctx->d_street_1_len = 0;
  pctx->d_street_2_len = 0;
  pctx->d_city_len = 0;
  pctx->d_state_len = 0;
  pctx->d_zip_len = 0;
  pctx->c_first_len = 0;
  pctx->c_middle_len = 0;
  pctx->c_street_1_len = 0;
  pctx->c_street_2_len = 0;
  pctx->c_city_len = 0;
  pctx->c_state_len = 0;
  pctx->c_zip_len = 0;
  pctx->c_phone_len = 0;
  pctx->c_since_len = 0;
  pctx->c_credit_len = 0;
  pctx->c_credit_lim_len = 0;
  pctx->c_discount_len = 0;
  pctx->c_balance_len = sizeof(double);
  pctx->c_data_len = 0;
  pctx->h_date_len = 0;
  pctx->retries_len =SIZ(retries) ;
  pctx->cr_date_len = 7;
```

```
   /* bind variables */


   OCIBNDPL(pctx->curp0, pctx->w_id_bp[0],
errhp,":w_id",ADR(w_id),SIZ(int),
            SQLT_INT, NULL);
   OCIBNDPL(pctx->curp0, pctx->d_id_bp[0],
errhp,":d_id",ADR(d_id),SIZ(int),
            SQLT_INT, NULL);
   OCIBND(pctx->curp0, pctx->c_w_id_bp[0],
errhp,":c_w_id",ADR(c_w_id),SIZ(int),
            SQLT_INT);
   OCIBND(pctx->curp0, pctx->c_d_id_bp[0],
errhp,":c_d_id",ADR(c_d_id),SIZ(int),
            SQLT_INT);
   OCIBND(pctx->curp0, pctx->c_id_bp[0],
errhp,":c_id",ADR(c_id),SIZ(int),
            SQLT_INT);
   OCIBNDPL(pctx->curp0, pctx->h_amount_bp[0],
errhp,":h_amount",ADR(h_amount),
            SIZ(float),SQLT_BFLOAT,  &pctx->h_amount_len);
   OCIBNDPL(pctx->curp0, pctx->c_last_bp[0],
errhp,":c_last",c_last,SIZ(c_last),
            SQLT_STR, &pctx->c_last_len);
   OCIBNDPL(pctx->curp0, pctx->w_street_1_bp[0],
errhp,":w_street_1",w_street_1,
            SIZ(w_street_1),SQLT_STR, &pctx->w_street_1_len);
   OCIBNDPL(pctx->curp0, pctx->w_street_2_bp[0],
errhp,":w_street_2",w_street_2,
            SIZ(w_street_2),SQLT_STR, &pctx->w_street_2_len);
   OCIBNDPL(pctx->curp0, pctx->w_city_bp[0],
errhp,":w_city",w_city,SIZ(w_city),
            SQLT_STR, &pctx->w_city_len);
   OCIBNDPL(pctx->curp0, pctx->w_state_bp[0], errhp,":w_state",w_state,
            SIZ(w_state), SQLT_STR, &pctx->w_state_len);
   OCIBNDPL(pctx->curp0, pctx->w_zip_bp[0],
errhp,":w_zip",w_zip,SIZ(w_zip),
            SQLT_STR, &pctx->w_zip_len);
   OCIBNDPL(pctx->curp0, pctx->d_street_1_bp[0],
errhp,":d_street_1",d_street_1,
            SIZ(d_street_1),SQLT_STR, &pctx->d_street_1_len);
   OCIBNDPL(pctx->curp0, pctx->d_street_2_bp[0],
errhp,":d_street_2",d_street_2,
            SIZ(d_street_2),SQLT_STR, &pctx->d_street_2_len);
   OCIBNDPL(pctx->curp0, pctx->d_city_bp[0],
errhp,":d_city",d_city,SIZ(d_city),
            SQLT_STR, &pctx->d_city_len);
   OCIBNDPL(pctx->curp0, pctx->d_state_bp[0], errhp,":d_state",d_state,
            SIZ(d_state), SQLT_STR, &pctx->d_state_len);
   OCIBNDPL(pctx->curp0, pctx->d_zip_bp[0],
errhp,":d_zip",d_zip,SIZ(d_zip),
            SQLT_STR, &pctx->d_zip_len);
   OCIBNDPL(pctx->curp0, pctx->c_first_bp[0], errhp,":c_first",c_first,
            SIZ(c_first), SQLT_STR, &pctx->c_first_len);
   OCIBNDPL(pctx->curp0, pctx->c_middle_bp[0],
errhp,":c_middle",c_middle,2,
            SQLT_AFC, &pctx->c_middle_len);
   OCIBNDPL(pctx->curp0, pctx->c_street_1_bp[0],
errhp,":c_street_1",c_street_1,
            SIZ(c_street_1),SQLT_STR, &pctx->c_street_1_len);
   OCIBNDPL(pctx->curp0, pctx->c_street_2_bp[0],
errhp,":c_street_2",c_street_2,
```


Copyright © 2004 NEC Corporation

```
          SIZ(c_street_2),SQLT_STR, &pctx->c_street_2_len);
   OCIBNDPL(pctx->curp0, pctx->c_city_bp[0],
errhp,":c_city",c_city,SIZ(c_city),
          SQLT_STR, &pctx->c_city_len);
   OCIBNDPL(pctx->curp0, pctx->c_state_bp[0], errhp,":c_state",c_state,
          SIZ(c_state), SQLT_STR, &pctx->c_state_len);
   OCIBNDPL(pctx->curp0, pctx->c_zip_bp[0],
errhp,":c_zip",c_zip,SIZ(c_zip),
          SQLT_STR,&pctx->c_zip_len);
   OCIBNDPL(pctx->curp0, pctx->c_phone_bp[0], errhp,":c_phone",c_phone,
          SIZ(c_phone), SQLT_STR, &pctx->c_phone_len);
   OCIBNDPL(pctx->curp0, pctx->c_since_bp[0], errhp,":c_since",&c_since,
          SIZ(OCIDate), SQLT_ODT, &pctx->c_since_len);
   OCIBNDPL(pctx->curp0, pctx->c_credit_bp[0],
errhp,":c_credit",c_credit,
          SIZ(c_credit),SQLT_CHR, &pctx->c_credit_len);
   OCIBNDPL(pctx->curp0, pctx->c_credit_lim_bp[0], errhp,":c_credit_lim",
          ADR(c_credit_lim),SIZ(int), SQLT_INT, &pctx-
>c_credit_lim_len);
   OCIBNDPL(pctx->curp0, pctx->c_discount_bp[0], errhp,":c_discount",
          ADR(c_discount),SIZ(c_discount), SQLT_FLT, &pctx-
>c_discount_len);
   OCIBNDPL(pctx->curp0, pctx->c_balance_bp[0], errhp,":c_balance",
          ADR(c_balance), SIZ(double),SQLT_BDOUBLE, &pctx-
>c_balance_len);
   OCIBNDPL(pctx->curp0, pctx->c_data_bp[0],
errhp,":c_data",c_data,SIZ(c_data),
          SQLT_STR, &pctx->c_data_len);
/*
   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);
*/
   OCIBNDPL(pctx->curp0, pctx->retries_bp[0],
errhp,":retry",ADR(retries),
          SIZ(int), SQLT_INT, &pctx->retries_len);
   OCIBNDPL(pctx->curp0, pctx->cr_date_bp[0],
errhp,":cr_date",ADR(cr_date),
          SIZ(OCIDate),SQLT_ODT, &pctx->cr_date_len);

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


   OCIBNDPL(pctx->curp1, pctx->w_id_bp[1],
errhp,":w_id",ADR(w_id),SIZ(int),
          SQLT_INT, &pctx->w_id_len);
   OCIBNDPL(pctx->curp1, pctx->d_id_bp[1],
errhp,":d_id",ADR(d_id),SIZ(int),
          SQLT_INT, &pctx->d_id_len);
   OCIBND(pctx->curp1, pctx->c_w_id_bp[1],
errhp,":c_w_id",ADR(c_w_id),SIZ(int),
          SQLT_INT);
   OCIBND(pctx->curp1, pctx->c_d_id_bp[1],
errhp,":c_d_id",ADR(c_d_id),SIZ(int),
          SQLT_INT);
   OCIBNDPL(pctx->curp1, pctx->c_id_bp[1],
errhp,":c_id",ADR(c_id),SIZ(int),
          SQLT_INT, &pctx->c_id_len);
   OCIBNDPL(pctx->curp1, pctx->h_amount_bp[1],
errhp,":h_amount",ADR(h_amount),
          SIZ(float),SQLT_BFLOAT, &pctx->h_amount_len);
```

```
   OCIBND(pctx->curp1, pctx->c_last_bp[1],
errhp,":c_last",c_last,SIZ(c_last),
          SQLT_STR);
   OCIBNDPL(pctx->curp1, pctx->w_street_1_bp[1],
errhp,":w_street_1",w_street_1,
          SIZ(w_street_1),SQLT_STR, &pctx->w_street_1_len);
   OCIBNDPL(pctx->curp1, pctx->w_street_2_bp[1],
errhp,":w_street_2",w_street_2,
          SIZ(w_street_2),SQLT_STR, &pctx->w_street_2_len);
   OCIBNDPL(pctx->curp1, pctx->w_city_bp[1],
errhp,":w_city",w_city,SIZ(w_city),
          SQLT_STR, &pctx->w_city_len);
   OCIBNDPL(pctx->curp1, pctx->w_state_bp[1], errhp,":w_state",w_state,
          SIZ(w_state), SQLT_STR, &pctx->w_state_len);
   OCIBNDPL(pctx->curp1, pctx->w_zip_bp[1],
errhp,":w_zip",w_zip,SIZ(w_zip),
          SQLT_STR, &pctx->w_zip_len);
   OCIBNDPL(pctx->curp1, pctx->d_street_1_bp[1],
errhp,":d_street_1",d_street_1,
          SIZ(d_street_1),SQLT_STR, &pctx->d_street_1_len);
   OCIBNDPL(pctx->curp1, pctx->d_street_2_bp[1],
errhp,":d_street_2",d_street_2,
          SIZ(d_street_2),SQLT_STR, &pctx->d_street_2_len);
   OCIBNDPL(pctx->curp1, pctx->d_city_bp[1],
errhp,":d_city",d_city,SIZ(d_city),
          SQLT_STR, &pctx->d_city_len);
   OCIBNDPL(pctx->curp1, pctx->d_state_bp[1], errhp,":d_state",d_state,
          SIZ(d_state), SQLT_STR, &pctx->d_state_len);
   OCIBNDPL(pctx->curp1, pctx->d_zip_bp[1],
errhp,":d_zip",d_zip,SIZ(d_zip),
          SQLT_STR, &pctx->d_zip_len);
   OCIBNDPL(pctx->curp1, pctx->c_first_bp[1], errhp,":c_first",c_first,
          SIZ(c_first), SQLT_STR, &pctx->c_first_len);
   OCIBNDPL(pctx->curp1, pctx->c_middle_bp[1],
errhp,":c_middle",c_middle,2,
          SQLT_AFC, &pctx->c_middle_len);

   OCIBNDPL(pctx->curp1, pctx->c_street_1_bp[1],
errhp,":c_street_1",c_street_1,
          SIZ(c_street_1),SQLT_STR, &pctx->c_street_1_len);
   OCIBNDPL(pctx->curp1, pctx->c_street_2_bp[1],
errhp,":c_street_2",c_street_2,
          SIZ(c_street_2),SQLT_STR, &pctx->c_street_2_len);
   OCIBNDPL(pctx->curp1, pctx->c_city_bp[1], errhp,":c_city",c_city,
          SIZ(c_city),SQLT_STR, &pctx->c_city_len);
   OCIBNDPL(pctx->curp1, pctx->c_state_bp[1], errhp,":c_state",c_state,
          SIZ(c_state), SQLT_STR, &pctx->c_state_len);
   OCIBNDPL(pctx->curp1, pctx->c_zip_bp[1],
errhp,":c_zip",c_zip,SIZ(c_zip),
          SQLT_STR, &pctx->c_zip_len);
   OCIBNDPL(pctx->curp1, pctx->c_phone_bp[1], errhp,":c_phone",c_phone,
          SIZ(c_phone), SQLT_STR, &pctx->c_phone_len);
   OCIBNDPL(pctx->curp1, pctx->c_since_bp[1], errhp,":c_since",&c_since,
          SIZ(OCIDate), SQLT_ODT, &pctx->c_since_len);
   OCIBNDPL(pctx->curp1, pctx->c_credit_bp[1],
errhp,":c_credit",c_credit,
          SIZ(c_credit),SQLT_CHR, &pctx->c_credit_len);
   OCIBNDPL(pctx->curp1, pctx->c_credit_lim_bp[1], errhp,":c_credit_lim",
          ADR(c_credit_lim),SIZ(int), SQLT_INT, &pctx-
>c_credit_lim_len);
   OCIBNDPL(pctx->curp1, pctx->c_discount_bp[1], errhp,":c_discount",
          ADR(c_discount),SIZ(c_discount), SQLT_FLT, &pctx-
>c_discount_len);
```


Copyright © 2004 NEC Corporation

```
   OCIBNDPL(pctx->curp1, pctx->c_balance_bp[1], errhp,":c_balance",
           ADR(c_balance), SIZ(double),SQLT_BDOUBLE, &pctx-
>c_balance_len);
   OCIBNDPL(pctx->curp1, pctx->c_data_bp[1],
errhp,":c_data",c_data,SIZ(c_data),
           SQLT_STR, &pctx->c_data_len);
/*
   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);
*/
   OCIBNDPL(pctx->curp1, pctx->retries_bp[1],
errhp,":retry",ADR(retries),
           SIZ(int), SQLT_INT, &pctx->retries_len);
   OCIBNDPL(pctx->curp1, pctx->cr_date_bp[1],
errhp,":cr_date",ADR(cr_date),
           SIZ(OCIDate),SQLT_ODT, &pctx->cr_date_len);

   return (0);

}



tkvcp ()

{

retry:

  pctx->w_id_len = SIZ(w_id);
  pctx->d_id_len = SIZ(d_id);
  pctx->c_w_id_len = 0;
  pctx->c_d_id_len = 0;
  pctx->c_id_len = 0;
  pctx->h_amount_len = SIZ(h_amount);
  pctx->c_last_len = SIZ(c_last);
  pctx->w_street_1_len = 0;
  pctx->w_street_2_len = 0;
  pctx->w_city_len = 0;
  pctx->w_state_len = 0;
  pctx->w_zip_len = 0;
  pctx->d_street_1_len = 0;
  pctx->d_street_2_len = 0;
  pctx->d_city_len = 0;
  pctx->d_state_len = 0;
  pctx->d_zip_len = 0;
  pctx->c_first_len = 0;
  pctx->c_middle_len = 0;
  pctx->c_street_1_len = 0;
  pctx->c_street_2_len = 0;
  pctx->c_city_len = 0;
  pctx->c_state_len = 0;
  pctx->c_zip_len = 0;
  pctx->c_phone_len = 0;
  pctx->c_since_len = 0;
  pctx->c_credit_len = 0;
  pctx->c_credit_lim_len = 0;
  pctx->c_discount_len = 0;
  pctx->c_balance_len = sizeof(double);
  pctx->c_data_len = 0;
  pctx->h_date_len = 0;
  pctx->retries_len = SIZ(retries);
  pctx->cr_date_len = 7;

  if(bylastname) {
    execstatus=OCIStmtExecute(tpcsvc,pctx->curp1,errhp,1,0,
               NULLP(CONST OCISnapshot),NULLP(OCISnapshot),
               OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS);
  } else {
    execstatus=OCIStmtExecute(tpcsvc,pctx->curp0,errhp,1,0,
               NULLP(CONST OCISnapshot),NULLP(OCISnapshot),
               OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS);
  }


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



void tkvcpdone ()

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

plsto.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: plsto.c 7010000.3 95/02/14 12:48:03 plai Generic<base>
$ Copyr (c) 1994 Oracle";
#endif /* RCSID */

/*=====================================================================+
 |        Copyright (c) 1994  Oracle Corp, Redwood Shores, CA         |
 |                    OPEN SYSTEMS PERFORMANCE GROUP                  |
 |                         All Rights Reserved                        |
 +=====================================================================+
 | FILENAME
 |    plsto.c
 | DESCRIPTION
 |    OCI version of STOCK LEVEL transaction in TPC-C benchmark.
 +=====================================================================*/
```


Copyright © 2004 NEC Corporation

```
#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 ordl, stok, dist \
            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) \
                 order by ol_o_id desc"
#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(float),SQLT_BFLOAT);
#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 -1;
      }
     }

  return (0);
}


void tkvcsdone ()

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

tpccpl.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: tpccpl.c 7030100.2 96/04/02 17:51:34 plai Generic<base> $ Copyr (c)
1994 Oracle";
#endif /* RCSID */
```


Copyright © 2004 NEC Corporation

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

+==============================================================
======+
 | FILENAME
 |   tpccpl.c
 | DESCRIPTION
 |   TPC-C transactions in PL/SQL.

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

#include <stdio.h>
#include <time.h>
#include <sys/timeb.h>
#include "tpcc.h"
#ifdef TUX
#include <userlog.h>
#else
#include <stdarg.h>
#endif

#define SQLTXT "alter session set isolation_level = serializable"
#define SQLTXTTRC "alter session set sql_trace = true"
#define SQLTXTTIM "alter session set timed_statistics = true"

FILE *lfp;
FILE *fopen ();
#ifdef ORA_NT
#undef boolean
//#include "dpbcore.h"
//#define gettime dpbtimef
#else
extern double gettime ();
#endif
int proc_no = 0;
static int logon = 0;
static int new_init = 0;
static int pay_init = 0;
static int ord_init = 0;
static int del_init_oci = 0;
static int del_init_plsql = 0;
static int sto_init = 0;
static int res_init = 0;

int execstatus;
int errcode;

struct timeb  timebuffer;//change

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;
float threshold;
int low_stock;

/* for delivery transaction */

int del_o_id[10];
int retries;

/* for order-status transaction */

int bylastname;
char c_last[17];
char c_first[17];
char c_middle[3];
double c_balance;
int o_id;
text o_entry_d[20];
ub4  datelen;
int o_carrier_id;
```


Copyright © 2004 NEC Corporation

```
int o_ol_cnt;
int ol_supply_w_id[15];
int ol_i_id[15];
float ol_quantity[15];
float ol_amount[15];
ub4  ol_del_len[15];
text ol_delivery_d[15][11];
/* xnie - begin */
OCIRowid *o_rowid;
/* xnie - end */

/* for payment transaction */

int c_w_id;
int c_d_id;
float 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];
float nol_quantity[15];
int nol_quanti10[15];
int nol_quanti91[15];
int nol_ytdqty[15];
float nol_amount[15];
int o_all_local;
float w_tax;
float d_tax;
float total_amount;
char i_name[15][25];
float s_quantity[15];
char brand_gen[15];
float i_price[15];
char brand_generic[15][1];
int status;
int tracelevel = 0;

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

#ifndef AVOID_DEADLOCK
int indx[NITEMS], ordl_cnt;
void swap(struct newstruct *str, int i, int j);
void q_sort(int *arr, struct newstruct *str, int left, int right);
#endif

/*
extern char oracle_home[256];
*/

/* NewOrder Binding stuff */


//for debug

void debuglog(char *filename, char *format, ...)
{
        FILE *fp;
        va_list args;
        char buf[4096];
```


Copyright © 2004 NEC Corporation

```
        int len;

        struct tm* newtime;
    time_t now;

        va_start( args, format );
        //_strtime( buf );

    time(&now);
    newtime=localtime(&now);
    strcpy(buf, asctime(newtime));

        strcat( buf," ");
        len = strlen( buf );
        (void)vsnprintf( buf+ len, sizeof( buf) - len - 1, format, args);
        buf[sizeof( buf )- 1]= '\0';
        va_end( args );

        if (fp = fopen(filename, "a"))
          {
                        fprintf(fp,"%s.\n", buf);
                        fflush(fp);
                        fclose(fp);
        }
}

#ifndef TUX
void userlog (char* fmtp, ...)
{
  va_list va;
  va_start(va,fmtp);
  vfprintf(stderr,fmtp,va);
  va_end(va);
}
#endif

/* vmm313 void ocierror(fname, lineno, errhp, status)  */
int ocierror(fname, lineno, errhp, status)
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:
          debuglog("/tmp/OCIErrorLog","2 Error - OCI_SUCCESS_WITH_INFO\n");
    //fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    //fprintf(stderr,"Error - OCI_SUCCESS_WITH_INFO\n");
    lstat = OCIErrorGet (errhp, recno++, (text *) NULL, &errcode, errbuf,
                  (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
          debuglog("/tmp/OCIErrorLog","2 Error - %s\n", errbuf);
    //fprintf(stderr,"Error - %s\n", errbuf);
    break;
  case OCI_NEED_DATA:
          debuglog("/tmp/OCIErrorLog","3 Module %s Line %d\n", fname, lineno);
    debuglog("/tmp/OCIErrorLog","3 Error - OCI_NEED_DATA\n");
    //fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    //fprintf(stderr,"Error - OCI_NEED_DATA\n");
    return (IRRECERR);
  case OCI_NO_DATA:
          debuglog("/tmp/OCIErrorLog","4 Module %s Line %d\n", fname, lineno);
    debuglog("/tmp/OCIErrorLog","4 Error - OCI_NO_DATA\n");
    //fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    //fprintf(stderr,"Error - OCI_NO_DATA\n");
    return (IRRECERR);
  case OCI_ERROR:
    lstat = OCIErrorGet (errhp, (ub4) 1,
                  (text *) NULL, &errcode, errbuf,
                        (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
    if (errcode == NOT_SERIALIZABLE)
          {
                  debuglog("/tmp/OCIErrorLog","5 ERROR");
                  return (errcode);
          }
    if (errcode == SNAPSHOT_TOO_OLD)
          {
                  debuglog("/tmp/OCIErrorLog","6 snapshot_too_old");
                  return (errcode);
          }
    while (lstat != OCI_NO_DATA)
```


Copyright © 2004 NEC Corporation

```
    {
            debuglog("/tmp/OCIErrorLog","7 Module %s Line %d\n", fname, lineno);
     debuglog("/tmp/OCIErrorLog","7 Error - %s\n", errbuf);
     //fprintf(stderr,"Module %s Line %d\n", fname, lineno);
     //fprintf(stderr,"Error - %s\n", errbuf);
     lstat = OCIErrorGet (errhp, recno++, (text *) NULL, &errcode, errbuf,
                    (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
    }
    return (errcode);
/* vmm313    TPCexit(1); */
/* vmm313    exit(1); */
  case OCI_INVALID_HANDLE:
            debuglog("/tmp/OCIErrorLog","8 Module %s Line %d\n", fname, lineno);
    debuglog("/tmp/OCIErrorLog","8 Error - OCI_INVALID_HANDLE\n");
    //fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    //fprintf(stderr,"Error - OCI_INVALID_HANDLE\n");
    TPCexit(1);
    exit(-1);
  case OCI_STILL_EXECUTING:
            debuglog("/tmp/OCIErrorLog","9 Module %s Line %d\n", fname, lineno);
    debuglog("/tmp/OCIErrorLog","9 Error - OCI_STILL_EXECUTE\n");
    //fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    //fprintf(stderr,"Error - OCI_STILL_EXECUTE\n");
    return (IRRECERR);
  case OCI_CONTINUE:
            debuglog("/tmp/OCIErrorLog","10 Module %s Line %d\n", fname, lineno);
    debuglog("/tmp/OCIErrorLog","10 Error - OCI_CONTINUE\n");
    //fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    //fprintf(stderr,"Error - OCI_CONTINUE\n");
    return (IRRECERR);
  default:
            debuglog("/tmp/OCIErrorLog","11 Module %s Line %d\n", fname, lineno);
    debuglog("/tmp/OCIErrorLog","11 Status - %s\n", status);
    //fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    //fprintf(stderr,"Status - %s\n", status);
    return (IRRECERR);
  }
  return (RECOVERR);
}


FILE  *vopen(fnam,mode)
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){
        fprintf(stderr," fopen on %s failed %d\n",fnam,fd);
        exit(-1);
    }
    return(fd);
}

int sqlfile(fnam,linebuf)
char   *fnam;
text   *linebuf;
{
FILE *fd;
int nulpt = 0;
char realfile[512];

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

/*
    sprintf(realfile,"%s/bench/tpc/tpcc/blocks/%s",oracle_home,fnam);
*/
    sprintf(realfile,"%s",fnam);
    fd = vopen(realfile,"r");
    while (fgets((char *)linebuf+nulpt, SQL_BUF_SIZE,fd))
    {
        nulpt = strlen((char *)linebuf);
    }
    return(nulpt);
}

#ifdef NOT
void vgetdate (unsigned char *oradt)
{
 struct tm *loctime;
 time_t  int_time;
```


Copyright © 2004 NEC Corporation

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

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

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

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

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

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


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

}
void cvtdmy (unsigned char *oradt, char *outdate)
{

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


      int day,month,year;

      memcpy(&Date,oradt,7);

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

      return;
}

void cvtdmyhms (unsigned char *oradt, char *outdate)
{

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


Copyright © 2004 NEC Corporation

```
    int day,month,year;
    int hour,min,sec;

    memcpy(&Date,oradt,7);

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

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

    return;
}
#endif


void TPCexit (void)

{

  if (new_init) {
    tkvcndone();
    new_init = 0;
  }
  if (pay_init) {
    tkvcpdone();
    pay_init = 0;
  }
  if (ord_init) {
    tkvcodone();
    ord_init = 0;
  }
  if (del_init_oci) {
    tkvcddone(0);
    del_init_oci = 0;
  }
  if (del_init_plsql) {
    tkvcddone(1);
    del_init_plsql = 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);

  if (lfp) {
          //fflush (lfp);
    fclose (lfp);
    lfp = NULL;
  }

}


TPCinit (id, uid, pwd)

int id;
char *uid;
char *pwd;

{

  char filename[40];
  text stmbuf[100];

  proc_no = id;
  sprintf (filename, "tpcc_%d.del", proc_no);
  if ((lfp = fopen (filename, "w")) == NULL) {
#ifdef TUX
    userlog ("Error in TPC-C server %d: Failed to open %s\n",
          proc_no, filename);
#else
    fprintf (stderr, "Error in TPC-C server %d: Failed to open %s\n",
          proc_no, filename);
#endif
```


Copyright © 2004 NEC Corporation

```
    return (-1);
  }


  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);
  OCIServerAttach(tpcsrv, errhp, (text *)0,0,OCI_DEFAULT); //change
  OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&tpcsvc, OCI_HTYPE_SVCCTX, 0 ,
(dvoid **)0);
  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);
  OCIERROR(errhp,OCIStmtExecute(tpcsvc, curi, errhp,1,0,0,0,OCI_DEFAULT));
  OCIHandleFree(curi, OCI_HTYPE_STMT);


/*
This is done in cvdrv.c
  if (tracelevel == 2) {
    OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);
    memset(stmbuf,0,100);
    sprintf ((char *) stmbuf, SQLTXTTRC);
    OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf),
                          OCI_NTV_SYNTAX, OCI_DEFAULT);
    OCIERROR(errhp, OCIStmtExecute(tpcsvc, curi, errhp,1,0,0,0,OCI_DEFAULT));
    OCIHandleFree((dvoid *)curi, OCI_HTYPE_STMT);
  }
*/
  if (tracelevel == 3) {
    OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);

    memset(stmbuf,0,100);
    sprintf ((char *) stmbuf, SQLTXTTIM);
    OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf),
                          OCI_NTV_SYNTAX, OCI_DEFAULT);
    OCIERROR(errhp, OCIStmtExecute(tpcsvc, curi, errhp,1,0,0,0,OCI_DEFAULT));
    OCIHandleFree((dvoid *)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 (tkvcoinit ()) {   /* order status */
    TPCexit ();

    return (-1);
  }
  else
    ord_init = 1;



  if (tkvcdinit (0)) {   /* delivery */
    TPCexit ();
```


Copyright © 2004 NEC Corporation

```
    return (-1);
  }
  else
    del_init_oci = 1;


  if (tkvcdinit (1)) {   /* delivery */
    TPCexit ();

    return (-1);
  }
  else
    del_init_plsql = 1;


  if (tkvcsinit ()) {   /* stock level */
    TPCexit ();

    return (-1);
  }
  else
    sto_init = 1;


  return (0);

}



TPCnew (str)

struct newstruct *str;

{

  int i;

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

#ifndef AVOID_DEADLOCK

  for (i = NITEMS; i > 0; i--) {
    if (nol_i_id[i-1] > 0) {
      ordl_cnt = i;
      break;
    }
  }

  for (i = 0; i < NITEMS; i++) indx[i] = i;

  q_sort(nol_i_id, str, 0, ordl_cnt-1);

#endif

/*
  vgetdate(cr_date); */

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

  if (str->newout.terror = tkvcn ()) {
    if (str->newout.terror != RECOVERR)
      str->newout.terror = IRRECERR;
    return (-1);
  }

  /* fill in date for o_entry_d from time in beginning of txn*/
/*
  cvtdmyhms(cr_date,o_entry_d);
*/
   datelen = sizeof(o_entry_d);
   OCIERROR(errhp,
      OCIDateToText(errhp,&cr_date,(text*)FULLDATE,SIZ(FULLDATE),(text*)0,0,
                &datelen,o_entry_d));

  str->newout.terror = NOERR;
  str->newout.o_id = o_id;
  str->newout.o_ol_cnt = o_ol_cnt;
  strncpy (str->newout.c_last, c_last, 17);
```


Copyright © 2004 NEC Corporation

```
  strncpy (str->newout.c_credit, c_credit, 3);
  str->newout.c_discount = c_discount;
  str->newout.w_tax = (float)(w_tax);
  str->newout.d_tax = (float)(d_tax);
  strncpy (str->newout.o_entry_d, (char*)o_entry_d, 20);
  str->newout.total_amount = total_amount;
  for (i = 0; i < o_ol_cnt; i++) {
            str->newout.ol_i_id[i] = nol_i_id[i];//add
            str->newout.ol_supply_w_id[i] = nol_supply_w_id[i];//add
    str->newout.ol_quantity[i] = nol_quantity[i];//add
            strncpy (str->newout.i_name[i], i_name[i], 25);
    str->newout.s_quantity[i] = (int) s_quantity[i];
    str->newout.brand_generic[i] = brand_generic[i][0];
    str->newout.i_price[i] = i_price[i]/100;
    str->newout.ol_amount[i] = nol_amount[i]/100;
  }

#ifndef AVOID_DEADLOCK
   //q_sort(indx, str, 0, ordl_cnt-1);
#endif

  if (status){
    strcpy (str->newout.status, "Item number is not valid");
            return(0); //add for abort transaction
            }
  else
    str->newout.status[0] = '\0';
  str->newout.retry = retries;
#if defined(TOP)  || defined(TUX)    /* changed mjb 17 feb for tuxedo */
  return(1);
#else
  return (0);
#endif

}



TPCpay (str)

struct paystruct *str;

{
```

```
  w_id = str->payin.w_id;
  d_id = str->payin.d_id;
  c_w_id = str->payin.c_w_id;
  c_d_id = str->payin.c_d_id;
  h_amount = str->payin.h_amount;
  bylastname = str->payin.bylastname;

/*
  vgetdate(cr_date);  */
  OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));

  if (bylastname) {
    c_id = 0;
    strncpy (c_last, str->payin.c_last, 17);
  }
  else {
    c_id = str->payin.c_id;
    strcpy (c_last, " ");
  }
  retries = 0;

  if (str->payout.terror = tkvcp ()) {
    if (str->payout.terror != RECOVERR)
      str->payout.terror = IRRECERR;
    return (-1);
  }

/*
  cvtdmyhms(cr_date,h_date);
*/
      hlen=SIZ(h_date);
      OCIERROR(errhp,OCIDateToText(errhp,&cr_date,
            (text*)FULLDATE,strlen(FULLDATE),(text*)0,0,&hlen,h_date));

/*
  cvtdmy(c_since,c_since_d);
*/
      sincelen=SIZ(c_since_d);
      OCIERROR(errhp,OCIDateToText(errhp,&c_since,
        (text*)SHORTDATE,strlen(SHORTDATE),(text*)0,0,&sincelen,c_since_d));



  str->payout.terror = NOERR;
```


Copyright © 2004 NEC Corporation

```
  strncpy (str->payout.w_street_1, w_street_1, 21);
  strncpy (str->payout.w_street_2, w_street_2, 21);
  strncpy (str->payout.w_city, w_city, 21);
  strncpy (str->payout.w_state, w_state, 3);
  strncpy (str->payout.w_zip, w_zip, 10);
  strncpy (str->payout.d_street_1, d_street_1, 21);
  strncpy (str->payout.d_street_2, d_street_2, 21);
  strncpy (str->payout.d_city, d_city, 21);
  strncpy (str->payout.d_state, d_state, 3);
  strncpy (str->payout.d_zip, d_zip, 10);
  str->payout.c_id = c_id;
  strncpy (str->payout.c_first, c_first, 17);
  strncpy (str->payout.c_middle, c_middle, 3);
  strncpy (str->payout.c_last, c_last, 17);
  strncpy (str->payout.c_street_1, c_street_1, 21);
  strncpy (str->payout.c_street_2, c_street_2, 21);
  strncpy (str->payout.c_city, c_city, 21);
  strncpy (str->payout.c_state, c_state, 3);
  strncpy (str->payout.c_zip, c_zip, 10);
  strncpy (str->payout.c_phone, c_phone, 17);
  strncpy (str->payout.c_since, (char*)c_since_d, 11);
  strncpy (str->payout.c_credit, c_credit, 3);
  str->payout.c_credit_lim = (float)(c_credit_lim)/100;
  str->payout.c_discount = c_discount;
  str->payout.c_balance = (float)(c_balance)/100;
  strncpy (str->payout.c_data, c_data, 201);
  strncpy (str->payout.h_date, (char*)h_date, 20);
  str->payout.retry = retries;
#if defined(TOP)  || defined(TUX)   /* changed mjb 17 Feb */
  return(1);
#else
  return (0);
#endif

}



TPCord (str)

struct ordstruct *str;

{
  int i;
  w_id = str->ordin.w_id;
  d_id = str->ordin.d_id;
  bylastname = str->ordin.bylastname;
  if (bylastname) {
    c_id = 0;
    strncpy (c_last, str->ordin.c_last, 17);
  }
  else {
    c_id = str->ordin.c_id;
    strcpy (c_last, " ");
  }
  retries = 0;

  if (str->ordout.terror = tkvco ()) {
    if (str->ordout.terror != RECOVERR)
      str->ordout.terror = IRRECERR;
    return (-1);
  }

  datelen = sizeof(o_entry_d);
  OCIERROR(errhp,

OCIDateToText(errhp,&o_entry_d_base,(text*)FULLDATE,SIZ(FULLDATE),(text*
)0,0,
                &datelen,o_entry_d));

  str->ordout.terror = NOERR;
  str->ordout.c_id = c_id;
  strncpy (str->ordout.c_last, c_last, 17);
  strncpy (str->ordout.c_first, c_first, 17);
  strncpy (str->ordout.c_middle, c_middle, 3);
  str->ordout.c_balance = c_balance/100;
  str->ordout.o_id = o_id;
  strncpy (str->ordout.o_entry_d, (char*)o_entry_d, 20);
  if ( o_carrier_id == 11 )
     str->ordout.o_carrier_id = 0;
  else
     str->ordout.o_carrier_id = o_carrier_id;
  str->ordout.o_ol_cnt = o_ol_cnt;
  for (i = 0; i < o_ol_cnt; i++) {
    ol_delivery_d[i][10] = '\0';
    if ( !strcmp((char*)ol_delivery_d[i],"15-09-1911") )
          strncpy((char*)ol_delivery_d[i],"NOT DELIVR",10);
```


Copyright © 2004 NEC Corporation

```
      str->ordout.ol_supply_w_id[i] = ol_supply_w_id[i];
      str->ordout.ol_i_id[i] = ol_i_id[i];
      str->ordout.ol_quantity[i] = (int) ol_quantity[i];
      str->ordout.ol_amount[i] = ol_amount[i]/100;
      strncpy (str->ordout.ol_delivery_d[i], (char*)ol_delivery_d[i], 11);
    }
    str->ordout.retry = retries;
#if defined(TOP)  || defined(TUX)
    return(1);
#else
    return (0);
#endif

}



TPCdel (str)

struct delstruct *str;

{

    double tr_end;
    int i;

    w_id = str->delin.w_id;
    o_carrier_id = str->delin.o_carrier_id;
    retries = 0;
/*
    vgetdate(cr_date);  */
    OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));

    str->delin.plsqlflag = 1; // 1:PLSQLDEL, 0:OCIDEL
    //We suspect PLSQLDEL has bugs.

    if (str->delout.terror = tkvcd (str->delin.plsqlflag)) {
      if(str->delout.terror == DEL_ERROR)
        return DEL_ERROR;
      if (str->delout.terror != RECOVERR)
         str->delout.terror = IRRECERR;
      return (-1);
    }
    //tr_end = gettime ();//error
    ftime(&timebuffer);
    tr_end = timebuffer.time + timebuffer.millitm/1000.0;
/* old type
    fprintf (lfp, "%d %d %f %f %d %d", str->delin.in_timing_int,
            (tr_end - str->delin.qtime) <= DELRT ? 1 : 0,
            str->delin.qtime, tr_end, w_id, o_carrier_id);
    for (i = 0; i < 10; i++) {
      fprintf (lfp, " %d %d", i + 1, del_o_id[i]);
      if (del_o_id[i] <= 0) {
#ifdef TUX
        userlog ("DELIVERY: no new order for w_id: %d, d_id %d\n",
                w_id, i + 1);
#else
        fprintf (stderr, "DELIVERY: no new order for w_id: %d, d_id %d\n",
                w_id, i + 1);
#endif
      }
    }
    fprintf (lfp, " %d\n", retries);
    str->delout.terror = NOERR;
    str->delout.retry = retries;

*/
fprintf (lfp, "%f %f %d %d",str->delin.qtime, tr_end, w_id, o_carrier_id);
    for (i = 0; i < 10; i++) {
      fprintf (lfp, " %d",del_o_id[i]);
      if (del_o_id[i] <= 0) {
#ifdef TUX
        userlog ("DELIVERY: no new order for w_id: %d, d_id %d\n",
                w_id, i + 1);
#else
        fprintf (stderr, "DELIVERY: no new order for w_id: %d, d_id %d\n",
                w_id, i + 1);
#endif
      }
    }
    fprintf (lfp, "\n");
    str->delout.terror = NOERR;
    str->delout.retry = retries;


#if defined(TOP) || defined(TUX)   /* changed mjb 17 feb */
    return(1);
```


Copyright © 2004 NEC Corporation

```
#else
  return (0);
#endif

}



TPCsto (str)

struct stostruct *str;

{

  w_id = str->stoin.w_id;
  d_id = str->stoin.d_id;
  threshold = (float) str->stoin.threshold;
  retries = 0;

  if (str->stoout.terror = tkvcs ()) {
    if (str->stoout.terror != RECOVERR)
      str->stoout.terror = IRRECERR;
    return (-1);
  }

  str->stoout.terror = NOERR;
  str->stoout.low_stock = low_stock;
  str->stoout.retry = retries;
#if defined(TOP) || defined(TUX) /* changed mjb 17 feb */
  return(1);
#else
  return (0);
#endif

}

#ifndef AVOID_DEADLOCK

void q_sort(int *arr,struct newstruct *str,int left, int right)
{
 int i, last;

 if(left >= right)
  return;
 swap(str,left,(left+right)/2);
 last = left;
 for(i=left+1;i<=right;i++)
  if(arr[i] < arr[left])
    swap(str,last,i);
 swap(str,left,last);
 q_sort(arr,str,left,last-1);
 q_sort(arr,str,last+1,right);
}

void swap(struct newstruct *str, int i, int j)
{
 float temp_float;
 int temp;
 char tmpstr[25];
 char tmpch;

 temp = indx[i];
 indx[i] = indx[j];
 indx[j] = temp;

 temp = nol_i_id[i];
 nol_i_id[i] = nol_i_id[j];
 nol_i_id[j] = temp;

 temp = nol_supply_w_id[i];
 nol_supply_w_id[i] = nol_supply_w_id[j];
 nol_supply_w_id[j] = temp;

 temp_float = nol_quantity[i];
 nol_quantity[i] = nol_quantity[j];
 nol_quantity[j] = temp_float;

 strncpy(tmpstr,str->newout.i_name[i], 25);
 strncpy(str->newout.i_name[i],str->newout.i_name[j], 25);
 strncpy(str->newout.i_name[j],tmpstr, 25);

 temp = str->newout.s_quantity[i];
 str->newout.s_quantity[i] = str->newout.s_quantity[j];
 str->newout.s_quantity[j] = temp;

 tmpch = str->newout.brand_generic[i];
 str->newout.brand_generic[i] = str->newout.brand_generic[j];
 str->newout.brand_generic[j] = tmpch;
```


Copyright © 2004 NEC Corporation

```
  temp_float = str->newout.i_price[i];
  str->newout.i_price[i] = str->newout.i_price[j];
  str->newout.i_price[j] = temp_float;

  temp_float = str->newout.ol_amount[i];
  str->newout.ol_amount[i] = str->newout.ol_amount[j];
  str->newout.ol_amount[j] = temp_float;

}

#endif
```

tpccsvr.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: tpccsvr.c 7030100.1 95/07/19 15:39:28 plai Generic<base> $ Copyr (c)
1995 Oracle";
#endif /* RCSID */

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

+==============================================================
======+
 | FILENAME
 |   tpccsvr.c
 | DESCRIPTION
 |   Tuxedo server for TPC-C. use a #define TUX
 |   TOPEND server for TPC-C. use a #define TOP

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



#include <stdio.h>
#include <math.h>
#ifdef WINDOWS
#include <windows.h>
#include <process.h>
#endif

#ifdef TUX
#include <atmi.h>         // must occur prior to include of tpccapi.h
#include <stdlib.h>       // for generation of random seed for server id
#include <time.h>         // for generation of random seed for server id
#endif

#define TPCSVR
#include "tpcc.h"
#include "tpcc_info.h"
#ifdef WINDOWS
#include "httpext.h"       //ISAPI DDL information header
#include "tpccapi.h"       //this dlls specific structure, value e.t. header
#endif


#ifdef TUX


  #include <tmenv.h>
  #include <xa.h>
  #include <userlog.h>

/* set up pointers for type casting */
struct newstruct *newinfo;
struct paystruct *payinfo;
struct ordstruct *ordinfo;
struct delstruct *delinfo;
struct stostruct *stoinfo;

//extern void TMlog();

#endif


void TMlog( char *format, ...)
{
        va_list args;
        char buf[4096];
        int len;
```


Copyright © 2004 NEC Corporation

```
        struct tm* newtime;
    time_t now;
        va_start( args, format );
#ifdef WINDOWS
        _strtime( buf );
#else
    time(&now);
    newtime=localtime(&now);
    strcpy(buf, asctime(newtime));
#endif
        strcat( buf," ");
        len = strlen( buf );
#ifdef WINDOWS
        (void)_vsnprintf( buf+ len, sizeof( buf) - len - 1, format, args);
#else
        (void)vsnprintf( buf+ len, sizeof( buf) - len - 1, format, args);
#endif
        buf[sizeof( buf )- 1]= '\0';
        va_end( args );
        userlog( buf );
}


/* FUNCTION: int tpsvrinit (int argc, char *argv[]);
 *
 * PURPOSE:     Connects into database
 * ARGUMENTS:   parameters passed in as  int svrid, char *uid, char *pwd, int
txntype
 *            do not check ordering, assume correct
 *            svrid:   an id number for server running
 *              uid:   the userid for the database
 *              pwd:   the password for the userid
 *          txntype:   transaction type the server will be running
 * RETURNS:     None
 *
 * COMMENTS:    None
 *
 */


int tpsvrinit (int argc, char *argv[])

{
 int svrid, txntype;
 char *uid, *pwd;
 int svrcnt;

    /* pull out the values from argv */
    svrid = atoi(argv[0]);
    uid = argv[1];
    pwd = argv[2];
    txntype = atoi(argv[3]);

#ifdef TUX

            //srand ( (unsigned)time( NULL ) );
     //svrcnt = rand();
     svrcnt = getpid();

    /* send 6 for all txns to be inited */
    /* fix uid an pwd for now, pull out later */
    /* not passing parameters through TUX  yet */
    if (TPCinit (svrcnt, "tpcc", "tpcc", 6)) {
            TMlog( " FAILED to init all txns types");

            return (-1);
    }

    return 0;

#else                                   // for topend
    if (TPCinit (svrid, uid, pwd, txntype)) {
            fprintf(stderr, "Failed in TPCinit (probably connecting).");
            exit (1);
    }

    return (1);
#endif

}


void tpsvrdone ()

{
```


Copyright © 2004 NEC Corporation

```
  TPCexit (0);

}

/* FUNCTION: int NEWORDER(CLIENTDATA *jobData, NewOrderData *neword,
int deadlock)
 *
 * PURPOSE:     This function handles the new order transaction.
 *
 * ARGUMENTS:   deadlock :  count of deadlocks encountered during txn
 *              jobData:    pointer to entire block of user data
 *              neword:     pointer to datastructure in jobData that contains the new order
data
 * RETURNS:            int     TRUE    transaction committed
 *                             FALSE   item number not valid
 *                             -1              deadlock max retry reached
 *
 *
 * COMMENTS:    None
 *
 */

#ifdef TOP
int NEWORDER(CLIENTDATA *jobData, NewOrderData *neword, int deadlock)
#else
void NEWORDER (TPSVCINFO *msg)
#endif

{

#ifdef TOP
  int result;

  result =  TPCnew(neword);

  return result;

#else                           // for Tuxedo



  newinfo = (struct newstruct *) msg->data;
  newinfo->retval = TPCnew (newinfo);  // set return value to 0 or -1
  // always return tpreturn success - let client side poll retval for actual error
  tpreturn (TPSUCCESS, 0, (char *) newinfo, sizeof (struct newstruct), 0);

#endif

}

/* FUNCTION: int PAYMENT(CLIENTDATA *jobData, PaymentData *paydata, int
deadlock)
 *
 * PURPOSE:     This function handles the new order transaction.
 *
 * ARGUMENTS:   deadlock :   count of deadlocks encountered during txn
 *              jobData:     pointer to entire block of user data
 *              paydata:     pointer to datastructure in jobData that contains the new order
data
 * RETURNS:            int     TRUE    transaction committed
 *                             FALSE   item number not valid
 *                             -1              deadlock max retry reached
 *
 *
 * COMMENTS:    None
 *
 */

#ifdef TOP
int PAYMENT(CLIENTDATA *jobData, PaymentData *paydata, int deadlock)
#else
void PAYMENT (TPSVCINFO *msg)
#endif

{

#ifdef TOP

      int result;

      result = TPCpay(paydata);

      return result;
#else
```


Copyright © 2004 NEC Corporation

```
  payinfo = (struct paystruct *) msg->data;
  payinfo->retval = TPCpay (payinfo);  // set return value to 1 or 0 or -1

  // always return tpreturn success - let client side poll retval for actual error
  tpreturn (TPSUCCESS, 0, (char *) payinfo, sizeof (struct paystruct), 0);

#endif

}

/* FUNCTION: int ORDERSTATUS(CLIENTDATA *jobData, OrderStatusData
*orddata, int deadlock)
 *
 * PURPOSE:     This function handles the new order transaction.
 *
 * ARGUMENTS:   deadlock :   count of deadlocks encountered during txn
 *              jobData:     pointer to entire block of user data
 *              orddata:     pointer to datastructure in jobData that contains the new order
data
 * RETURNS:            int     TRUE    transaction committed
 *                             FALSE   item number not valid
 *                             -1              deadlock max retry reached
 *
 *
 * COMMENTS:    None
 *
 */



#ifdef TOP
int ORDERSTATUS(CLIENTDATA *jobData, OrderStatusData *orddata, int
deadlock)
#else
void ORDERSTATUS (TPSVCINFO *msg)
#endif

{

#ifdef TOP
        int result;

        result = TPCord(orddata);
        return result;

#else


  ordinfo = (struct ordstruct *) msg->data;
  ordinfo->retval = TPCord (ordinfo);  // set return value to 0 or -1

  // always return tpreturn success - let client side poll retval for actual error
  tpreturn (TPSUCCESS, 0, (char *) ordinfo, sizeof (struct ordstruct), 0);


#endif

}

/* FUNCTION: int DELIVERY(CLIENTDATA *jobData, DeliveryData *deldata, int
deadlock)
 *
 * PURPOSE:     This function handles the new order transaction.
 *
 * ARGUMENTS:   deadlock :   count of deadlocks encountered during txn
 *              jobData:     pointer to entire block of user data
 *              stodata:     pointer to datastructure in jobData that contains the new order
data
 * RETURNS:            int     TRUE    transaction committed
 *                             FALSE   item number not valid
 *                             -1              deadlock max retry reached
 *
 *
 * COMMENTS:    None
 *
 */

#ifdef TOP
int DELIVERY(CLIENTDATA *jobData, DeliveryData *deldata, int deadlock)
#else
void DELIVERY (TPSVCINFO *msg)
#endif

{

#ifdef TOP
        int result;
```


Copyright © 2004 NEC Corporation

```
        result = TPCdel(deldata);

        return result;

#else


  delinfo = (struct delstruct *) msg->data;
  delinfo->retval = TPCdel (delinfo);  // set return value to 0 or -1

  // always return tpreturn success - let client side poll retval for actual error
  tpreturn (TPSUCCESS, 0, (char *) delinfo, sizeof (struct delstruct), 0);

#endif

}



/* FUNCTION: int STOCKLEVEL(CLIENTDATA *jobData, StockLevelData
*stodata, int deadlock)
 *
 * PURPOSE:     This function handles the new order transaction.
 *
 * ARGUMENTS:   deadlock :   count of deadlocks encountered during txn
 *              jobData:     pointer to entire block of user data
 *              stodata:     pointer to datastructure in jobData that contains the new order
data
 * RETURNS:             int     TRUE    transaction committed
 *                              FALSE   item number not valid
 *                              -1              deadlock max retry reached
 *
 *
 * COMMENTS:    None
 *
 */

#ifdef TOP
int STOCKLEVEL(CLIENTDATA *jobData, StockLevelData *stodata, int deadlock)
#else
void STOCKLEVEL (TPSVCINFO *msg)
#endif
{

#ifdef TOP

        int result;

        result = TPCsto(stodata);

        return result;

#else


  stoinfo = (struct stostruct *) msg->data;
  stoinfo->retval = TPCsto (stoinfo);  // set return value to 0 or -1

  // always return tpreturn success - let client side poll retval for actual error
  tpreturn (TPSUCCESS, 0, (char *) stoinfo, sizeof (struct stostruct), 0);


#endif
}

/* FUNCTION: int OPSTUXSERVER(CLIENTDATA *jobData, NewOrderData
*neword, int deadlock)
 *
 * PURPOSE:     This function handles all transactions.
 *
 * ARGUMENTS:   deadlock :  count of deadlocks encountered during txn
 *              jobData:    pointer to entire block of user data
 *              neword:     pointer to datastructure in jobData that contains the new order
data
 * RETURNS:             int     TRUE    transaction committed
 *                              FALSE   item number not valid
 *                              -1              deadlock max retry reached
 *
 *
 * COMMENTS:    None
 *
 */

#ifdef TOP
int OPSTUXSERVER(CLIENTDATA *jobData, NewOrderData *neword, int
deadlock)
```


Copyright © 2004 NEC Corporation

```
#else
void OPSTUXSERVER (TPSVCINFO *msg)
#endif

{

#ifdef TOP
  int result;

  result =  TPCnew(neword);

  return result;

#else                          // for Tuxedo

  if (msg->len == sizeof(struct newstruct)) {  // len for neworder
     //if (msg->len == 928) {  // len for neworder for old
     newinfo = (struct newstruct *) msg->data;
     newinfo->retval = TPCnew (newinfo);  // set return value to 0 or -1

     // always return tpreturn success - let client side poll retval for actual error
     tpreturn (TPSUCCESS, 0, (char *) newinfo, sizeof (struct newstruct), 0);
  }
  else if (msg->len == sizeof(struct paystruct)) {
            //if (msg->len == 616) {  // len for payment for old
     payinfo = (struct paystruct *) msg->data;
            payinfo->retval = TPCpay (payinfo);  // set return value to 1 or 0 or -1

            // always return tpreturn success - let client side poll retval for actual error
            tpreturn (TPSUCCESS, 0, (char *) payinfo, sizeof (struct paystruct), 0);
  }
  else if (msg->len == sizeof(struct ordstruct)) {
     //if (msg->len == 544) {  // len for order status
     ordinfo = (struct ordstruct *) msg->data;
     ordinfo->retval = TPCord (ordinfo);  // set return value to 0 or -1

     // always return tpreturn success - let client side poll retval for actual error
     tpreturn (TPSUCCESS, 0, (char *) ordinfo, sizeof (struct ordstruct), 0);
  }
  else if (msg->len == sizeof(struct delstruct)) {
     //if (msg->len == 40) {  // len for delivery
     delinfo = (struct delstruct *) msg->data;
     delinfo->retval = TPCdel (delinfo);  // set return value to 0 or -1

     // always return tpreturn success - let client side poll retval for actual error
     tpreturn (TPSUCCESS, 0, (char *) delinfo, sizeof (struct delstruct), 0);
  }
  else {  // assume rest is stock level

     stoinfo = (struct stostruct *) msg->data;
     stoinfo->retval = TPCsto (stoinfo);  // set return value to 0 or -1

     // always return tpreturn success - let client side poll retval for actual error
     tpreturn (TPSUCCESS, 0, (char *) stoinfo, sizeof (struct stostruct), 0);
  }

#endif

}
```

bs-mb.c

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

#if defined(__cplusplus)
extern "C" {
#endif
extern int _tmrunserver _((int));
extern void OPSTUXSERVER _((TPSVCINFO *));
#if defined(__cplusplus)
}
#endif

static struct tmdsptchtbl_t _tmdsptchtbl[] = {
        { "OPSTUXSERVER", "OPSTUXSERVER", (void (*) _((TPSVCINFO *)))
OPSTUXSERVER, 0, 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 */
```


Copyright © 2004 NEC Corporation

```
        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()));
}
```

tpccapi.h

```
#ifndef TPCCAPI_H
#define TPCCAPI_H

/*==================================================================+
 |        Copyright (c) 1997  Oracle Corp, Redwood Shores, CA       |
 |                    OPEN SYSTEMS PERFORMANCE GROUP                |
 |                         All Rights Reserved                      |
 +==================================================================+
 | FILENAME
 |    tpccapi.h
 | DESCRIPTION
 |    header file to tpcc.dll
 +==================================================================*/


//VERSION RESOURCE DEFINES
#define BOOL int
#define TRUE 1
#define FALSE 0
#define _APS_NEXT_RESOURCE_VALUE        101
#define _APS_NEXT_COMMAND_VALUE         40001
#define _APS_NEXT_CONTROL_VALUE         1000
#define _APS_NEXT_SYMED_VALUE           101

typedef char    *LPSTR;

      //note that the welcome form must be processed first as terminal
ids
      //assigned here, once the terminal id is assigned then the forms
can
      //be processed in any order.

#define WELCOME_FORM    1
                        //beginning form no term id assigned, form id
#define MAIN_MENU_FORM  2
                        //term id assigned main menu form id
#define NEW_ORDER_FORM  3
                        //new order form id
#define PAYMENT_FORM    4
                        //payment form id
#define DELIVERY_FORM   5
                        //delivery form id
#define ORDER_STATUS_FORM       6
                        //order status id
#define STOCK_LEVEL_FORM        7
                        //stock level form id

      //This macro is used to prevent the compiler error unused formal
parameter
#define UNUSEDPARAM(x) (x = x)

// this macro for convert IIS struct to Apache
#define EXTENSION_CONTROL_BLOCK              ext_ctrl_block

      //error message structure used in ErrorMessage API
typedef struct _SERRORMSG
{
      int       iError;                 //error id of message
      char      szMsg[80];              //message to sent to browser
} SERRORMSG;

        //This structure defines the data necessary to keep distinct for
each
        //terminal or client connection.
typedef struct _CLIENTDATA
{
      int       inUse;          //in use flag allows client entries to
be reused
      int       w_id;           //warehouse id assigned at welcome form
      int       d_id;           //district id assigned at welcome form
      int       iProcessRequestCounter;         //request counter
      int       iSyncId;        //syncronization id
      int       iTickCount;     //time of last access;
      int       iTermId;        //terminal id of http stream connection
      char      szBuffer[4096]; //form buffer; HTML form built for
client here
      //for TOPEND
      char      type;
      int       retval;
      int       client;
      BOOL      bFailed;
#ifdef TOPEND
      long      *tp_structadr;
#else // assume tuxedo
      struct newstruct          *newOrderDataPtr;       //new order
Tuxedo buffer
      struct paystruct          *paymentDataPtr;        //payment Tuxedo
buffer
```


Copyright © 2004 NEC Corporation

```
      struct ordstruct          *orderStatusDataPtr;    //order status
Tuxedo buffer
      struct delstruct          *deliveryDataPtr;       //delivery
Tuxedo buffer
      struct stostruct          *stockLevelDataPtr;     //stock level
Tuxedo buffer
#endif
      union {
      struct newstruct          newOrderData;   //new order form data
      struct paystruct          paymentData;    //payment form data
      struct ordstruct  orderStatusData;        //order status form data
      struct delstruct          deliveryData;   //delivery form data
      struct stostruct          stocklevelData; //stock level form data
      } trans;
} CLIENTDATA;

typedef CLIENTDATA *PCLIENTDATA;        //pointer to client structure
       //This structure is used to define the operational interface for
       //terminal id support
typedef struct _TERM
{
int     iAvailable;     //total allocated terminal array entries
int     iNext;          //next available terminal array element
int     iMasterSyncId;  //syncronization id
BOOL    bInit;          //structure has been initialized flag
CLIENTDATA      *pClientData;   //pointer to allocated client data
void    (*Init)(void);  //API to initialize this structure
int     (*Allocate)(void); //API to allocate new terminal entry;array id
returnd
void    (*Restore)(void);       //API to free terminal data
//int     (*Add)(EXTENSION_CONTROL_BLOCK *pECB, char *pQueryString);
//API to add a terminal id to array,
int     (*Add)(EXTENSION_CONTROL_BLOCK *, char *);       //API to add a
terminal id to array,
                                        //this context will be passed from
the
                                        //browser to the tpcc.dll in the
                                        //TERMID= key in the HTTP string.
void    (*Delete)(EXTENSION_CONTROL_BLOCK *pECB, int id); //API to free
                                                          //resources used
by a
                                                          //terminal array
                                                          //entry
apr_pool_t *p;  // per server config pool
} TERM;

typedef TERM *PTERM;    //pointer to terminal structure type

      //this structure allows the EXTENSION CONTROL BLOCK to be passed
to the
      //msg and error handlers.
typedef struct _ECBINFO
{
      int       iTermId;        //terminal id
      int       iSyncId;        //browser sync id
      BOOL      bDeadlock;      //deadlock condition flag
      BOOL      bFailed;        //cleared before sql transaction, set in
err
                              //handlers if an error occurs
      EXTENSION_CONTROL_BLOCK *pECB;    //inetsrv current connection
structure
                                        //information
} ECBINFO, *PECBINFO;
```

```
//function prototypes

#ifdef TOPEND // in tuxedo defined in atmi.h
extern int tpsvrinit (int argc, char * argv[]);
             // argv arguments are  int pid, char *uid, char *pwd, int
txntype

extern int NEWORDER(CLIENTDATA *jobData, NewOrderData *neword, int
deadlock);
extern int PAYMENT(CLIENTDATA *jobData, PaymentData *paydata, int
deadlock);
extern int ORDERSTATUS(CLIENTDATA *jobData, OrderStatusData *orddata,
int deadlock);
extern int STOCKLEVEL(CLIENTDATA *jobData, StockLevelData *stodata, int
deadlock);
extern int DELIVERY(CLIENTDATA *jobData, DeliveryData *deldata, int
deadlock);
#endif

#ifdef TUX
extern void NEWORDER(TPSVCINFO *msg);
extern void PAYMENT(TPSVCINFO *msg);
extern void ORDERSTATUS(TPSVCINFO *msg);
extern void STOCKLEVEL(TPSVCINFO *msg);
extern void DELIVERY(TPSVCINFO *msg);
#endif


//BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID
lpReserved);
static BOOL IsValidTermId(int TermId);
BOOL ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd, int
*pFormId, int *pTermId, int *pSyncId);
void NewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId);
void PaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId);
void DeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId);
void OrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId);
void StockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId);
void Exitcmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId);
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId);
void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId);
void ClearCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId);
void MenuCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId);
static void WriteZString(EXTENSION_CONTROL_BLOCK *pECB, char *szStr);
static void h_printf(EXTENSION_CONTROL_BLOCK *pECB, char *format, ...);
void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int iError, int
iErrorType, char *szMsg, int iTermId, int iSyncId);
static BOOL GetKeyValue(char *pQueryString, char *pKey, char *pValue,
int iMax);
void TermInit(server_rec *s);
```


Copyright © 2004 NEC Corporation

```
static void TermRestore(EXTENSION_CONTROL_BLOCK *pECB);
int TermAllocate(server_rec* s);
static int TermAdd(EXTENSION_CONTROL_BLOCK *pECB, char *pQueryString);
static void TermDelete(EXTENSION_CONTROL_BLOCK *pECB, int id);
BOOL Init(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId, char
*szServer, char *szUser, char *szPassword, char *szDatabase);
static BOOL Close(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId);
static void FormatString(char *szDest, char *szPic, char *szSrc);
static char *MakeStockLevelForm(int iTermId, int iSyncId, BOOL bInput);
static char *MakeMainMenuForm(int iTermId, int iSyncId);
static char *MakeWelcomeForm(void);
static char *MakeNewOrderForm(int iTermId, int iSyncId, BOOL bInput,
BOOL bValid, char *execution_status);
static char *MakePaymentForm(int iTermId, int iSyncId, BOOL bInput);
static char *MakeOrderStatusForm(int iTermId, int iSyncId, BOOL bInput);
static char *MakeDeliveryForm(int iTermId, int iSyncId, BOOL bInput,
char *execution_status);
void UtilStrCpy(char * pDest, char * pSrc, int n);
static void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId);
static void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId);
static void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId);
static void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId);
static void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId);
static int GetNewOrderData(LPSTR lpszQueryString, NewOrderData
*pNewOrderData);
static int GetPaymentData(LPSTR lpszQueryString, PaymentData
*pPaymentData);
static int GetOrderStatusData(LPSTR lpszQueryString, OrderStatusData
*pOrderStatusData);
static BOOL ReadRegistrySettings(void);
static BOOL IsNumeric(char *ptr);
static void FormatHTMLString(char *szBuff, char *szStr, int iLen);
static void Log(char *szType, char *szStr);
int do_neworder(NewOrderData *pNOData);
int do_payment(PaymentData *pPAYData);
int do_orderstatus(OrderStatusData *pOSData);
int do_delivery(pDeliveryData pDelivery);
int do_stocklevel(StockLevelData *pSTOData);
int end_neworder(NewOrderData *pNOData);
int end_payment(PaymentData *pPAYData);
int end_orderstatus(OrderStatusData *pOSData);
int end_stocklevel(StockLevelData *pSTOData);
int do_all_txns(int txn);
int do_disconnect( );
DWORD  HttpExtensionProc(EXTENSION_CONTROL_BLOCK *pECB);
apr_status_t tpcc_shm_create(tpcc_shm_t* m, apr_pool_t* p);

#endif /* TPCCAPI_H */
```

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 ();
```


Copyright © 2004 NEC Corporation

```
extern void TPCdumpsto ();
extern void TPCdumpexit ();
#ifndef TUX
extern void userlog(char* fmtp, ...);
#endif

/* 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.hh24: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 float 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 float ol_quantity[15];
extern float ol_amount[15];
ub4    ol_del_len[15];
extern text ol_delivery_d[15][11];
/* xnie - begin */
extern OCIRowid *o_rowid;
/* xnie - end */

/* for payment transaction */

extern int c_w_id;
extern int c_d_id;
extern float 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];
```


Copyright © 2004 NEC Corporation

```
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 float nol_quantity[15];
extern int nol_quanti10[15];
extern int nol_quanti91[15];
extern int nol_ytdqty[15];
extern float nol_amount[15];
extern int o_all_local;
extern float w_tax;
extern float 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 float 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 float 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 oI_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(x)  (x  *)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));



/* bind arrays for sql */
#define
OCIBNDRA(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode) \
       DISCARD ocierror(__FILE__,__LINE__,(errp), \

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

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


/* use with callback data */
#define OCIBNDRAD(stmp,bndp,errp,sqlvar,progvl,ftype,indp,ctxp,\
                  cbf_nodata,cbf_data) \
       DISCARD ocierror(__FILE__,__LINE__,(errp), \

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

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



/* bind in/out for plsql without indicator and rcode */
#define OCIBNDPL(stmp,bndp,errp,sqlvar,progv,progvl,ftype,alen) \
    DISCARD ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
    DISCARD ocierror(__FILE__,__LINE__,(errp), \
      OCIBindByName((stmp),&(bndp),(errp),(CONST text *)(sqlvar), \
        (sb4)strlen((CONST char *)(sqlvar)),
(dvoid*)(progv),(progvl),(ftype),\
```


Copyright © 2004 NEC Corporation

```
              NULLP(dvoid),(alen), NULLP(ub2),
0,NULLP(ub4),OCI_DEFAULT));

/* bind  in values for plsql with indicator and rcode */
#define
OCIBNDR(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode) \
       DISCARD ocierror(__FILE__,__LINE__,(errp), \

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

/* bind in/out for plsql arrays witout indicator and rcode */
#define OCIBNDPLA(stmp,bndp,errp,sqlvar,progv,progvl,ftype,alen,ms,cu) \
       DISCARD ocierror(__FILE__,__LINE__, (errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0));\
       DISCARD ocierror(__FILE__,__LINE__,(errp),\
         OCIBindByName((stmp),&(bndp),(errp),(CONST text *)(sqlvar), \
               (sb4)strlen((CONST char *) (sqlvar)),(void *)(progv), \

(progvl),(ftype),NULL,(alen),NULL,(ms),(cu),OCI_DEFAULT));

/* bind in/out values for plsql with indicator and rcode */
#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)));


#ifndef TPCSVR

/* 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;
   int ol_i_id[15]; //add
   int ol_supply_w_id[15];//add
   int ol_quantity[15];//add
};

struct newstruct {
   int    retval;
   int old_quantity[15];
   struct newinstruct newin;
   struct newoutstruct newout;
};


/* Payment */

struct payinstruct {
   int w_id;
   int d_id;
   int c_w_id;
   int c_d_id;
```


Copyright © 2004 NEC Corporation

```
   int c_id;
   int bylastname;
   float 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 {
   int retval;
   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 {
   int retval;
   struct ordinstruct ordin;
   struct ordoutstruct ordout;
};


/* Delivery */

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

struct deloutstruct {
   int terror;
   int retry;
};

struct delstruct {
   int retval;
   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 {
   int retval;
   struct stoinstruct stoin;
   struct stooutstruct stoout;
};

#endif

#endif
```


Copyright © 2004 NEC Corporation

Makefile

```
#gcc=/usr/local/bin/gcc
#==============================================================
=====+
#       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#               OPEN SYSTEMS PERFORMANCE GROUP                  |
#                   All Rights Reserved                         |
#==============================================================
=====+
# FILENAME
#     Makefile
# DESCRIPTION
#     Makefile for batch driver, load program and tx testing.
#==============================================================
=====
#
#  Programs:
#
#     tpcc.exe       : OCI TPC-C generator
#     tpccload.exe   : Database loader for TPC-C
#     single_txn.exe : OCI program to test the TPC-C transactions
#     getrand.exe    :
#     90per.exe      :
#     runtpb.exe     :
#     sleep.exe      :
#     press_return.exe     :
#     runid.exe      :
#

#all: compile load
all: generic

#include $(ORACLE_HOME)/bench/buildtools/prefix.mk

#I_SYM=-I
#include $(ORACLE_HOME)/rdbms/lib/env_rdbms.mk

LINK=gcc
#/opt/SunProd/SUNWspro6.1/bin/../WS6U1/bin/cc
CC=gcc
#/opt/SunProd/SUNWspro6.1/bin/../WS6U1/bin/cc

#LDFLAGS=-L$(ORACLE_HOME)/rdbms/lib/ -L$(ORACLE_HOME)/lib/ -dy \
#-L$(TUXDIR)/lib/ -L$(ORACLE_HOME)/rdbms/lib/ -L$(ORACLE_HOME)/lib/ \
#$(ORACLE_HOME)/rdbms/lib/defopt.o -lclntsh \
#`cat $(ORACLE_HOME)/lib/sysliblist` \
#-o $@

BENCHRUN_SRC_DIR=/home/oracle/tpcc6000/benchrun/source
CFLAGS=-I$(TUXDIR)/include -DTUX -g

LDFLAGS=-L$(ORACLE_HOME)/rdbms/lib/ -L$(ORACLE_HOME)/lib/ -dy \
-L$(TUXDIR)/lib/ -L$(ORACLE_HOME)/rdbms/lib/ -L$(ORACLE_HOME)/lib/ \
$(ORACLE_HOME)/rdbms/lib/defopt.o -lclntsh \
`cat $(ORACLE_HOME)/lib/sysliblist` \
-L$(ORACLE_HOME)/lib -laio -lm -o $@
#-R$(ORACLE_HOME)/lib -laio -lthread -lposix4 -lkstat -lm -o $@

I_SYM=-I

TARGS=compile cleanup
REMOVE=rm

#DPB_LIB_DIR=${BENCHRUN_SRC_DIR}/lib
#DPB_LIB=$(DPB_LIB_DIR)/dpblibunix.o

TPCBIN=../../bin
INCLUDE=$(I_SYM). $(I_SYM)$(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
# $(I_SYM)$(DPB_LIB_DIR)
ITUX=$(I_SYM)$(ROOTDIR)/include

MEMBS=
OBJS=tpccload.o c_trans.o c_drv_o7.o c_dump.o tpccpl.o getrand.o 90per.o report.o
errrpt.o sleep.o press_return.o runid.o
CTRAN_OBJS=plnew.o plpay.o plord.o pldel.o plsto.o
CTRANPOCI_OBJS=plnew.o plpay_oci.o plord.o pldel.o plsto.o
CTRANTUX_OBJS=plnew_tux.o plpay.o plord.o pldel_tux.o plsto.o
OTHER_OBJS=c_drv_val.o test_drv.o test_sample.o test_tran.o single_txn_ran.o
TUX_OBJS=c_drv_tux.o tpccpl_tux.o tpccsvr.o tpccpl.o bs-mb.o
```


Copyright © 2004 NEC Corporation

```
files:

compile: $(OBJS) $(DPB_LIB)
	@-$(DOTARGS)

load: $(TPCBIN)/tpcc.exe $(TPCBIN)/tpccload.exe \
	$(TPCBIN)/single_txn.exe $(TPCBIN)/90per.exe \
	$(TPCBIN)/runtpb.exe $(TPCBIN)/getrand.exe \
	$(TPCBIN)/sleep.exe $(TPCBIN)/runid.exe \
	$(TPCBIN)/single_txn_ran.exe \
	$(TPCBIN)/press_return.exe
	@-$(DOTARGS)

cleanup:
	$(REMOVE) $(OBJS) $(CTRAN_OBJS) $(CTRANTUX_OBJS)
$(OTHER_OBJS) \
	$(TPCBIN)/tpcc.exe $(TPCBIN)/tpccload.exe \
	$(TPCBIN)/single_txn.exe $(TPCBIN)/90per.exe \
	$(TPCBIN)/runtpb.exe $(TPCBIN)/getrand.exe \
	$(TPCBIN)/sleep.exe $(TPCBIN)/runid.exe \
	$(TPCBIN)/single_txn_ran.exe \
	$(TPCBIN)/press_return.exe \
	$(TUX_OBJS)
	@-$(DOTARGS)

#$(DPB_LIB):
#	( cd $(DPB_LIB_DIR); $(MAKE) -f Makefile.unix )

report.o: report.c results.h
	$(CC) $(CFLAGS) $(INCLUDE) -c report.c

errrpt.o: errrpt.c results.h
	$(CC) $(CFLAGS) $(INCLUDE) -c errrpt.c

sleep.o: sleep.c
	$(CC) $(CFLAGS) $(INCLUDE) -c sleep.c

press_return.o: press_return.c
	$(CC) $(CFLAGS) $(INCLUDE) -c press_return.c

runid.o: runid.c
	$(CC) $(CFLAGS) $(INCLUDE) -c runid.c

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

c_drv_o7.o: c_drv_o7.c tpcc.h
	$(CC) $(CFLAGS) $(INCLUDE) -c c_drv_o7.c

c_drv_val.o: c_drv.c tpcc.h
	$(CP) c_drv.c c_drv_val.c
	-$(CC) $(CFLAGS) -DVALIDATE $(INCLUDE) -c c_drv_val.c
	$(REMOVE) c_drv_val.c

c_drv_tux.o: c_drv.c tpcc.h
	$(CP) c_drv.c c_drv_tux.c
	-$(CC) $(CFLAGS) -DTUX $(INCLUDE) $(ITUX) -c c_drv_tux.c
	$(REMOVE) c_drv_tux.c

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

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

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

runtpb.o: runtpb.c
	$(CC) $(CFLAGS) -DORA_AUX $(INCLUDE) -c runtpb.c

c_trans.o: $(CTRAN_OBJS)
	$(LD) -r -o $@ $(CTRAN_OBJS)

c_trans_tux.o: $(CTRANTUX_OBJS)
	$(LD) -r -o $@ $(CTRANTUX_OBJS)

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

tpccpl_tux.o: tpccpl.c tpcc.h
	$(CP) tpccpl.c tpccpl_tux.c
	-$(CC) $(CFLAGS) -DTUX $(INCLUDE) $(ITUX) -c tpccpl_tux.c
	$(REMOVE) tpccpl_tux.c

plnew_tux.o: plnew.c tpcc.h
	$(CP) plnew.c plnew_tux.c
	-$(CC) $(CFLAGS) -DTUX $(INCLUDE) $(ITUX) -c plnew_tux.c
```


Copyright © 2004 NEC Corporation

```
	$(REMOVE) plnew_tux.c

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

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

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

pldel_tux.o: pldel.c tpcc.h
	$(CP) pldel.c pldel_tux.c
	-$(CC) $(CFLAGS) -DTUX $(INCLUDE) $(ITUX) -c pldel_tux.c
	$(REMOVE) pldel_tux.c

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

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

tpccsvr.o: tpccsvr.c tpcc.h
	$(CC) $(CFLAGS) $(INCLUDE) $(ITUX) -c tpccsvr.c

bs-mb.o:  bs-mb.c
	$(CC) $(CFLAGS) $(INCLUDE) $(ITUX) -c bs-mb.c

generic : $(CTRAN_OBJS) tpccsvr.o bs-mb.o tpccpl.o
	$(LINK) $(LDFLAGS) \
	-ltux -lbuft -lfml -lfml32 -lengine -ldl -lpthread /usr/lib/libcrypt.a \
	tpccsvr.o bs-mb.o tpccpl.o plnew.o pldel.o plsto.o plord.o plpay.o \
	$(SSABED) $(DEF_OPT) $(TTLIBS) -lc


getrand.o: getrand.c
	$(CC) $(CFLAGS) $(INCLUDE) -c getrand.c

90per.o: 90per.c
	$(CC) $(CFLAGS) $(INCLUDE) -c 90per.c

$(TPCBIN)/getrand.exe: getrand.o
	$(LINK) $(LDFLAGS) \
	getrand.o -lc

$(TPCBIN)/sleep.exe: sleep.o
	$(LINK) $(LDFLAGS) \
	sleep.o -lc

$(TPCBIN)/press_return.exe: press_return.o
	$(LINK) $(LDFLAGS) \
	press_return.o -lc

$(TPCBIN)/runid.exe: runid.o
	$(LINK) $(LDFLAGS) \
	runid.o $(DPB_LIB) -lc

$(TPCBIN)/90per.exe: 90per.o
	$(LINK) $(LDFLAGS) \
	90per.o -lc

$(TPCBIN)/tpccload.exe: tpccload.o $(DPB_LIB)
	$(LINK) $(LDFLAGS) \
	tpccload.o $(DPB_LIB) \
	$(SSABED) $(DEF_OPT) $(TTLIBS) -lc

$(TPCBIN)/runtpb.exe: runtpb.o $(DPB_LIB)
	$(LINK) $(LDFLAGS) \
	runtpb.o $(DPB_LIB) \
	$(SSABED) $(DEF_OPT) $(TTLIBS) -lc

$(TPCBIN)/tpcc.exe: c_drv_o7.o c_trans.o tpccpl.o c_dump.o report.o errrpt.o
$(DPB_LIB)
	$(LINK) $(LDFLAGS) \
	c_drv_o7.o c_trans.o tpccpl.o c_dump.o errrpt.o report.o $(DPB_LIB)\
	$(SSABED) $(DEF_OPT) $(TTLIBS) -lc

$(TPCBIN)/single_txn.exe: single_txn.o $(DPB_LIB) c_trans.o tpccpl.o c_dump.o
	$(LINK) $(LDFLAGS) \
	single_txn.o c_trans.o tpccpl.o c_dump.o $(DPB_LIB) \
	$(SSABED) $(DEF_OPT) $(TTLIBS) -lc

$(TPCBIN)/single_txn_ran.exe: single_txn_ran.o $(DPB_LIB) c_trans.o tpccpl.o
c_dump.o
	$(LINK) $(LDFLAGS) \
	single_txn_ran.o c_trans.o tpccpl.o c_dump.o $(DPB_LIB) \
	$(SSABED) $(DEF_OPT) $(TTLIBS) -lc
```


Copyright © 2004 NEC Corporation

paynz.sql

```
DECLARE /* paynz */
      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 ware
               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 inittpcc.ware_name, :w_street_1, :w_street_2, :w_city,
                   :w_state, :w_zip;

           UPDATE  cust
              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
inittpcc.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;


         IF :c_credit = 'BC' THEN
            UPDATE cust
               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 = inittpcc.cust_rowid
         RETURNING substr(c_data,1, 200)
              INTO :c_data;

         END IF;

            UPDATE dist
               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
inittpcc.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 hist  (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, inittpcc.ware_name || '    ' ||
inittpcc.dist_name);
         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 ware
            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 inittpcc.ware_name,
                   :w_street_1, :w_street_2, :w_city, :w_state, :w_zip;


         SELECT rowid
         BULK COLLECT INTO inittpcc.row_id
         FROM cust
         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;

       inittpcc.c_num := sql%rowcount;
       inittpcc.cust_rowid := inittpcc.row_id((inittpcc.c_num) / 2);

        UPDATE cust
          SET c_balance = c_balance - :h_amount,
              c_ytd_payment = c_ytd_payment+ :h_amount,
              c_payment_cnt = c_payment_cnt+1
        WHERE rowid = inittpcc.cust_rowid
        RETURNING
```


Copyright © 2004 NEC Corporation

```
                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 cust
              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 = inittpcc.cust_rowid
              RETURNING substr(c_data,1, 200)
              INTO :c_data;

         END IF;

         UPDATE dist
            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 inittpcc.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 hist (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, inittpcc.ware_name || '    ' ||
inittpcc.dist_name);

         EXIT;

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

      END LOOP;
   END;
```

tkvcinin.sql

```
-- The initnew package for storing variables used in the
-- New Order anonymous block

CREATE OR REPLACE PACKAGE inittpcc
AS
 TYPE intarray IS TABLE OF INTEGER INDEX BY BINARY_INTEGER;
 TYPE distarray IS TABLE OF VARCHAR(24) INDEX BY BINARY_INTEGER;
 nulldate      DATE;
 TYPE rowidarray IS TABLE OF ROWID INDEX BY PLS_INTEGER;
 s_dist                    distarray;
 idx1arr            intarray;
 s_remote                  intarray;
 dist               intarray;
 row_id             rowidarray;
 cust_rowid           rowid;
 dist_name            VARCHAR2(11);
 ware_name            VARCHAR2(11);
 c_num              PLS_INTEGER;

 PROCEDURE init_no(idxarr intarray);
 PROCEDURE init_del;
 PROCEDURE init_pay;
END inittpcc;
/
show errors;

CREATE OR REPLACE PACKAGE BODY inittpcc AS
  PROCEDURE init_no (idxarr  intarray)
  IS
  BEGIN
      -- initialize null date
   nulldate := TO_DATE('01-01-1811', 'MM-DD-YYYY');
    idx1arr := idxarr;
  END init_no;

  PROCEDURE init_del
  IS
  BEGIN
    FOR i IN 1 .. 10 LOOP
      dist(i) := i;
    END LOOP;
  END init_del;

  PROCEDURE init_pay IS
  BEGIN
```


Copyright © 2004 NEC Corporation

```
  NULL;
 END init_pay;

END inittpcc;
/
show errors
exit
```

tkvcpdel.sql

```
declare
 TYPE numarray IS TABLE OF NUMBER INDEX BY BINARY_INTEGER;
 TYPE numlist is varray (10) of number;
 dist numarray;
 amt numarray ;
 cnt pls_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);

BEGIN
 LOOP BEGIN
  FORALL d IN 1..10
   DELETE FROM nord N
    WHERE no_d_id = inittpcc.dist(d)
     AND no_w_id = :w_id
     AND no_o_id = (select min(no_o_id)
                    from nord
                    where no_d_id = N.no_d_id
                    and no_w_id = N.no_w_id)
    RETURNING no_d_id, no_o_id BULK COLLECT INTO :d_id, :order_id;

  :ordcnt := SQL%ROWCOUNT;

  FORALL o in 1.. :ordcnt
   UPDATE ordr SET o_carrier_id = :carrier_id
   WHERE o_id = :order_id (o)
    AND o_d_id = :d_id(o)
    AND o_w_id = :w_id
    RETURNING o_c_id BULK COLLECT INTO :o_c_id;

  FORALL o in 1.. :ordcnt
   UPDATE ordl SET ol_delivery_d = :now
   WHERE ol_w_id = :w_id
    AND ol_d_id = :d_id(o)
    AND ol_o_id = :order_id(o)
   RETURNING sum(ol_amount) BULK COLLECT INTO  :sums;

  FORALL c IN 1.. :ordcnt
   UPDATE cust
    SET c_balance = c_balance + :sums(c),
        c_delivery_cnt = c_delivery_cnt + 1
   WHERE c_w_id = :w_id
    AND c_d_id = :d_id(c)
    AND c_id = :o_c_id(c);
  COMMIT;
  EXIT;
  EXCEPTION
   WHEN not_serializable OR deadlock OR snapshot_too_old
   THEN
    ROLLBACK;
    :retry := :retry + 1;
  END;

 END LOOP; -- for retry
END;
```

tkvcpnew.sql

```
-- New Order Anonymous block

 DECLARE
     idx                      PLS_INTEGER;
     dummy_local              PLS_INTEGER;
     cache_ol_cnt             PLS_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 .. cache_ol_cnt
         UPDATE  stock_item
         SET s_order_cnt = s_order_cnt + 1,
```


Copyright © 2004 NEC Corporation

```
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                             THEN s_quantity +91
                             ELSE s_quantity
                        END) - :ol_quantity(idx)
          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,
                    i_price*:ol_quantity(idx),
            CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                 THEN 'G'
                  ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                        THEN 'G'
                        ELSE 'B'
                        END)
            END
        BULK COLLECT INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                        :ol_amount,:brand_generic;
   END u1;

   PROCEDURE u2 IS
   BEGIN
        FORALL idx IN 1 .. cache_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 = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                             THEN s_quantity +91
                             ELSE s_quantity
                        END) - :ol_quantity(idx)
          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,
                    i_price*:ol_quantity(idx),
            CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                 THEN 'G'
                  ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                        THEN 'G'
                        ELSE 'B'
                        END)
            END
        BULK COLLECT INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                        :ol_amount,:brand_generic;
   END u2;

   PROCEDURE u3 IS
   BEGIN
        FORALL idx IN 1 .. cache_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 = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                             THEN s_quantity +91
                             ELSE s_quantity
                        END) - :ol_quantity(idx)
          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,
                    i_price*:ol_quantity(idx),
            CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                 THEN 'G'
                  ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                        THEN 'G'
                        ELSE 'B'
                        END)
            END
        BULK COLLECT INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                        :ol_amount,:brand_generic;
   END u3;

   PROCEDURE u4 IS
   BEGIN
        FORALL idx IN 1 .. cache_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 = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                             THEN s_quantity +91
                             ELSE s_quantity
                        END) - :ol_quantity(idx)
          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,
                    i_price*:ol_quantity(idx),
            CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                 THEN 'G'
                  ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                        THEN 'G'
                        ELSE 'B'
                        END)
            END
        BULK COLLECT INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                        :ol_amount,:brand_generic;
   END u4;

   PROCEDURE u5 IS
   BEGIN
        FORALL idx IN 1 .. cache_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 = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                             THEN s_quantity +91
                             ELSE s_quantity
                        END) - :ol_quantity(idx)
          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,
                    i_price*:ol_quantity(idx),
            CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                 THEN 'G'
                  ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                        THEN 'G'
                        ELSE 'B'
                        END)
            END
```


Copyright © 2004 NEC Corporation

```
        BULK COLLECT INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                          :ol_amount,:brand_generic;
   END u5;

   PROCEDURE u6 IS
   BEGIN
        FORALL idx IN 1 .. cache_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 = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                             THEN s_quantity +91
                             ELSE s_quantity
                        END) - :ol_quantity(idx)
          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,
                    i_price*:ol_quantity(idx),
            CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                 THEN 'G'
                  ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                        THEN 'G'
                        ELSE 'B'
                        END)
            END
        BULK COLLECT INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                          :ol_amount,:brand_generic;
   END u6;

   PROCEDURE u7 IS
   BEGIN
        FORALL idx IN 1 .. cache_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 = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                             THEN s_quantity +91
                             ELSE s_quantity
                        END) - :ol_quantity(idx)
          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,
                    i_price*:ol_quantity(idx),
            CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                 THEN 'G'
                  ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                        THEN 'G'
                        ELSE 'B'
                        END)
            END
        BULK COLLECT INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                          :ol_amount,:brand_generic;
   END u7;

   PROCEDURE u8 IS
   BEGIN
        FORALL idx IN 1 .. cache_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 = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                             THEN s_quantity +91
                             ELSE s_quantity
                        END) - :ol_quantity(idx)
          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,
                    i_price*:ol_quantity(idx),
            CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                 THEN 'G'
                  ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                        THEN 'G'
                        ELSE 'B'
                        END)
            END
        BULK COLLECT INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                          :ol_amount,:brand_generic;
   END u8;

   PROCEDURE u9 IS
   BEGIN
        FORALL idx IN 1 .. cache_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 = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                             THEN s_quantity +91
                             ELSE s_quantity
                        END) - :ol_quantity(idx)
          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,
                    i_price*:ol_quantity(idx),
            CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                 THEN 'G'
                  ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                        THEN 'G'
                        ELSE 'B'
                        END)
            END
        BULK COLLECT INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                          :ol_amount,:brand_generic;
   END u9;

   PROCEDURE u10 IS
   BEGIN
        FORALL idx IN 1 .. cache_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 = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                             THEN s_quantity +91
                             ELSE s_quantity
                        END) - :ol_quantity(idx)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
```


Copyright © 2004 NEC Corporation

```
          RETURNING i_price, i_name, s_quantity, s_dist_10,
                    i_price*:ol_quantity(idx),
            CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                 THEN 'G'
                  ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                        THEN 'G'
                        ELSE 'B'
                        END)
            END
        BULK COLLECT INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                          :ol_amount,:brand_generic;
   END u10;

   PROCEDURE fix_items IS
     rows_lost                      PLS_INTEGER;
     max_index                      PLS_INTEGER;
     temp_index                     PLS_INTEGER;
   BEGIN
     idx := 1;
     rows_lost := 0;
     max_index := dummy_local;

     WHILE (max_index != cache_ol_cnt) LOOP

      WHILE (idx <= sql%rowcount AND
                     sql%bulk_rowcount(idx + rows_lost) = 1)
      LOOP
        idx := idx + 1;
      END LOOP;

      temp_index := max_index;
      WHILE (temp_index >= idx + rows_lost) LOOP
         :ol_amount(temp_index + 1)      :=  :ol_amount(temp_index);
         :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);
         inittpcc.s_dist(temp_index + 1) :=
inittpcc.s_dist(temp_index);
         :brand_generic(temp_index + 1)  :=  :brand_generic(temp_index);
         temp_index := temp_index - 1;
      END LOOP;

      IF (idx + rows_lost <= cache_ol_cnt) THEN
        :i_price(idx + rows_lost)        :=   0;
        :i_name(idx + rows_lost)         :=   'NO ITEM';
        :s_quantity(idx + rows_lost)     :=   0;
        inittpcc.s_dist(idx + rows_lost) := NULL;
        :brand_generic(idx + rows_lost)  := ' ';
        :ol_amount(idx + rows_lost)      := 0;
        rows_lost := rows_lost + 1;
        max_index := max_index + 1;
      END IF;

    END LOOP;
   END fix_items;

   BEGIN
      LOOP BEGIN
         cache_ol_cnt := :o_ol_cnt;

         UPDATE dist 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 cust , ware
          WHERE c_id = :c_id AND c_d_id = :d_id AND c_w_id = w_id
          AND  w_id = :w_id;


         INSERT INTO nord (no_o_id, no_d_id, no_w_id)
          VALUES (:o_id, :d_id, :w_id);

         INSERT INTO ordr  (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);


         dummy_local :=  :d_id;

         IF (dummy_local < 6) THEN
           IF (dummy_local < 3) THEN
             IF (dummy_local = 1) THEN
                u1;
             ELSE
                u2;
             END IF;
           ELSE
             IF (dummy_local = 3) THEN
                u3;
             ELSIF (dummy_local = 4) then
                u4;
             ELSE
                u5;
             END IF;
           END IF;
         ELSE
           IF (dummy_local < 8) THEN
             IF (dummy_local = 6) THEN
                u6;
             ELSE
                u7;
             END IF;
           ELSE
             IF (dummy_local = 8) THEN
                u8;
             ELSIF (dummy_local = 9) then
                u9;
             ELSE
                u10;
             END IF;
           END IF;
         END IF;

        dummy_local := sql%rowcount;

        IF (dummy_local != cache_ol_cnt )  THEN fix_items; END IF;

        FORALL idx IN 1..dummy_local
         INSERT INTO ordl
            (ol_o_id, ol_d_id, ol_w_id, ol_number, ol_delivery_d,
ol_i_id,
```


Copyright © 2004 NEC Corporation

```
                 ol_supply_w_id, ol_quantity,ol_amount,ol_dist_info)
         VALUES (:o_id, :d_id, :w_id, inittpcc.idx1arr(idx),
inittpcc.nulldate,
                 :ol_i_id(idx), :ol_supply_w_id(idx),
                 :ol_quantity(idx), :ol_amount(idx),
inittpcc.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;
```

Appendix B : Database Design

B.1 Database build scripts

rawmap.sh

```
raw /dev/raw/raw21 /dev/sdab1
chown oracle:dba /dev/raw/raw21
chmod 660 /dev/raw/raw21
raw /dev/raw/raw93 /dev/sdab2
chown oracle:dba /dev/raw/raw93
chmod 660 /dev/raw/raw93
raw /dev/raw/raw165 /dev/sdab3
chown oracle:dba /dev/raw/raw165
chmod 660 /dev/raw/raw165
raw /dev/raw/raw237 /dev/sdab5
chown oracle:dba /dev/raw/raw237
chmod 660 /dev/raw/raw237
raw /dev/raw/raw309 /dev/sdab6
chown oracle:dba /dev/raw/raw309
chmod 660 /dev/raw/raw309
raw /dev/raw/raw381 /dev/sdab7
chown oracle:dba /dev/raw/raw381
chmod 660 /dev/raw/raw381
raw /dev/raw/raw453 /dev/sdab8
chown oracle:dba /dev/raw/raw453
chmod 660 /dev/raw/raw453
raw /dev/raw/raw525 /dev/sdab9
chown oracle:dba /dev/raw/raw525
chmod 660 /dev/raw/raw525
raw /dev/raw/raw597 /dev/sdab10
chown oracle:dba /dev/raw/raw597
chmod 660 /dev/raw/raw597
raw /dev/raw/raw669 /dev/sdab11
chown oracle:dba /dev/raw/raw669
chmod 660 /dev/raw/raw669
raw /dev/raw/raw741 /dev/sdab12
chown oracle:dba /dev/raw/raw741
chmod 660 /dev/raw/raw741
raw /dev/raw/raw813 /dev/sdab13
chown oracle:dba /dev/raw/raw813
chmod 660 /dev/raw/raw813
raw /dev/raw/raw885 /dev/sdab14
chown oracle:dba /dev/raw/raw885
chmod 660 /dev/raw/raw885
raw /dev/raw/raw957 /dev/sdab15
chown oracle:dba /dev/raw/raw957
chmod 660 /dev/raw/raw957
raw /dev/raw/raw22 /dev/sdac1
chown oracle:dba /dev/raw/raw22
chmod 660 /dev/raw/raw22
raw /dev/raw/raw94 /dev/sdac2
chown oracle:dba /dev/raw/raw94
chmod 660 /dev/raw/raw94
raw /dev/raw/raw166 /dev/sdac3
chown oracle:dba /dev/raw/raw166
chmod 660 /dev/raw/raw166
raw /dev/raw/raw238 /dev/sdac5
chown oracle:dba /dev/raw/raw238
chmod 660 /dev/raw/raw238
raw /dev/raw/raw310 /dev/sdac6
chown oracle:dba /dev/raw/raw310
chmod 660 /dev/raw/raw310
raw /dev/raw/raw382 /dev/sdac7
chown oracle:dba /dev/raw/raw382
chmod 660 /dev/raw/raw382
raw /dev/raw/raw454 /dev/sdac8
chown oracle:dba /dev/raw/raw454
chmod 660 /dev/raw/raw454
raw /dev/raw/raw526 /dev/sdac9
chown oracle:dba /dev/raw/raw526
chmod 660 /dev/raw/raw526
raw /dev/raw/raw598 /dev/sdac10
chown oracle:dba /dev/raw/raw598
chmod 660 /dev/raw/raw598
raw /dev/raw/raw670 /dev/sdac11
```


Copyright © 2004 NEC Corporation

```
chown oracle:dba /dev/raw/raw670
chmod 660 /dev/raw/raw670
raw /dev/raw/raw742 /dev/sdac12
chown oracle:dba /dev/raw/raw742
chmod 660 /dev/raw/raw742
raw /dev/raw/raw814 /dev/sdac13
chown oracle:dba /dev/raw/raw814
chmod 660 /dev/raw/raw814
raw /dev/raw/raw886 /dev/sdac14
chown oracle:dba /dev/raw/raw886
chmod 660 /dev/raw/raw886
raw /dev/raw/raw958 /dev/sdac15
chown oracle:dba /dev/raw/raw958
chmod 660 /dev/raw/raw958
raw /dev/raw/raw23 /dev/sdae1
chown oracle:dba /dev/raw/raw23
chmod 660 /dev/raw/raw23
raw /dev/raw/raw95 /dev/sdae2
chown oracle:dba /dev/raw/raw95
chmod 660 /dev/raw/raw95
raw /dev/raw/raw167 /dev/sdae3
chown oracle:dba /dev/raw/raw167
chmod 660 /dev/raw/raw167
raw /dev/raw/raw239 /dev/sdae5
chown oracle:dba /dev/raw/raw239
chmod 660 /dev/raw/raw239
raw /dev/raw/raw311 /dev/sdae6
chown oracle:dba /dev/raw/raw311
chmod 660 /dev/raw/raw311
raw /dev/raw/raw383 /dev/sdae7
chown oracle:dba /dev/raw/raw383
chmod 660 /dev/raw/raw383
raw /dev/raw/raw455 /dev/sdae8
chown oracle:dba /dev/raw/raw455
chmod 660 /dev/raw/raw455
raw /dev/raw/raw527 /dev/sdae9
chown oracle:dba /dev/raw/raw527
chmod 660 /dev/raw/raw527
raw /dev/raw/raw599 /dev/sdae10
chown oracle:dba /dev/raw/raw599
chmod 660 /dev/raw/raw599
raw /dev/raw/raw671 /dev/sdae11
chown oracle:dba /dev/raw/raw671
chmod 660 /dev/raw/raw671
raw /dev/raw/raw743 /dev/sdae12
chown oracle:dba /dev/raw/raw743
chmod 660 /dev/raw/raw743
raw /dev/raw/raw815 /dev/sdae13
chown oracle:dba /dev/raw/raw815
chmod 660 /dev/raw/raw815
raw /dev/raw/raw887 /dev/sdae14
chown oracle:dba /dev/raw/raw887
chmod 660 /dev/raw/raw887
raw /dev/raw/raw959 /dev/sdae15
chown oracle:dba /dev/raw/raw959
chmod 660 /dev/raw/raw959
raw /dev/raw/raw24 /dev/sdaf1
chown oracle:dba /dev/raw/raw24
chmod 660 /dev/raw/raw24
raw /dev/raw/raw96 /dev/sdaf2
chown oracle:dba /dev/raw/raw96
chmod 660 /dev/raw/raw96
raw /dev/raw/raw168 /dev/sdaf3
chown oracle:dba /dev/raw/raw168
chmod 660 /dev/raw/raw168
raw /dev/raw/raw240 /dev/sdaf5
chown oracle:dba /dev/raw/raw240
chmod 660 /dev/raw/raw240
raw /dev/raw/raw312 /dev/sdaf6
chown oracle:dba /dev/raw/raw312
chmod 660 /dev/raw/raw312
raw /dev/raw/raw384 /dev/sdaf7
chown oracle:dba /dev/raw/raw384
chmod 660 /dev/raw/raw384
raw /dev/raw/raw456 /dev/sdaf8
chown oracle:dba /dev/raw/raw456
chmod 660 /dev/raw/raw456
raw /dev/raw/raw528 /dev/sdaf9
chown oracle:dba /dev/raw/raw528
chmod 660 /dev/raw/raw528
raw /dev/raw/raw600 /dev/sdaf10
chown oracle:dba /dev/raw/raw600
chmod 660 /dev/raw/raw600
raw /dev/raw/raw672 /dev/sdaf11
chown oracle:dba /dev/raw/raw672
chmod 660 /dev/raw/raw672
raw /dev/raw/raw744 /dev/sdaf12
chown oracle:dba /dev/raw/raw744
chmod 660 /dev/raw/raw744
raw /dev/raw/raw816 /dev/sdaf13
chown oracle:dba /dev/raw/raw816
chmod 660 /dev/raw/raw816
raw /dev/raw/raw888 /dev/sdaf14
chown oracle:dba /dev/raw/raw888
chmod 660 /dev/raw/raw888
raw /dev/raw/raw960 /dev/sdaf15
chown oracle:dba /dev/raw/raw960
chmod 660 /dev/raw/raw960
raw /dev/raw/raw25 /dev/sdag1
chown oracle:dba /dev/raw/raw25
chmod 660 /dev/raw/raw25
raw /dev/raw/raw97 /dev/sdag2
chown oracle:dba /dev/raw/raw97
chmod 660 /dev/raw/raw97
raw /dev/raw/raw169 /dev/sdag3
chown oracle:dba /dev/raw/raw169
chmod 660 /dev/raw/raw169
raw /dev/raw/raw241 /dev/sdag5
chown oracle:dba /dev/raw/raw241
chmod 660 /dev/raw/raw241
raw /dev/raw/raw313 /dev/sdag6
chown oracle:dba /dev/raw/raw313
```


Copyright © 2004 NEC Corporation

```
chmod 660 /dev/raw/raw313
raw /dev/raw/raw385 /dev/sdag7
chown oracle:dba /dev/raw/raw385
chmod 660 /dev/raw/raw385
raw /dev/raw/raw457 /dev/sdag8
chown oracle:dba /dev/raw/raw457
chmod 660 /dev/raw/raw457
raw /dev/raw/raw529 /dev/sdag9
chown oracle:dba /dev/raw/raw529
chmod 660 /dev/raw/raw529
raw /dev/raw/raw601 /dev/sdag10
chown oracle:dba /dev/raw/raw601
chmod 660 /dev/raw/raw601
raw /dev/raw/raw673 /dev/sdag11
chown oracle:dba /dev/raw/raw673
chmod 660 /dev/raw/raw673
raw /dev/raw/raw745 /dev/sdag12
chown oracle:dba /dev/raw/raw745
chmod 660 /dev/raw/raw745
raw /dev/raw/raw817 /dev/sdag13
chown oracle:dba /dev/raw/raw817
chmod 660 /dev/raw/raw817
raw /dev/raw/raw889 /dev/sdag14
chown oracle:dba /dev/raw/raw889
chmod 660 /dev/raw/raw889
raw /dev/raw/raw961 /dev/sdag15
chown oracle:dba /dev/raw/raw961
chmod 660 /dev/raw/raw961
raw /dev/raw/raw26 /dev/sdah1
chown oracle:dba /dev/raw/raw26
chmod 660 /dev/raw/raw26
raw /dev/raw/raw98 /dev/sdah2
chown oracle:dba /dev/raw/raw98
chmod 660 /dev/raw/raw98
raw /dev/raw/raw170 /dev/sdah3
chown oracle:dba /dev/raw/raw170
chmod 660 /dev/raw/raw170
raw /dev/raw/raw242 /dev/sdah5
chown oracle:dba /dev/raw/raw242
chmod 660 /dev/raw/raw242
raw /dev/raw/raw314 /dev/sdah6
chown oracle:dba /dev/raw/raw314
chmod 660 /dev/raw/raw314
raw /dev/raw/raw386 /dev/sdah7
chown oracle:dba /dev/raw/raw386
chmod 660 /dev/raw/raw386
raw /dev/raw/raw458 /dev/sdah8
chown oracle:dba /dev/raw/raw458
chmod 660 /dev/raw/raw458
raw /dev/raw/raw530 /dev/sdah9
chown oracle:dba /dev/raw/raw530
chmod 660 /dev/raw/raw530
raw /dev/raw/raw602 /dev/sdah10
chown oracle:dba /dev/raw/raw602
chmod 660 /dev/raw/raw602
raw /dev/raw/raw674 /dev/sdah11
chown oracle:dba /dev/raw/raw674
chmod 660 /dev/raw/raw674
raw /dev/raw/raw746 /dev/sdah12
chown oracle:dba /dev/raw/raw746
chmod 660 /dev/raw/raw746
raw /dev/raw/raw818 /dev/sdah13
chown oracle:dba /dev/raw/raw818
chmod 660 /dev/raw/raw818
raw /dev/raw/raw890 /dev/sdah14
chown oracle:dba /dev/raw/raw890
chmod 660 /dev/raw/raw890
raw /dev/raw/raw962 /dev/sdah15
chown oracle:dba /dev/raw/raw962
chmod 660 /dev/raw/raw962
raw /dev/raw/raw27 /dev/sdai1
chown oracle:dba /dev/raw/raw27
chmod 660 /dev/raw/raw27
raw /dev/raw/raw99 /dev/sdai2
chown oracle:dba /dev/raw/raw99
chmod 660 /dev/raw/raw99
raw /dev/raw/raw171 /dev/sdai3
chown oracle:dba /dev/raw/raw171
chmod 660 /dev/raw/raw171
raw /dev/raw/raw243 /dev/sdai5
chown oracle:dba /dev/raw/raw243
chmod 660 /dev/raw/raw243
raw /dev/raw/raw315 /dev/sdai6
chown oracle:dba /dev/raw/raw315
chmod 660 /dev/raw/raw315
raw /dev/raw/raw387 /dev/sdai7
chown oracle:dba /dev/raw/raw387
chmod 660 /dev/raw/raw387
raw /dev/raw/raw459 /dev/sdai8
chown oracle:dba /dev/raw/raw459
chmod 660 /dev/raw/raw459
raw /dev/raw/raw531 /dev/sdai9
chown oracle:dba /dev/raw/raw531
chmod 660 /dev/raw/raw531
raw /dev/raw/raw603 /dev/sdai10
chown oracle:dba /dev/raw/raw603
chmod 660 /dev/raw/raw603
raw /dev/raw/raw675 /dev/sdai11
chown oracle:dba /dev/raw/raw675
chmod 660 /dev/raw/raw675
raw /dev/raw/raw747 /dev/sdai12
chown oracle:dba /dev/raw/raw747
chmod 660 /dev/raw/raw747
raw /dev/raw/raw819 /dev/sdai13
chown oracle:dba /dev/raw/raw819
chmod 660 /dev/raw/raw819
raw /dev/raw/raw891 /dev/sdai14
chown oracle:dba /dev/raw/raw891
chmod 660 /dev/raw/raw891
raw /dev/raw/raw963 /dev/sdai15
chown oracle:dba /dev/raw/raw963
chmod 660 /dev/raw/raw963
```


Copyright © 2004 NEC Corporation

```
raw /dev/raw/raw28 /dev/sdaj1
chown oracle:dba /dev/raw/raw28
chmod 660 /dev/raw/raw28
raw /dev/raw/raw100 /dev/sdaj2
chown oracle:dba /dev/raw/raw100
chmod 660 /dev/raw/raw100
raw /dev/raw/raw172 /dev/sdaj3
chown oracle:dba /dev/raw/raw172
chmod 660 /dev/raw/raw172
raw /dev/raw/raw244 /dev/sdaj5
chown oracle:dba /dev/raw/raw244
chmod 660 /dev/raw/raw244
raw /dev/raw/raw316 /dev/sdaj6
chown oracle:dba /dev/raw/raw316
chmod 660 /dev/raw/raw316
raw /dev/raw/raw388 /dev/sdaj7
chown oracle:dba /dev/raw/raw388
chmod 660 /dev/raw/raw388
raw /dev/raw/raw460 /dev/sdaj8
chown oracle:dba /dev/raw/raw460
chmod 660 /dev/raw/raw460
raw /dev/raw/raw532 /dev/sdaj9
chown oracle:dba /dev/raw/raw532
chmod 660 /dev/raw/raw532
raw /dev/raw/raw604 /dev/sdaj10
chown oracle:dba /dev/raw/raw604
chmod 660 /dev/raw/raw604
raw /dev/raw/raw676 /dev/sdaj11
chown oracle:dba /dev/raw/raw676
chmod 660 /dev/raw/raw676
raw /dev/raw/raw748 /dev/sdaj12
chown oracle:dba /dev/raw/raw748
chmod 660 /dev/raw/raw748
raw /dev/raw/raw820 /dev/sdaj13
chown oracle:dba /dev/raw/raw820
chmod 660 /dev/raw/raw820
raw /dev/raw/raw892 /dev/sdaj14
chown oracle:dba /dev/raw/raw892
chmod 660 /dev/raw/raw892
raw /dev/raw/raw964 /dev/sdaj15
chown oracle:dba /dev/raw/raw964
chmod 660 /dev/raw/raw964
raw /dev/raw/raw29 /dev/sdal1
chown oracle:dba /dev/raw/raw29
chmod 660 /dev/raw/raw29
raw /dev/raw/raw101 /dev/sdal2
chown oracle:dba /dev/raw/raw101
chmod 660 /dev/raw/raw101
raw /dev/raw/raw173 /dev/sdal3
chown oracle:dba /dev/raw/raw173
chmod 660 /dev/raw/raw173
raw /dev/raw/raw245 /dev/sdal5
chown oracle:dba /dev/raw/raw245
chmod 660 /dev/raw/raw245
raw /dev/raw/raw317 /dev/sdal6
chown oracle:dba /dev/raw/raw317
chmod 660 /dev/raw/raw317
raw /dev/raw/raw389 /dev/sdal7
chown oracle:dba /dev/raw/raw389
chmod 660 /dev/raw/raw389
raw /dev/raw/raw461 /dev/sdal8
chown oracle:dba /dev/raw/raw461
chmod 660 /dev/raw/raw461
raw /dev/raw/raw533 /dev/sdal9
chown oracle:dba /dev/raw/raw533
chmod 660 /dev/raw/raw533
raw /dev/raw/raw605 /dev/sdal10
chown oracle:dba /dev/raw/raw605
chmod 660 /dev/raw/raw605
raw /dev/raw/raw677 /dev/sdal11
chown oracle:dba /dev/raw/raw677
chmod 660 /dev/raw/raw677
raw /dev/raw/raw749 /dev/sdal12
chown oracle:dba /dev/raw/raw749
chmod 660 /dev/raw/raw749
raw /dev/raw/raw821 /dev/sdal13
chown oracle:dba /dev/raw/raw821
chmod 660 /dev/raw/raw821
raw /dev/raw/raw893 /dev/sdal14
chown oracle:dba /dev/raw/raw893
chmod 660 /dev/raw/raw893
raw /dev/raw/raw965 /dev/sdal15
chown oracle:dba /dev/raw/raw965
chmod 660 /dev/raw/raw965
raw /dev/raw/raw30 /dev/sdam1
chown oracle:dba /dev/raw/raw30
chmod 660 /dev/raw/raw30
raw /dev/raw/raw102 /dev/sdam2
chown oracle:dba /dev/raw/raw102
chmod 660 /dev/raw/raw102
raw /dev/raw/raw174 /dev/sdam3
chown oracle:dba /dev/raw/raw174
chmod 660 /dev/raw/raw174
raw /dev/raw/raw246 /dev/sdam5
chown oracle:dba /dev/raw/raw246
chmod 660 /dev/raw/raw246
raw /dev/raw/raw318 /dev/sdam6
chown oracle:dba /dev/raw/raw318
chmod 660 /dev/raw/raw318
raw /dev/raw/raw390 /dev/sdam7
chown oracle:dba /dev/raw/raw390
chmod 660 /dev/raw/raw390
raw /dev/raw/raw462 /dev/sdam8
chown oracle:dba /dev/raw/raw462
chmod 660 /dev/raw/raw462
raw /dev/raw/raw534 /dev/sdam9
chown oracle:dba /dev/raw/raw534
chmod 660 /dev/raw/raw534
raw /dev/raw/raw606 /dev/sdam10
chown oracle:dba /dev/raw/raw606
chmod 660 /dev/raw/raw606
raw /dev/raw/raw678 /dev/sdam11
```


Copyright © 2004 NEC Corporation

```
chown oracle:dba /dev/raw/raw678
chmod 660 /dev/raw/raw678
raw /dev/raw/raw750 /dev/sdam12
chown oracle:dba /dev/raw/raw750
chmod 660 /dev/raw/raw750
raw /dev/raw/raw822 /dev/sdam13
chown oracle:dba /dev/raw/raw822
chmod 660 /dev/raw/raw822
raw /dev/raw/raw894 /dev/sdam14
chown oracle:dba /dev/raw/raw894
chmod 660 /dev/raw/raw894
raw /dev/raw/raw966 /dev/sdam15
chown oracle:dba /dev/raw/raw966
chmod 660 /dev/raw/raw966
raw /dev/raw/raw31 /dev/sdao1
chown oracle:dba /dev/raw/raw31
chmod 660 /dev/raw/raw31
raw /dev/raw/raw103 /dev/sdao2
chown oracle:dba /dev/raw/raw103
chmod 660 /dev/raw/raw103
raw /dev/raw/raw175 /dev/sdao3
chown oracle:dba /dev/raw/raw175
chmod 660 /dev/raw/raw175
raw /dev/raw/raw247 /dev/sdao5
chown oracle:dba /dev/raw/raw247
chmod 660 /dev/raw/raw247
raw /dev/raw/raw319 /dev/sdao6
chown oracle:dba /dev/raw/raw319
chmod 660 /dev/raw/raw319
raw /dev/raw/raw391 /dev/sdao7
chown oracle:dba /dev/raw/raw391
chmod 660 /dev/raw/raw391
raw /dev/raw/raw463 /dev/sdao8
chown oracle:dba /dev/raw/raw463
chmod 660 /dev/raw/raw463
raw /dev/raw/raw535 /dev/sdao9
chown oracle:dba /dev/raw/raw535
chmod 660 /dev/raw/raw535
raw /dev/raw/raw607 /dev/sdao10
chown oracle:dba /dev/raw/raw607
chmod 660 /dev/raw/raw607
raw /dev/raw/raw679 /dev/sdao11
chown oracle:dba /dev/raw/raw679
chmod 660 /dev/raw/raw679
raw /dev/raw/raw751 /dev/sdao12
chown oracle:dba /dev/raw/raw751
chmod 660 /dev/raw/raw751
raw /dev/raw/raw823 /dev/sdao13
chown oracle:dba /dev/raw/raw823
chmod 660 /dev/raw/raw823
raw /dev/raw/raw895 /dev/sdao14
chown oracle:dba /dev/raw/raw895
chmod 660 /dev/raw/raw895
raw /dev/raw/raw967 /dev/sdao15
chown oracle:dba /dev/raw/raw967
chmod 660 /dev/raw/raw967
raw /dev/raw/raw32 /dev/sdap1
chown oracle:dba /dev/raw/raw32
chmod 660 /dev/raw/raw32
raw /dev/raw/raw104 /dev/sdap2
chown oracle:dba /dev/raw/raw104
chmod 660 /dev/raw/raw104
raw /dev/raw/raw176 /dev/sdap3
chown oracle:dba /dev/raw/raw176
chmod 660 /dev/raw/raw176
raw /dev/raw/raw248 /dev/sdap5
chown oracle:dba /dev/raw/raw248
chmod 660 /dev/raw/raw248
raw /dev/raw/raw320 /dev/sdap6
chown oracle:dba /dev/raw/raw320
chmod 660 /dev/raw/raw320
raw /dev/raw/raw392 /dev/sdap7
chown oracle:dba /dev/raw/raw392
chmod 660 /dev/raw/raw392
raw /dev/raw/raw464 /dev/sdap8
chown oracle:dba /dev/raw/raw464
chmod 660 /dev/raw/raw464
raw /dev/raw/raw536 /dev/sdap9
chown oracle:dba /dev/raw/raw536
chmod 660 /dev/raw/raw536
raw /dev/raw/raw608 /dev/sdap10
chown oracle:dba /dev/raw/raw608
chmod 660 /dev/raw/raw608
raw /dev/raw/raw680 /dev/sdap11
chown oracle:dba /dev/raw/raw680
chmod 660 /dev/raw/raw680
raw /dev/raw/raw752 /dev/sdap12
chown oracle:dba /dev/raw/raw752
chmod 660 /dev/raw/raw752
raw /dev/raw/raw824 /dev/sdap13
chown oracle:dba /dev/raw/raw824
chmod 660 /dev/raw/raw824
raw /dev/raw/raw896 /dev/sdap14
chown oracle:dba /dev/raw/raw896
chmod 660 /dev/raw/raw896
raw /dev/raw/raw968 /dev/sdap15
chown oracle:dba /dev/raw/raw968
chmod 660 /dev/raw/raw968
raw /dev/raw/raw33 /dev/sdar1
chown oracle:dba /dev/raw/raw33
chmod 660 /dev/raw/raw33
raw /dev/raw/raw105 /dev/sdar2
chown oracle:dba /dev/raw/raw105
chmod 660 /dev/raw/raw105
raw /dev/raw/raw177 /dev/sdar3
chown oracle:dba /dev/raw/raw177
chmod 660 /dev/raw/raw177
raw /dev/raw/raw249 /dev/sdar5
chown oracle:dba /dev/raw/raw249
chmod 660 /dev/raw/raw249
raw /dev/raw/raw321 /dev/sdar6
chown oracle:dba /dev/raw/raw321
```


Copyright © 2004 NEC Corporation

```
chmod 660 /dev/raw/raw321
raw /dev/raw/raw393 /dev/sdar7
chown oracle:dba /dev/raw/raw393
chmod 660 /dev/raw/raw393
raw /dev/raw/raw465 /dev/sdar8
chown oracle:dba /dev/raw/raw465
chmod 660 /dev/raw/raw465
raw /dev/raw/raw537 /dev/sdar9
chown oracle:dba /dev/raw/raw537
chmod 660 /dev/raw/raw537
raw /dev/raw/raw609 /dev/sdar10
chown oracle:dba /dev/raw/raw609
chmod 660 /dev/raw/raw609
raw /dev/raw/raw681 /dev/sdar11
chown oracle:dba /dev/raw/raw681
chmod 660 /dev/raw/raw681
raw /dev/raw/raw753 /dev/sdar12
chown oracle:dba /dev/raw/raw753
chmod 660 /dev/raw/raw753
raw /dev/raw/raw825 /dev/sdar13
chown oracle:dba /dev/raw/raw825
chmod 660 /dev/raw/raw825
raw /dev/raw/raw897 /dev/sdar14
chown oracle:dba /dev/raw/raw897
chmod 660 /dev/raw/raw897
raw /dev/raw/raw969 /dev/sdar15
chown oracle:dba /dev/raw/raw969
chmod 660 /dev/raw/raw969
raw /dev/raw/raw34 /dev/sdas1
chown oracle:dba /dev/raw/raw34
chmod 660 /dev/raw/raw34
raw /dev/raw/raw106 /dev/sdas2
chown oracle:dba /dev/raw/raw106
chmod 660 /dev/raw/raw106
raw /dev/raw/raw178 /dev/sdas3
chown oracle:dba /dev/raw/raw178
chmod 660 /dev/raw/raw178
raw /dev/raw/raw250 /dev/sdas5
chown oracle:dba /dev/raw/raw250
chmod 660 /dev/raw/raw250
raw /dev/raw/raw322 /dev/sdas6
chown oracle:dba /dev/raw/raw322
chmod 660 /dev/raw/raw322
raw /dev/raw/raw394 /dev/sdas7
chown oracle:dba /dev/raw/raw394
chmod 660 /dev/raw/raw394
raw /dev/raw/raw466 /dev/sdas8
chown oracle:dba /dev/raw/raw466
chmod 660 /dev/raw/raw466
raw /dev/raw/raw538 /dev/sdas9
chown oracle:dba /dev/raw/raw538
chmod 660 /dev/raw/raw538
raw /dev/raw/raw610 /dev/sdas10
chown oracle:dba /dev/raw/raw610
chmod 660 /dev/raw/raw610
raw /dev/raw/raw682 /dev/sdas11
chown oracle:dba /dev/raw/raw682
chmod 660 /dev/raw/raw682
raw /dev/raw/raw754 /dev/sdas12
chown oracle:dba /dev/raw/raw754
chmod 660 /dev/raw/raw754
raw /dev/raw/raw826 /dev/sdas13
chown oracle:dba /dev/raw/raw826
chmod 660 /dev/raw/raw826
raw /dev/raw/raw898 /dev/sdas14
chown oracle:dba /dev/raw/raw898
chmod 660 /dev/raw/raw898
raw /dev/raw/raw970 /dev/sdas15
chown oracle:dba /dev/raw/raw970
chmod 660 /dev/raw/raw970
raw /dev/raw/raw35 /dev/sdau1
chown oracle:dba /dev/raw/raw35
chmod 660 /dev/raw/raw35
raw /dev/raw/raw107 /dev/sdau2
chown oracle:dba /dev/raw/raw107
chmod 660 /dev/raw/raw107
raw /dev/raw/raw179 /dev/sdau3
chown oracle:dba /dev/raw/raw179
chmod 660 /dev/raw/raw179
raw /dev/raw/raw251 /dev/sdau5
chown oracle:dba /dev/raw/raw251
chmod 660 /dev/raw/raw251
raw /dev/raw/raw323 /dev/sdau6
chown oracle:dba /dev/raw/raw323
chmod 660 /dev/raw/raw323
raw /dev/raw/raw395 /dev/sdau7
chown oracle:dba /dev/raw/raw395
chmod 660 /dev/raw/raw395
raw /dev/raw/raw467 /dev/sdau8
chown oracle:dba /dev/raw/raw467
chmod 660 /dev/raw/raw467
raw /dev/raw/raw539 /dev/sdau9
chown oracle:dba /dev/raw/raw539
chmod 660 /dev/raw/raw539
raw /dev/raw/raw611 /dev/sdau10
chown oracle:dba /dev/raw/raw611
chmod 660 /dev/raw/raw611
raw /dev/raw/raw683 /dev/sdau11
chown oracle:dba /dev/raw/raw683
chmod 660 /dev/raw/raw683
raw /dev/raw/raw755 /dev/sdau12
chown oracle:dba /dev/raw/raw755
chmod 660 /dev/raw/raw755
raw /dev/raw/raw827 /dev/sdau13
chown oracle:dba /dev/raw/raw827
chmod 660 /dev/raw/raw827
raw /dev/raw/raw899 /dev/sdau14
chown oracle:dba /dev/raw/raw899
chmod 660 /dev/raw/raw899
raw /dev/raw/raw971 /dev/sdau15
chown oracle:dba /dev/raw/raw971
chmod 660 /dev/raw/raw971
```


Copyright © 2004 NEC Corporation

```
raw /dev/raw/raw36 /dev/sdav1
chown oracle:dba /dev/raw/raw36
chmod 660 /dev/raw/raw36
raw /dev/raw/raw108 /dev/sdav2
chown oracle:dba /dev/raw/raw108
chmod 660 /dev/raw/raw108
raw /dev/raw/raw180 /dev/sdav3
chown oracle:dba /dev/raw/raw180
chmod 660 /dev/raw/raw180
raw /dev/raw/raw252 /dev/sdav5
chown oracle:dba /dev/raw/raw252
chmod 660 /dev/raw/raw252
raw /dev/raw/raw324 /dev/sdav6
chown oracle:dba /dev/raw/raw324
chmod 660 /dev/raw/raw324
raw /dev/raw/raw396 /dev/sdav7
chown oracle:dba /dev/raw/raw396
chmod 660 /dev/raw/raw396
raw /dev/raw/raw468 /dev/sdav8
chown oracle:dba /dev/raw/raw468
chmod 660 /dev/raw/raw468
raw /dev/raw/raw540 /dev/sdav9
chown oracle:dba /dev/raw/raw540
chmod 660 /dev/raw/raw540
raw /dev/raw/raw612 /dev/sdav10
chown oracle:dba /dev/raw/raw612
chmod 660 /dev/raw/raw612
raw /dev/raw/raw684 /dev/sdav11
chown oracle:dba /dev/raw/raw684
chmod 660 /dev/raw/raw684
raw /dev/raw/raw756 /dev/sdav12
chown oracle:dba /dev/raw/raw756
chmod 660 /dev/raw/raw756
raw /dev/raw/raw828 /dev/sdav13
chown oracle:dba /dev/raw/raw828
chmod 660 /dev/raw/raw828
raw /dev/raw/raw900 /dev/sdav14
chown oracle:dba /dev/raw/raw900
chmod 660 /dev/raw/raw900
raw /dev/raw/raw972 /dev/sdav15
chown oracle:dba /dev/raw/raw972
chmod 660 /dev/raw/raw972
raw /dev/raw/raw37 /dev/sdax1
chown oracle:dba /dev/raw/raw37
chmod 660 /dev/raw/raw37
raw /dev/raw/raw109 /dev/sdax2
chown oracle:dba /dev/raw/raw109
chmod 660 /dev/raw/raw109
raw /dev/raw/raw181 /dev/sdax3
chown oracle:dba /dev/raw/raw181
chmod 660 /dev/raw/raw181
raw /dev/raw/raw253 /dev/sdax5
chown oracle:dba /dev/raw/raw253
chmod 660 /dev/raw/raw253
raw /dev/raw/raw325 /dev/sdax6
chown oracle:dba /dev/raw/raw325
chmod 660 /dev/raw/raw325
raw /dev/raw/raw397 /dev/sdax7
chown oracle:dba /dev/raw/raw397
chmod 660 /dev/raw/raw397
raw /dev/raw/raw469 /dev/sdax8
chown oracle:dba /dev/raw/raw469
chmod 660 /dev/raw/raw469
raw /dev/raw/raw541 /dev/sdax9
chown oracle:dba /dev/raw/raw541
chmod 660 /dev/raw/raw541
raw /dev/raw/raw613 /dev/sdax10
chown oracle:dba /dev/raw/raw613
chmod 660 /dev/raw/raw613
raw /dev/raw/raw685 /dev/sdax11
chown oracle:dba /dev/raw/raw685
chmod 660 /dev/raw/raw685
raw /dev/raw/raw757 /dev/sdax12
chown oracle:dba /dev/raw/raw757
chmod 660 /dev/raw/raw757
raw /dev/raw/raw829 /dev/sdax13
chown oracle:dba /dev/raw/raw829
chmod 660 /dev/raw/raw829
raw /dev/raw/raw901 /dev/sdax14
chown oracle:dba /dev/raw/raw901
chmod 660 /dev/raw/raw901
raw /dev/raw/raw973 /dev/sdax15
chown oracle:dba /dev/raw/raw973
chmod 660 /dev/raw/raw973
raw /dev/raw/raw38 /dev/sday1
chown oracle:dba /dev/raw/raw38
chmod 660 /dev/raw/raw38
raw /dev/raw/raw110 /dev/sday2
chown oracle:dba /dev/raw/raw110
chmod 660 /dev/raw/raw110
raw /dev/raw/raw182 /dev/sday3
chown oracle:dba /dev/raw/raw182
chmod 660 /dev/raw/raw182
raw /dev/raw/raw254 /dev/sday5
chown oracle:dba /dev/raw/raw254
chmod 660 /dev/raw/raw254
raw /dev/raw/raw326 /dev/sday6
chown oracle:dba /dev/raw/raw326
chmod 660 /dev/raw/raw326
raw /dev/raw/raw398 /dev/sday7
chown oracle:dba /dev/raw/raw398
chmod 660 /dev/raw/raw398
raw /dev/raw/raw470 /dev/sday8
chown oracle:dba /dev/raw/raw470
chmod 660 /dev/raw/raw470
raw /dev/raw/raw542 /dev/sday9
chown oracle:dba /dev/raw/raw542
chmod 660 /dev/raw/raw542
raw /dev/raw/raw614 /dev/sday10
chown oracle:dba /dev/raw/raw614
chmod 660 /dev/raw/raw614
raw /dev/raw/raw686 /dev/sday11
```


Copyright © 2004 NEC Corporation

```
chown oracle:dba /dev/raw/raw686
chmod 660 /dev/raw/raw686
raw /dev/raw/raw758 /dev/sday12
chown oracle:dba /dev/raw/raw758
chmod 660 /dev/raw/raw758
raw /dev/raw/raw830 /dev/sday13
chown oracle:dba /dev/raw/raw830
chmod 660 /dev/raw/raw830
raw /dev/raw/raw902 /dev/sday14
chown oracle:dba /dev/raw/raw902
chmod 660 /dev/raw/raw902
raw /dev/raw/raw974 /dev/sday15
chown oracle:dba /dev/raw/raw974
chmod 660 /dev/raw/raw974
raw /dev/raw/raw3 /dev/sdb1
chown oracle:dba /dev/raw/raw3
chmod 660 /dev/raw/raw3
raw /dev/raw/raw75 /dev/sdb2
chown oracle:dba /dev/raw/raw75
chmod 660 /dev/raw/raw75
raw /dev/raw/raw147 /dev/sdb3
chown oracle:dba /dev/raw/raw147
chmod 660 /dev/raw/raw147
raw /dev/raw/raw219 /dev/sdb5
chown oracle:dba /dev/raw/raw219
chmod 660 /dev/raw/raw219
raw /dev/raw/raw291 /dev/sdb6
chown oracle:dba /dev/raw/raw291
chmod 660 /dev/raw/raw291
raw /dev/raw/raw363 /dev/sdb7
chown oracle:dba /dev/raw/raw363
chmod 660 /dev/raw/raw363
raw /dev/raw/raw435 /dev/sdb8
chown oracle:dba /dev/raw/raw435
chmod 660 /dev/raw/raw435
raw /dev/raw/raw507 /dev/sdb9
chown oracle:dba /dev/raw/raw507
chmod 660 /dev/raw/raw507
raw /dev/raw/raw579 /dev/sdb10
chown oracle:dba /dev/raw/raw579
chmod 660 /dev/raw/raw579
raw /dev/raw/raw651 /dev/sdb11
chown oracle:dba /dev/raw/raw651
chmod 660 /dev/raw/raw651
raw /dev/raw/raw723 /dev/sdb12
chown oracle:dba /dev/raw/raw723
chmod 660 /dev/raw/raw723
raw /dev/raw/raw795 /dev/sdb13
chown oracle:dba /dev/raw/raw795
chmod 660 /dev/raw/raw795
raw /dev/raw/raw867 /dev/sdb14
chown oracle:dba /dev/raw/raw867
chmod 660 /dev/raw/raw867
raw /dev/raw/raw939 /dev/sdb15
chown oracle:dba /dev/raw/raw939
chmod 660 /dev/raw/raw939
raw /dev/raw/raw39 /dev/sdba1
chown oracle:dba /dev/raw/raw39
chmod 660 /dev/raw/raw39
raw /dev/raw/raw111 /dev/sdba2
chown oracle:dba /dev/raw/raw111
chmod 660 /dev/raw/raw111
raw /dev/raw/raw183 /dev/sdba3
chown oracle:dba /dev/raw/raw183
chmod 660 /dev/raw/raw183
raw /dev/raw/raw255 /dev/sdba5
chown oracle:dba /dev/raw/raw255
chmod 660 /dev/raw/raw255
raw /dev/raw/raw327 /dev/sdba6
chown oracle:dba /dev/raw/raw327
chmod 660 /dev/raw/raw327
raw /dev/raw/raw399 /dev/sdba7
chown oracle:dba /dev/raw/raw399
chmod 660 /dev/raw/raw399
raw /dev/raw/raw471 /dev/sdba8
chown oracle:dba /dev/raw/raw471
chmod 660 /dev/raw/raw471
raw /dev/raw/raw543 /dev/sdba9
chown oracle:dba /dev/raw/raw543
chmod 660 /dev/raw/raw543
raw /dev/raw/raw615 /dev/sdba10
chown oracle:dba /dev/raw/raw615
chmod 660 /dev/raw/raw615
raw /dev/raw/raw687 /dev/sdba11
chown oracle:dba /dev/raw/raw687
chmod 660 /dev/raw/raw687
raw /dev/raw/raw759 /dev/sdba12
chown oracle:dba /dev/raw/raw759
chmod 660 /dev/raw/raw759
raw /dev/raw/raw831 /dev/sdba13
chown oracle:dba /dev/raw/raw831
chmod 660 /dev/raw/raw831
raw /dev/raw/raw903 /dev/sdba14
chown oracle:dba /dev/raw/raw903
chmod 660 /dev/raw/raw903
raw /dev/raw/raw975 /dev/sdba15
chown oracle:dba /dev/raw/raw975
chmod 660 /dev/raw/raw975
raw /dev/raw/raw40 /dev/sdbb1
chown oracle:dba /dev/raw/raw40
chmod 660 /dev/raw/raw40
raw /dev/raw/raw112 /dev/sdbb2
chown oracle:dba /dev/raw/raw112
chmod 660 /dev/raw/raw112
raw /dev/raw/raw184 /dev/sdbb3
chown oracle:dba /dev/raw/raw184
chmod 660 /dev/raw/raw184
raw /dev/raw/raw256 /dev/sdbb5
chown oracle:dba /dev/raw/raw256
chmod 660 /dev/raw/raw256
raw /dev/raw/raw328 /dev/sdbb6
chown oracle:dba /dev/raw/raw328
```


Copyright © 2004 NEC Corporation

```
chmod 660 /dev/raw/raw328
raw /dev/raw/raw400 /dev/sdbb7
chown oracle:dba /dev/raw/raw400
chmod 660 /dev/raw/raw400
raw /dev/raw/raw472 /dev/sdbb8
chown oracle:dba /dev/raw/raw472
chmod 660 /dev/raw/raw472
raw /dev/raw/raw544 /dev/sdbb9
chown oracle:dba /dev/raw/raw544
chmod 660 /dev/raw/raw544
raw /dev/raw/raw616 /dev/sdbb10
chown oracle:dba /dev/raw/raw616
chmod 660 /dev/raw/raw616
raw /dev/raw/raw688 /dev/sdbb11
chown oracle:dba /dev/raw/raw688
chmod 660 /dev/raw/raw688
raw /dev/raw/raw760 /dev/sdbb12
chown oracle:dba /dev/raw/raw760
chmod 660 /dev/raw/raw760
raw /dev/raw/raw832 /dev/sdbb13
chown oracle:dba /dev/raw/raw832
chmod 660 /dev/raw/raw832
raw /dev/raw/raw904 /dev/sdbb14
chown oracle:dba /dev/raw/raw904
chmod 660 /dev/raw/raw904
raw /dev/raw/raw976 /dev/sdbb15
chown oracle:dba /dev/raw/raw976
chmod 660 /dev/raw/raw976
raw /dev/raw/raw41 /dev/sdbc1
chown oracle:dba /dev/raw/raw41
chmod 660 /dev/raw/raw41
raw /dev/raw/raw113 /dev/sdbc2
chown oracle:dba /dev/raw/raw113
chmod 660 /dev/raw/raw113
raw /dev/raw/raw185 /dev/sdbc3
chown oracle:dba /dev/raw/raw185
chmod 660 /dev/raw/raw185
raw /dev/raw/raw257 /dev/sdbc5
chown oracle:dba /dev/raw/raw257
chmod 660 /dev/raw/raw257
raw /dev/raw/raw329 /dev/sdbc6
chown oracle:dba /dev/raw/raw329
chmod 660 /dev/raw/raw329
raw /dev/raw/raw401 /dev/sdbc7
chown oracle:dba /dev/raw/raw401
chmod 660 /dev/raw/raw401
raw /dev/raw/raw473 /dev/sdbc8
chown oracle:dba /dev/raw/raw473
chmod 660 /dev/raw/raw473
raw /dev/raw/raw545 /dev/sdbc9
chown oracle:dba /dev/raw/raw545
chmod 660 /dev/raw/raw545
raw /dev/raw/raw617 /dev/sdbc10
chown oracle:dba /dev/raw/raw617
chmod 660 /dev/raw/raw617
raw /dev/raw/raw689 /dev/sdbc11
chown oracle:dba /dev/raw/raw689
chmod 660 /dev/raw/raw689
raw /dev/raw/raw761 /dev/sdbc12
chown oracle:dba /dev/raw/raw761
chmod 660 /dev/raw/raw761
raw /dev/raw/raw833 /dev/sdbc13
chown oracle:dba /dev/raw/raw833
chmod 660 /dev/raw/raw833
raw /dev/raw/raw905 /dev/sdbc14
chown oracle:dba /dev/raw/raw905
chmod 660 /dev/raw/raw905
raw /dev/raw/raw977 /dev/sdbc15
chown oracle:dba /dev/raw/raw977
chmod 660 /dev/raw/raw977
raw /dev/raw/raw42 /dev/sdbd1
chown oracle:dba /dev/raw/raw42
chmod 660 /dev/raw/raw42
raw /dev/raw/raw114 /dev/sdbd2
chown oracle:dba /dev/raw/raw114
chmod 660 /dev/raw/raw114
raw /dev/raw/raw186 /dev/sdbd3
chown oracle:dba /dev/raw/raw186
chmod 660 /dev/raw/raw186
raw /dev/raw/raw258 /dev/sdbd5
chown oracle:dba /dev/raw/raw258
chmod 660 /dev/raw/raw258
raw /dev/raw/raw330 /dev/sdbd6
chown oracle:dba /dev/raw/raw330
chmod 660 /dev/raw/raw330
raw /dev/raw/raw402 /dev/sdbd7
chown oracle:dba /dev/raw/raw402
chmod 660 /dev/raw/raw402
raw /dev/raw/raw474 /dev/sdbd8
chown oracle:dba /dev/raw/raw474
chmod 660 /dev/raw/raw474
raw /dev/raw/raw546 /dev/sdbd9
chown oracle:dba /dev/raw/raw546
chmod 660 /dev/raw/raw546
raw /dev/raw/raw618 /dev/sdbd10
chown oracle:dba /dev/raw/raw618
chmod 660 /dev/raw/raw618
raw /dev/raw/raw690 /dev/sdbd11
chown oracle:dba /dev/raw/raw690
chmod 660 /dev/raw/raw690
raw /dev/raw/raw762 /dev/sdbd12
chown oracle:dba /dev/raw/raw762
chmod 660 /dev/raw/raw762
raw /dev/raw/raw834 /dev/sdbd13
chown oracle:dba /dev/raw/raw834
chmod 660 /dev/raw/raw834
raw /dev/raw/raw906 /dev/sdbd14
chown oracle:dba /dev/raw/raw906
chmod 660 /dev/raw/raw906
raw /dev/raw/raw978 /dev/sdbd15
chown oracle:dba /dev/raw/raw978
chmod 660 /dev/raw/raw978
```


Copyright © 2004 NEC Corporation

```
raw /dev/raw/raw43 /dev/sdbe1
chown oracle:dba /dev/raw/raw43
chmod 660 /dev/raw/raw43
raw /dev/raw/raw115 /dev/sdbe2
chown oracle:dba /dev/raw/raw115
chmod 660 /dev/raw/raw115
raw /dev/raw/raw187 /dev/sdbe3
chown oracle:dba /dev/raw/raw187
chmod 660 /dev/raw/raw187
raw /dev/raw/raw259 /dev/sdbe5
chown oracle:dba /dev/raw/raw259
chmod 660 /dev/raw/raw259
raw /dev/raw/raw331 /dev/sdbe6
chown oracle:dba /dev/raw/raw331
chmod 660 /dev/raw/raw331
raw /dev/raw/raw403 /dev/sdbe7
chown oracle:dba /dev/raw/raw403
chmod 660 /dev/raw/raw403
raw /dev/raw/raw475 /dev/sdbe8
chown oracle:dba /dev/raw/raw475
chmod 660 /dev/raw/raw475
raw /dev/raw/raw547 /dev/sdbe9
chown oracle:dba /dev/raw/raw547
chmod 660 /dev/raw/raw547
raw /dev/raw/raw619 /dev/sdbe10
chown oracle:dba /dev/raw/raw619
chmod 660 /dev/raw/raw619
raw /dev/raw/raw691 /dev/sdbe11
chown oracle:dba /dev/raw/raw691
chmod 660 /dev/raw/raw691
raw /dev/raw/raw763 /dev/sdbe12
chown oracle:dba /dev/raw/raw763
chmod 660 /dev/raw/raw763
raw /dev/raw/raw835 /dev/sdbe13
chown oracle:dba /dev/raw/raw835
chmod 660 /dev/raw/raw835
raw /dev/raw/raw907 /dev/sdbe14
chown oracle:dba /dev/raw/raw907
chmod 660 /dev/raw/raw907
raw /dev/raw/raw979 /dev/sdbe15
chown oracle:dba /dev/raw/raw979
chmod 660 /dev/raw/raw979
raw /dev/raw/raw44 /dev/sdbf1
chown oracle:dba /dev/raw/raw44
chmod 660 /dev/raw/raw44
raw /dev/raw/raw116 /dev/sdbf2
chown oracle:dba /dev/raw/raw116
chmod 660 /dev/raw/raw116
raw /dev/raw/raw188 /dev/sdbf3
chown oracle:dba /dev/raw/raw188
chmod 660 /dev/raw/raw188
raw /dev/raw/raw260 /dev/sdbf5
chown oracle:dba /dev/raw/raw260
chmod 660 /dev/raw/raw260
raw /dev/raw/raw332 /dev/sdbf6
chown oracle:dba /dev/raw/raw332
chmod 660 /dev/raw/raw332
raw /dev/raw/raw404 /dev/sdbf7
chown oracle:dba /dev/raw/raw404
chmod 660 /dev/raw/raw404
raw /dev/raw/raw476 /dev/sdbf8
chown oracle:dba /dev/raw/raw476
chmod 660 /dev/raw/raw476
raw /dev/raw/raw548 /dev/sdbf9
chown oracle:dba /dev/raw/raw548
chmod 660 /dev/raw/raw548
raw /dev/raw/raw620 /dev/sdbf10
chown oracle:dba /dev/raw/raw620
chmod 660 /dev/raw/raw620
raw /dev/raw/raw692 /dev/sdbf11
chown oracle:dba /dev/raw/raw692
chmod 660 /dev/raw/raw692
raw /dev/raw/raw764 /dev/sdbf12
chown oracle:dba /dev/raw/raw764
chmod 660 /dev/raw/raw764
raw /dev/raw/raw836 /dev/sdbf13
chown oracle:dba /dev/raw/raw836
chmod 660 /dev/raw/raw836
raw /dev/raw/raw908 /dev/sdbf14
chown oracle:dba /dev/raw/raw908
chmod 660 /dev/raw/raw908
raw /dev/raw/raw980 /dev/sdbf15
chown oracle:dba /dev/raw/raw980
chmod 660 /dev/raw/raw980
raw /dev/raw/raw45 /dev/sdbh1
chown oracle:dba /dev/raw/raw45
chmod 660 /dev/raw/raw45
raw /dev/raw/raw117 /dev/sdbh2
chown oracle:dba /dev/raw/raw117
chmod 660 /dev/raw/raw117
raw /dev/raw/raw189 /dev/sdbh3
chown oracle:dba /dev/raw/raw189
chmod 660 /dev/raw/raw189
raw /dev/raw/raw261 /dev/sdbh5
chown oracle:dba /dev/raw/raw261
chmod 660 /dev/raw/raw261
raw /dev/raw/raw333 /dev/sdbh6
chown oracle:dba /dev/raw/raw333
chmod 660 /dev/raw/raw333
raw /dev/raw/raw405 /dev/sdbh7
chown oracle:dba /dev/raw/raw405
chmod 660 /dev/raw/raw405
raw /dev/raw/raw477 /dev/sdbh8
chown oracle:dba /dev/raw/raw477
chmod 660 /dev/raw/raw477
raw /dev/raw/raw549 /dev/sdbh9
chown oracle:dba /dev/raw/raw549
chmod 660 /dev/raw/raw549
raw /dev/raw/raw621 /dev/sdbh10
chown oracle:dba /dev/raw/raw621
chmod 660 /dev/raw/raw621
raw /dev/raw/raw693 /dev/sdbh11
```


Copyright © 2004 NEC Corporation

```
chown oracle:dba /dev/raw/raw693
chmod 660 /dev/raw/raw693
raw /dev/raw/raw765 /dev/sdbh12
chown oracle:dba /dev/raw/raw765
chmod 660 /dev/raw/raw765
raw /dev/raw/raw837 /dev/sdbh13
chown oracle:dba /dev/raw/raw837
chmod 660 /dev/raw/raw837
raw /dev/raw/raw909 /dev/sdbh14
chown oracle:dba /dev/raw/raw909
chmod 660 /dev/raw/raw909
raw /dev/raw/raw981 /dev/sdbh15
chown oracle:dba /dev/raw/raw981
chmod 660 /dev/raw/raw981
raw /dev/raw/raw46 /dev/sdbi1
chown oracle:dba /dev/raw/raw46
chmod 660 /dev/raw/raw46
raw /dev/raw/raw118 /dev/sdbi2
chown oracle:dba /dev/raw/raw118
chmod 660 /dev/raw/raw118
raw /dev/raw/raw190 /dev/sdbi3
chown oracle:dba /dev/raw/raw190
chmod 660 /dev/raw/raw190
raw /dev/raw/raw262 /dev/sdbi5
chown oracle:dba /dev/raw/raw262
chmod 660 /dev/raw/raw262
raw /dev/raw/raw334 /dev/sdbi6
chown oracle:dba /dev/raw/raw334
chmod 660 /dev/raw/raw334
raw /dev/raw/raw406 /dev/sdbi7
chown oracle:dba /dev/raw/raw406
chmod 660 /dev/raw/raw406
raw /dev/raw/raw478 /dev/sdbi8
chown oracle:dba /dev/raw/raw478
chmod 660 /dev/raw/raw478
raw /dev/raw/raw550 /dev/sdbi9
chown oracle:dba /dev/raw/raw550
chmod 660 /dev/raw/raw550
raw /dev/raw/raw622 /dev/sdbi10
chown oracle:dba /dev/raw/raw622
chmod 660 /dev/raw/raw622
raw /dev/raw/raw694 /dev/sdbi11
chown oracle:dba /dev/raw/raw694
chmod 660 /dev/raw/raw694
raw /dev/raw/raw766 /dev/sdbi12
chown oracle:dba /dev/raw/raw766
chmod 660 /dev/raw/raw766
raw /dev/raw/raw838 /dev/sdbi13
chown oracle:dba /dev/raw/raw838
chmod 660 /dev/raw/raw838
raw /dev/raw/raw910 /dev/sdbi14
chown oracle:dba /dev/raw/raw910
chmod 660 /dev/raw/raw910
raw /dev/raw/raw982 /dev/sdbi15
chown oracle:dba /dev/raw/raw982
chmod 660 /dev/raw/raw982
raw /dev/raw/raw47 /dev/sdbk1
chown oracle:dba /dev/raw/raw47
chmod 660 /dev/raw/raw47
raw /dev/raw/raw119 /dev/sdbk2
chown oracle:dba /dev/raw/raw119
chmod 660 /dev/raw/raw119
raw /dev/raw/raw191 /dev/sdbk3
chown oracle:dba /dev/raw/raw191
chmod 660 /dev/raw/raw191
raw /dev/raw/raw263 /dev/sdbk5
chown oracle:dba /dev/raw/raw263
chmod 660 /dev/raw/raw263
raw /dev/raw/raw335 /dev/sdbk6
chown oracle:dba /dev/raw/raw335
chmod 660 /dev/raw/raw335
raw /dev/raw/raw407 /dev/sdbk7
chown oracle:dba /dev/raw/raw407
chmod 660 /dev/raw/raw407
raw /dev/raw/raw479 /dev/sdbk8
chown oracle:dba /dev/raw/raw479
chmod 660 /dev/raw/raw479
raw /dev/raw/raw551 /dev/sdbk9
chown oracle:dba /dev/raw/raw551
chmod 660 /dev/raw/raw551
raw /dev/raw/raw623 /dev/sdbk10
chown oracle:dba /dev/raw/raw623
chmod 660 /dev/raw/raw623
raw /dev/raw/raw695 /dev/sdbk11
chown oracle:dba /dev/raw/raw695
chmod 660 /dev/raw/raw695
raw /dev/raw/raw767 /dev/sdbk12
chown oracle:dba /dev/raw/raw767
chmod 660 /dev/raw/raw767
raw /dev/raw/raw839 /dev/sdbk13
chown oracle:dba /dev/raw/raw839
chmod 660 /dev/raw/raw839
raw /dev/raw/raw911 /dev/sdbk14
chown oracle:dba /dev/raw/raw911
chmod 660 /dev/raw/raw911
raw /dev/raw/raw983 /dev/sdbk15
chown oracle:dba /dev/raw/raw983
chmod 660 /dev/raw/raw983
raw /dev/raw/raw48 /dev/sdbl1
chown oracle:dba /dev/raw/raw48
chmod 660 /dev/raw/raw48
raw /dev/raw/raw120 /dev/sdbl2
chown oracle:dba /dev/raw/raw120
chmod 660 /dev/raw/raw120
raw /dev/raw/raw192 /dev/sdbl3
chown oracle:dba /dev/raw/raw192
chmod 660 /dev/raw/raw192
raw /dev/raw/raw264 /dev/sdbl5
chown oracle:dba /dev/raw/raw264
chmod 660 /dev/raw/raw264
raw /dev/raw/raw336 /dev/sdbl6
chown oracle:dba /dev/raw/raw336
```


Copyright © 2004 NEC Corporation

```
chmod 660 /dev/raw/raw336
raw /dev/raw/raw408 /dev/sdbl7
chown oracle:dba /dev/raw/raw408
chmod 660 /dev/raw/raw408
raw /dev/raw/raw480 /dev/sdbl8
chown oracle:dba /dev/raw/raw480
chmod 660 /dev/raw/raw480
raw /dev/raw/raw552 /dev/sdbl9
chown oracle:dba /dev/raw/raw552
chmod 660 /dev/raw/raw552
raw /dev/raw/raw624 /dev/sdbl10
chown oracle:dba /dev/raw/raw624
chmod 660 /dev/raw/raw624
raw /dev/raw/raw696 /dev/sdbl11
chown oracle:dba /dev/raw/raw696
chmod 660 /dev/raw/raw696
raw /dev/raw/raw768 /dev/sdbl12
chown oracle:dba /dev/raw/raw768
chmod 660 /dev/raw/raw768
raw /dev/raw/raw840 /dev/sdbl13
chown oracle:dba /dev/raw/raw840
chmod 660 /dev/raw/raw840
raw /dev/raw/raw912 /dev/sdbl14
chown oracle:dba /dev/raw/raw912
chmod 660 /dev/raw/raw912
raw /dev/raw/raw984 /dev/sdbl15
chown oracle:dba /dev/raw/raw984
chmod 660 /dev/raw/raw984
raw /dev/raw/raw49 /dev/sdbn1
chown oracle:dba /dev/raw/raw49
chmod 660 /dev/raw/raw49
raw /dev/raw/raw121 /dev/sdbn2
chown oracle:dba /dev/raw/raw121
chmod 660 /dev/raw/raw121
raw /dev/raw/raw193 /dev/sdbn3
chown oracle:dba /dev/raw/raw193
chmod 660 /dev/raw/raw193
raw /dev/raw/raw265 /dev/sdbn5
chown oracle:dba /dev/raw/raw265
chmod 660 /dev/raw/raw265
raw /dev/raw/raw337 /dev/sdbn6
chown oracle:dba /dev/raw/raw337
chmod 660 /dev/raw/raw337
raw /dev/raw/raw409 /dev/sdbn7
chown oracle:dba /dev/raw/raw409
chmod 660 /dev/raw/raw409
raw /dev/raw/raw481 /dev/sdbn8
chown oracle:dba /dev/raw/raw481
chmod 660 /dev/raw/raw481
raw /dev/raw/raw553 /dev/sdbn9
chown oracle:dba /dev/raw/raw553
chmod 660 /dev/raw/raw553
raw /dev/raw/raw625 /dev/sdbn10
chown oracle:dba /dev/raw/raw625
chmod 660 /dev/raw/raw625
raw /dev/raw/raw697 /dev/sdbn11
chown oracle:dba /dev/raw/raw697
chmod 660 /dev/raw/raw697
raw /dev/raw/raw769 /dev/sdbn12
chown oracle:dba /dev/raw/raw769
chmod 660 /dev/raw/raw769
raw /dev/raw/raw841 /dev/sdbn13
chown oracle:dba /dev/raw/raw841
chmod 660 /dev/raw/raw841
raw /dev/raw/raw913 /dev/sdbn14
chown oracle:dba /dev/raw/raw913
chmod 660 /dev/raw/raw913
raw /dev/raw/raw985 /dev/sdbn15
chown oracle:dba /dev/raw/raw985
chmod 660 /dev/raw/raw985
raw /dev/raw/raw50 /dev/sdbo1
chown oracle:dba /dev/raw/raw50
chmod 660 /dev/raw/raw50
raw /dev/raw/raw122 /dev/sdbo2
chown oracle:dba /dev/raw/raw122
chmod 660 /dev/raw/raw122
raw /dev/raw/raw194 /dev/sdbo3
chown oracle:dba /dev/raw/raw194
chmod 660 /dev/raw/raw194
raw /dev/raw/raw266 /dev/sdbo5
chown oracle:dba /dev/raw/raw266
chmod 660 /dev/raw/raw266
raw /dev/raw/raw338 /dev/sdbo6
chown oracle:dba /dev/raw/raw338
chmod 660 /dev/raw/raw338
raw /dev/raw/raw410 /dev/sdbo7
chown oracle:dba /dev/raw/raw410
chmod 660 /dev/raw/raw410
raw /dev/raw/raw482 /dev/sdbo8
chown oracle:dba /dev/raw/raw482
chmod 660 /dev/raw/raw482
raw /dev/raw/raw554 /dev/sdbo9
chown oracle:dba /dev/raw/raw554
chmod 660 /dev/raw/raw554
raw /dev/raw/raw626 /dev/sdbo10
chown oracle:dba /dev/raw/raw626
chmod 660 /dev/raw/raw626
raw /dev/raw/raw698 /dev/sdbo11
chown oracle:dba /dev/raw/raw698
chmod 660 /dev/raw/raw698
raw /dev/raw/raw770 /dev/sdbo12
chown oracle:dba /dev/raw/raw770
chmod 660 /dev/raw/raw770
raw /dev/raw/raw842 /dev/sdbo13
chown oracle:dba /dev/raw/raw842
chmod 660 /dev/raw/raw842
raw /dev/raw/raw914 /dev/sdbo14
chown oracle:dba /dev/raw/raw914
chmod 660 /dev/raw/raw914
raw /dev/raw/raw986 /dev/sdbo15
chown oracle:dba /dev/raw/raw986
chmod 660 /dev/raw/raw986
```


Copyright © 2004 NEC Corporation

```
raw /dev/raw/raw51 /dev/sdbq1
chown oracle:dba /dev/raw/raw51
chmod 660 /dev/raw/raw51
raw /dev/raw/raw123 /dev/sdbq2
chown oracle:dba /dev/raw/raw123
chmod 660 /dev/raw/raw123
raw /dev/raw/raw195 /dev/sdbq3
chown oracle:dba /dev/raw/raw195
chmod 660 /dev/raw/raw195
raw /dev/raw/raw267 /dev/sdbq5
chown oracle:dba /dev/raw/raw267
chmod 660 /dev/raw/raw267
raw /dev/raw/raw339 /dev/sdbq6
chown oracle:dba /dev/raw/raw339
chmod 660 /dev/raw/raw339
raw /dev/raw/raw411 /dev/sdbq7
chown oracle:dba /dev/raw/raw411
chmod 660 /dev/raw/raw411
raw /dev/raw/raw483 /dev/sdbq8
chown oracle:dba /dev/raw/raw483
chmod 660 /dev/raw/raw483
raw /dev/raw/raw555 /dev/sdbq9
chown oracle:dba /dev/raw/raw555
chmod 660 /dev/raw/raw555
raw /dev/raw/raw627 /dev/sdbq10
chown oracle:dba /dev/raw/raw627
chmod 660 /dev/raw/raw627
raw /dev/raw/raw699 /dev/sdbq11
chown oracle:dba /dev/raw/raw699
chmod 660 /dev/raw/raw699
raw /dev/raw/raw771 /dev/sdbq12
chown oracle:dba /dev/raw/raw771
chmod 660 /dev/raw/raw771
raw /dev/raw/raw843 /dev/sdbq13
chown oracle:dba /dev/raw/raw843
chmod 660 /dev/raw/raw843
raw /dev/raw/raw915 /dev/sdbq14
chown oracle:dba /dev/raw/raw915
chmod 660 /dev/raw/raw915
raw /dev/raw/raw987 /dev/sdbq15
chown oracle:dba /dev/raw/raw987
chmod 660 /dev/raw/raw987
raw /dev/raw/raw52 /dev/sdbr1
chown oracle:dba /dev/raw/raw52
chmod 660 /dev/raw/raw52
raw /dev/raw/raw124 /dev/sdbr2
chown oracle:dba /dev/raw/raw124
chmod 660 /dev/raw/raw124
raw /dev/raw/raw196 /dev/sdbr3
chown oracle:dba /dev/raw/raw196
chmod 660 /dev/raw/raw196
raw /dev/raw/raw268 /dev/sdbr5
chown oracle:dba /dev/raw/raw268
chmod 660 /dev/raw/raw268
raw /dev/raw/raw340 /dev/sdbr6
chown oracle:dba /dev/raw/raw340
chmod 660 /dev/raw/raw340
raw /dev/raw/raw412 /dev/sdbr7
chown oracle:dba /dev/raw/raw412
chmod 660 /dev/raw/raw412
raw /dev/raw/raw484 /dev/sdbr8
chown oracle:dba /dev/raw/raw484
chmod 660 /dev/raw/raw484
raw /dev/raw/raw556 /dev/sdbr9
chown oracle:dba /dev/raw/raw556
chmod 660 /dev/raw/raw556
raw /dev/raw/raw628 /dev/sdbr10
chown oracle:dba /dev/raw/raw628
chmod 660 /dev/raw/raw628
raw /dev/raw/raw700 /dev/sdbr11
chown oracle:dba /dev/raw/raw700
chmod 660 /dev/raw/raw700
raw /dev/raw/raw772 /dev/sdbr12
chown oracle:dba /dev/raw/raw772
chmod 660 /dev/raw/raw772
raw /dev/raw/raw844 /dev/sdbr13
chown oracle:dba /dev/raw/raw844
chmod 660 /dev/raw/raw844
raw /dev/raw/raw916 /dev/sdbr14
chown oracle:dba /dev/raw/raw916
chmod 660 /dev/raw/raw916
raw /dev/raw/raw988 /dev/sdbr15
chown oracle:dba /dev/raw/raw988
chmod 660 /dev/raw/raw988
raw /dev/raw/raw53 /dev/sdbt1
chown oracle:dba /dev/raw/raw53
chmod 660 /dev/raw/raw53
raw /dev/raw/raw125 /dev/sdbt2
chown oracle:dba /dev/raw/raw125
chmod 660 /dev/raw/raw125
raw /dev/raw/raw197 /dev/sdbt3
chown oracle:dba /dev/raw/raw197
chmod 660 /dev/raw/raw197
raw /dev/raw/raw269 /dev/sdbt5
chown oracle:dba /dev/raw/raw269
chmod 660 /dev/raw/raw269
raw /dev/raw/raw341 /dev/sdbt6
chown oracle:dba /dev/raw/raw341
chmod 660 /dev/raw/raw341
raw /dev/raw/raw413 /dev/sdbt7
chown oracle:dba /dev/raw/raw413
chmod 660 /dev/raw/raw413
raw /dev/raw/raw485 /dev/sdbt8
chown oracle:dba /dev/raw/raw485
chmod 660 /dev/raw/raw485
raw /dev/raw/raw557 /dev/sdbt9
chown oracle:dba /dev/raw/raw557
chmod 660 /dev/raw/raw557
raw /dev/raw/raw629 /dev/sdbt10
chown oracle:dba /dev/raw/raw629
chmod 660 /dev/raw/raw629
raw /dev/raw/raw701 /dev/sdbt11
```


Copyright © 2004 NEC Corporation

```
chown oracle:dba /dev/raw/raw701
chmod 660 /dev/raw/raw701
raw /dev/raw/raw773 /dev/sdbt12
chown oracle:dba /dev/raw/raw773
chmod 660 /dev/raw/raw773
raw /dev/raw/raw845 /dev/sdbt13
chown oracle:dba /dev/raw/raw845
chmod 660 /dev/raw/raw845
raw /dev/raw/raw917 /dev/sdbt14
chown oracle:dba /dev/raw/raw917
chmod 660 /dev/raw/raw917
raw /dev/raw/raw989 /dev/sdbt15
chown oracle:dba /dev/raw/raw989
chmod 660 /dev/raw/raw989
raw /dev/raw/raw54 /dev/sdbu1
chown oracle:dba /dev/raw/raw54
chmod 660 /dev/raw/raw54
raw /dev/raw/raw126 /dev/sdbu2
chown oracle:dba /dev/raw/raw126
chmod 660 /dev/raw/raw126
raw /dev/raw/raw198 /dev/sdbu3
chown oracle:dba /dev/raw/raw198
chmod 660 /dev/raw/raw198
raw /dev/raw/raw270 /dev/sdbu5
chown oracle:dba /dev/raw/raw270
chmod 660 /dev/raw/raw270
raw /dev/raw/raw342 /dev/sdbu6
chown oracle:dba /dev/raw/raw342
chmod 660 /dev/raw/raw342
raw /dev/raw/raw414 /dev/sdbu7
chown oracle:dba /dev/raw/raw414
chmod 660 /dev/raw/raw414
raw /dev/raw/raw486 /dev/sdbu8
chown oracle:dba /dev/raw/raw486
chmod 660 /dev/raw/raw486
raw /dev/raw/raw558 /dev/sdbu9
chown oracle:dba /dev/raw/raw558
chmod 660 /dev/raw/raw558
raw /dev/raw/raw630 /dev/sdbu10
chown oracle:dba /dev/raw/raw630
chmod 660 /dev/raw/raw630
raw /dev/raw/raw702 /dev/sdbu11
chown oracle:dba /dev/raw/raw702
chmod 660 /dev/raw/raw702
raw /dev/raw/raw774 /dev/sdbu12
chown oracle:dba /dev/raw/raw774
chmod 660 /dev/raw/raw774
raw /dev/raw/raw846 /dev/sdbu13
chown oracle:dba /dev/raw/raw846
chmod 660 /dev/raw/raw846
raw /dev/raw/raw918 /dev/sdbu14
chown oracle:dba /dev/raw/raw918
chmod 660 /dev/raw/raw918
raw /dev/raw/raw990 /dev/sdbu15
chown oracle:dba /dev/raw/raw990
chmod 660 /dev/raw/raw990
raw /dev/raw/raw55 /dev/sdbw1
chown oracle:dba /dev/raw/raw55
chmod 660 /dev/raw/raw55
raw /dev/raw/raw127 /dev/sdbw2
chown oracle:dba /dev/raw/raw127
chmod 660 /dev/raw/raw127
raw /dev/raw/raw199 /dev/sdbw3
chown oracle:dba /dev/raw/raw199
chmod 660 /dev/raw/raw199
raw /dev/raw/raw271 /dev/sdbw5
chown oracle:dba /dev/raw/raw271
chmod 660 /dev/raw/raw271
raw /dev/raw/raw343 /dev/sdbw6
chown oracle:dba /dev/raw/raw343
chmod 660 /dev/raw/raw343
raw /dev/raw/raw415 /dev/sdbw7
chown oracle:dba /dev/raw/raw415
chmod 660 /dev/raw/raw415
raw /dev/raw/raw487 /dev/sdbw8
chown oracle:dba /dev/raw/raw487
chmod 660 /dev/raw/raw487
raw /dev/raw/raw559 /dev/sdbw9
chown oracle:dba /dev/raw/raw559
chmod 660 /dev/raw/raw559
raw /dev/raw/raw631 /dev/sdbw10
chown oracle:dba /dev/raw/raw631
chmod 660 /dev/raw/raw631
raw /dev/raw/raw703 /dev/sdbw11
chown oracle:dba /dev/raw/raw703
chmod 660 /dev/raw/raw703
raw /dev/raw/raw775 /dev/sdbw12
chown oracle:dba /dev/raw/raw775
chmod 660 /dev/raw/raw775
raw /dev/raw/raw847 /dev/sdbw13
chown oracle:dba /dev/raw/raw847
chmod 660 /dev/raw/raw847
raw /dev/raw/raw919 /dev/sdbw14
chown oracle:dba /dev/raw/raw919
chmod 660 /dev/raw/raw919
raw /dev/raw/raw991 /dev/sdbw15
chown oracle:dba /dev/raw/raw991
chmod 660 /dev/raw/raw991
raw /dev/raw/raw56 /dev/sdbx1
chown oracle:dba /dev/raw/raw56
chmod 660 /dev/raw/raw56
raw /dev/raw/raw128 /dev/sdbx2
chown oracle:dba /dev/raw/raw128
chmod 660 /dev/raw/raw128
raw /dev/raw/raw200 /dev/sdbx3
chown oracle:dba /dev/raw/raw200
chmod 660 /dev/raw/raw200
raw /dev/raw/raw272 /dev/sdbx5
chown oracle:dba /dev/raw/raw272
chmod 660 /dev/raw/raw272
raw /dev/raw/raw344 /dev/sdbx6
chown oracle:dba /dev/raw/raw344
```


Copyright © 2004 NEC Corporation

```
chmod 660 /dev/raw/raw344
raw /dev/raw/raw416 /dev/sdbx7
chown oracle:dba /dev/raw/raw416
chmod 660 /dev/raw/raw416
raw /dev/raw/raw488 /dev/sdbx8
chown oracle:dba /dev/raw/raw488
chmod 660 /dev/raw/raw488
raw /dev/raw/raw560 /dev/sdbx9
chown oracle:dba /dev/raw/raw560
chmod 660 /dev/raw/raw560
raw /dev/raw/raw632 /dev/sdbx10
chown oracle:dba /dev/raw/raw632
chmod 660 /dev/raw/raw632
raw /dev/raw/raw704 /dev/sdbx11
chown oracle:dba /dev/raw/raw704
chmod 660 /dev/raw/raw704
raw /dev/raw/raw776 /dev/sdbx12
chown oracle:dba /dev/raw/raw776
chmod 660 /dev/raw/raw776
raw /dev/raw/raw848 /dev/sdbx13
chown oracle:dba /dev/raw/raw848
chmod 660 /dev/raw/raw848
raw /dev/raw/raw920 /dev/sdbx14
chown oracle:dba /dev/raw/raw920
chmod 660 /dev/raw/raw920
raw /dev/raw/raw992 /dev/sdbx15
chown oracle:dba /dev/raw/raw992
chmod 660 /dev/raw/raw992
raw /dev/raw/raw57 /dev/sdbz1
chown oracle:dba /dev/raw/raw57
chmod 660 /dev/raw/raw57
raw /dev/raw/raw129 /dev/sdbz2
chown oracle:dba /dev/raw/raw129
chmod 660 /dev/raw/raw129
raw /dev/raw/raw201 /dev/sdbz3
chown oracle:dba /dev/raw/raw201
chmod 660 /dev/raw/raw201
raw /dev/raw/raw273 /dev/sdbz5
chown oracle:dba /dev/raw/raw273
chmod 660 /dev/raw/raw273
raw /dev/raw/raw345 /dev/sdbz6
chown oracle:dba /dev/raw/raw345
chmod 660 /dev/raw/raw345
raw /dev/raw/raw417 /dev/sdbz7
chown oracle:dba /dev/raw/raw417
chmod 660 /dev/raw/raw417
raw /dev/raw/raw489 /dev/sdbz8
chown oracle:dba /dev/raw/raw489
chmod 660 /dev/raw/raw489
raw /dev/raw/raw561 /dev/sdbz9
chown oracle:dba /dev/raw/raw561
chmod 660 /dev/raw/raw561
raw /dev/raw/raw633 /dev/sdbz10
chown oracle:dba /dev/raw/raw633
chmod 660 /dev/raw/raw633
raw /dev/raw/raw705 /dev/sdbz11
chown oracle:dba /dev/raw/raw705
chmod 660 /dev/raw/raw705
raw /dev/raw/raw777 /dev/sdbz12
chown oracle:dba /dev/raw/raw777
chmod 660 /dev/raw/raw777
raw /dev/raw/raw849 /dev/sdbz13
chown oracle:dba /dev/raw/raw849
chmod 660 /dev/raw/raw849
raw /dev/raw/raw921 /dev/sdbz14
chown oracle:dba /dev/raw/raw921
chmod 660 /dev/raw/raw921
raw /dev/raw/raw993 /dev/sdbz15
chown oracle:dba /dev/raw/raw993
chmod 660 /dev/raw/raw993
raw /dev/raw/raw4 /dev/sdc1
chown oracle:dba /dev/raw/raw4
chmod 660 /dev/raw/raw4
raw /dev/raw/raw76 /dev/sdc2
chown oracle:dba /dev/raw/raw76
chmod 660 /dev/raw/raw76
raw /dev/raw/raw148 /dev/sdc3
chown oracle:dba /dev/raw/raw148
chmod 660 /dev/raw/raw148
raw /dev/raw/raw220 /dev/sdc5
chown oracle:dba /dev/raw/raw220
chmod 660 /dev/raw/raw220
raw /dev/raw/raw292 /dev/sdc6
chown oracle:dba /dev/raw/raw292
chmod 660 /dev/raw/raw292
raw /dev/raw/raw364 /dev/sdc7
chown oracle:dba /dev/raw/raw364
chmod 660 /dev/raw/raw364
raw /dev/raw/raw436 /dev/sdc8
chown oracle:dba /dev/raw/raw436
chmod 660 /dev/raw/raw436
raw /dev/raw/raw508 /dev/sdc9
chown oracle:dba /dev/raw/raw508
chmod 660 /dev/raw/raw508
raw /dev/raw/raw580 /dev/sdc10
chown oracle:dba /dev/raw/raw580
chmod 660 /dev/raw/raw580
raw /dev/raw/raw652 /dev/sdc11
chown oracle:dba /dev/raw/raw652
chmod 660 /dev/raw/raw652
raw /dev/raw/raw724 /dev/sdc12
chown oracle:dba /dev/raw/raw724
chmod 660 /dev/raw/raw724
raw /dev/raw/raw796 /dev/sdc13
chown oracle:dba /dev/raw/raw796
chmod 660 /dev/raw/raw796
raw /dev/raw/raw868 /dev/sdc14
chown oracle:dba /dev/raw/raw868
chmod 660 /dev/raw/raw868
raw /dev/raw/raw940 /dev/sdc15
chown oracle:dba /dev/raw/raw940
chmod 660 /dev/raw/raw940
```


Copyright © 2004 NEC Corporation

```
raw /dev/raw/raw58 /dev/sdca1
chown oracle:dba /dev/raw/raw58
chmod 660 /dev/raw/raw58
raw /dev/raw/raw130 /dev/sdca2
chown oracle:dba /dev/raw/raw130
chmod 660 /dev/raw/raw130
raw /dev/raw/raw202 /dev/sdca3
chown oracle:dba /dev/raw/raw202
chmod 660 /dev/raw/raw202
raw /dev/raw/raw274 /dev/sdca5
chown oracle:dba /dev/raw/raw274
chmod 660 /dev/raw/raw274
raw /dev/raw/raw346 /dev/sdca6
chown oracle:dba /dev/raw/raw346
chmod 660 /dev/raw/raw346
raw /dev/raw/raw418 /dev/sdca7
chown oracle:dba /dev/raw/raw418
chmod 660 /dev/raw/raw418
raw /dev/raw/raw490 /dev/sdca8
chown oracle:dba /dev/raw/raw490
chmod 660 /dev/raw/raw490
raw /dev/raw/raw562 /dev/sdca9
chown oracle:dba /dev/raw/raw562
chmod 660 /dev/raw/raw562
raw /dev/raw/raw634 /dev/sdca10
chown oracle:dba /dev/raw/raw634
chmod 660 /dev/raw/raw634
raw /dev/raw/raw706 /dev/sdca11
chown oracle:dba /dev/raw/raw706
chmod 660 /dev/raw/raw706
raw /dev/raw/raw778 /dev/sdca12
chown oracle:dba /dev/raw/raw778
chmod 660 /dev/raw/raw778
raw /dev/raw/raw850 /dev/sdca13
chown oracle:dba /dev/raw/raw850
chmod 660 /dev/raw/raw850
raw /dev/raw/raw922 /dev/sdca14
chown oracle:dba /dev/raw/raw922
chmod 660 /dev/raw/raw922
raw /dev/raw/raw994 /dev/sdca15
chown oracle:dba /dev/raw/raw994
chmod 660 /dev/raw/raw994
raw /dev/raw/raw59 /dev/sdcc1
chown oracle:dba /dev/raw/raw59
chmod 660 /dev/raw/raw59
raw /dev/raw/raw131 /dev/sdcc2
chown oracle:dba /dev/raw/raw131
chmod 660 /dev/raw/raw131
raw /dev/raw/raw203 /dev/sdcc3
chown oracle:dba /dev/raw/raw203
chmod 660 /dev/raw/raw203
raw /dev/raw/raw275 /dev/sdcc5
chown oracle:dba /dev/raw/raw275
chmod 660 /dev/raw/raw275
raw /dev/raw/raw347 /dev/sdcc6
chown oracle:dba /dev/raw/raw347
chmod 660 /dev/raw/raw347
raw /dev/raw/raw419 /dev/sdcc7
chown oracle:dba /dev/raw/raw419
chmod 660 /dev/raw/raw419
raw /dev/raw/raw491 /dev/sdcc8
chown oracle:dba /dev/raw/raw491
chmod 660 /dev/raw/raw491
raw /dev/raw/raw563 /dev/sdcc9
chown oracle:dba /dev/raw/raw563
chmod 660 /dev/raw/raw563
raw /dev/raw/raw635 /dev/sdcc10
chown oracle:dba /dev/raw/raw635
chmod 660 /dev/raw/raw635
raw /dev/raw/raw707 /dev/sdcc11
chown oracle:dba /dev/raw/raw707
chmod 660 /dev/raw/raw707
raw /dev/raw/raw779 /dev/sdcc12
chown oracle:dba /dev/raw/raw779
chmod 660 /dev/raw/raw779
raw /dev/raw/raw851 /dev/sdcc13
chown oracle:dba /dev/raw/raw851
chmod 660 /dev/raw/raw851
raw /dev/raw/raw923 /dev/sdcc14
chown oracle:dba /dev/raw/raw923
chmod 660 /dev/raw/raw923
raw /dev/raw/raw995 /dev/sdcc15
chown oracle:dba /dev/raw/raw995
chmod 660 /dev/raw/raw995
raw /dev/raw/raw60 /dev/sdcd1
chown oracle:dba /dev/raw/raw60
chmod 660 /dev/raw/raw60
raw /dev/raw/raw132 /dev/sdcd2
chown oracle:dba /dev/raw/raw132
chmod 660 /dev/raw/raw132
raw /dev/raw/raw204 /dev/sdcd3
chown oracle:dba /dev/raw/raw204
chmod 660 /dev/raw/raw204
raw /dev/raw/raw276 /dev/sdcd5
chown oracle:dba /dev/raw/raw276
chmod 660 /dev/raw/raw276
raw /dev/raw/raw348 /dev/sdcd6
chown oracle:dba /dev/raw/raw348
chmod 660 /dev/raw/raw348
raw /dev/raw/raw420 /dev/sdcd7
chown oracle:dba /dev/raw/raw420
chmod 660 /dev/raw/raw420
raw /dev/raw/raw492 /dev/sdcd8
chown oracle:dba /dev/raw/raw492
chmod 660 /dev/raw/raw492
raw /dev/raw/raw564 /dev/sdcd9
chown oracle:dba /dev/raw/raw564
chmod 660 /dev/raw/raw564
raw /dev/raw/raw636 /dev/sdcd10
chown oracle:dba /dev/raw/raw636
chmod 660 /dev/raw/raw636
raw /dev/raw/raw708 /dev/sdcd11
```


Copyright © 2004 NEC Corporation

```
chown oracle:dba /dev/raw/raw708
chmod 660 /dev/raw/raw708
raw /dev/raw/raw780 /dev/sdcd12
chown oracle:dba /dev/raw/raw780
chmod 660 /dev/raw/raw780
raw /dev/raw/raw852 /dev/sdcd13
chown oracle:dba /dev/raw/raw852
chmod 660 /dev/raw/raw852
raw /dev/raw/raw924 /dev/sdcd14
chown oracle:dba /dev/raw/raw924
chmod 660 /dev/raw/raw924
raw /dev/raw/raw996 /dev/sdcd15
chown oracle:dba /dev/raw/raw996
chmod 660 /dev/raw/raw996
raw /dev/raw/raw61 /dev/sdce1
chown oracle:dba /dev/raw/raw61
chmod 660 /dev/raw/raw61
raw /dev/raw/raw133 /dev/sdce2
chown oracle:dba /dev/raw/raw133
chmod 660 /dev/raw/raw133
raw /dev/raw/raw205 /dev/sdce3
chown oracle:dba /dev/raw/raw205
chmod 660 /dev/raw/raw205
raw /dev/raw/raw277 /dev/sdce5
chown oracle:dba /dev/raw/raw277
chmod 660 /dev/raw/raw277
raw /dev/raw/raw349 /dev/sdce6
chown oracle:dba /dev/raw/raw349
chmod 660 /dev/raw/raw349
raw /dev/raw/raw421 /dev/sdce7
chown oracle:dba /dev/raw/raw421
chmod 660 /dev/raw/raw421
raw /dev/raw/raw493 /dev/sdce8
chown oracle:dba /dev/raw/raw493
chmod 660 /dev/raw/raw493
raw /dev/raw/raw565 /dev/sdce9
chown oracle:dba /dev/raw/raw565
chmod 660 /dev/raw/raw565
raw /dev/raw/raw637 /dev/sdce10
chown oracle:dba /dev/raw/raw637
chmod 660 /dev/raw/raw637
raw /dev/raw/raw709 /dev/sdce11
chown oracle:dba /dev/raw/raw709
chmod 660 /dev/raw/raw709
raw /dev/raw/raw781 /dev/sdce12
chown oracle:dba /dev/raw/raw781
chmod 660 /dev/raw/raw781
raw /dev/raw/raw853 /dev/sdce13
chown oracle:dba /dev/raw/raw853
chmod 660 /dev/raw/raw853
raw /dev/raw/raw925 /dev/sdce14
chown oracle:dba /dev/raw/raw925
chmod 660 /dev/raw/raw925
raw /dev/raw/raw997 /dev/sdce15
chown oracle:dba /dev/raw/raw997
chmod 660 /dev/raw/raw997
raw /dev/raw/raw62 /dev/sdcf1
chown oracle:dba /dev/raw/raw62
chmod 660 /dev/raw/raw62
raw /dev/raw/raw134 /dev/sdcf2
chown oracle:dba /dev/raw/raw134
chmod 660 /dev/raw/raw134
raw /dev/raw/raw206 /dev/sdcf3
chown oracle:dba /dev/raw/raw206
chmod 660 /dev/raw/raw206
raw /dev/raw/raw278 /dev/sdcf5
chown oracle:dba /dev/raw/raw278
chmod 660 /dev/raw/raw278
raw /dev/raw/raw350 /dev/sdcf6
chown oracle:dba /dev/raw/raw350
chmod 660 /dev/raw/raw350
raw /dev/raw/raw422 /dev/sdcf7
chown oracle:dba /dev/raw/raw422
chmod 660 /dev/raw/raw422
raw /dev/raw/raw494 /dev/sdcf8
chown oracle:dba /dev/raw/raw494
chmod 660 /dev/raw/raw494
raw /dev/raw/raw566 /dev/sdcf9
chown oracle:dba /dev/raw/raw566
chmod 660 /dev/raw/raw566
raw /dev/raw/raw638 /dev/sdcf10
chown oracle:dba /dev/raw/raw638
chmod 660 /dev/raw/raw638
raw /dev/raw/raw710 /dev/sdcf11
chown oracle:dba /dev/raw/raw710
chmod 660 /dev/raw/raw710
raw /dev/raw/raw782 /dev/sdcf12
chown oracle:dba /dev/raw/raw782
chmod 660 /dev/raw/raw782
raw /dev/raw/raw854 /dev/sdcf13
chown oracle:dba /dev/raw/raw854
chmod 660 /dev/raw/raw854
raw /dev/raw/raw926 /dev/sdcf14
chown oracle:dba /dev/raw/raw926
chmod 660 /dev/raw/raw926
raw /dev/raw/raw998 /dev/sdcf15
chown oracle:dba /dev/raw/raw998
chmod 660 /dev/raw/raw998
raw /dev/raw/raw63 /dev/sdcg1
chown oracle:dba /dev/raw/raw63
chmod 660 /dev/raw/raw63
raw /dev/raw/raw135 /dev/sdcg2
chown oracle:dba /dev/raw/raw135
chmod 660 /dev/raw/raw135
raw /dev/raw/raw207 /dev/sdcg3
chown oracle:dba /dev/raw/raw207
chmod 660 /dev/raw/raw207
raw /dev/raw/raw279 /dev/sdcg5
chown oracle:dba /dev/raw/raw279
chmod 660 /dev/raw/raw279
raw /dev/raw/raw351 /dev/sdcg6
chown oracle:dba /dev/raw/raw351
```


Copyright © 2004 NEC Corporation

```
chmod 660 /dev/raw/raw351
raw /dev/raw/raw423 /dev/sdcg7
chown oracle:dba /dev/raw/raw423
chmod 660 /dev/raw/raw423
raw /dev/raw/raw495 /dev/sdcg8
chown oracle:dba /dev/raw/raw495
chmod 660 /dev/raw/raw495
raw /dev/raw/raw567 /dev/sdcg9
chown oracle:dba /dev/raw/raw567
chmod 660 /dev/raw/raw567
raw /dev/raw/raw639 /dev/sdcg10
chown oracle:dba /dev/raw/raw639
chmod 660 /dev/raw/raw639
raw /dev/raw/raw711 /dev/sdcg11
chown oracle:dba /dev/raw/raw711
chmod 660 /dev/raw/raw711
raw /dev/raw/raw783 /dev/sdcg12
chown oracle:dba /dev/raw/raw783
chmod 660 /dev/raw/raw783
raw /dev/raw/raw855 /dev/sdcg13
chown oracle:dba /dev/raw/raw855
chmod 660 /dev/raw/raw855
raw /dev/raw/raw927 /dev/sdcg14
chown oracle:dba /dev/raw/raw927
chmod 660 /dev/raw/raw927
raw /dev/raw/raw999 /dev/sdcg15
chown oracle:dba /dev/raw/raw999
chmod 660 /dev/raw/raw999
raw /dev/raw/raw64 /dev/sdch1
chown oracle:dba /dev/raw/raw64
chmod 660 /dev/raw/raw64
raw /dev/raw/raw136 /dev/sdch2
chown oracle:dba /dev/raw/raw136
chmod 660 /dev/raw/raw136
raw /dev/raw/raw208 /dev/sdch3
chown oracle:dba /dev/raw/raw208
chmod 660 /dev/raw/raw208
raw /dev/raw/raw280 /dev/sdch5
chown oracle:dba /dev/raw/raw280
chmod 660 /dev/raw/raw280
raw /dev/raw/raw352 /dev/sdch6
chown oracle:dba /dev/raw/raw352
chmod 660 /dev/raw/raw352
raw /dev/raw/raw424 /dev/sdch7
chown oracle:dba /dev/raw/raw424
chmod 660 /dev/raw/raw424
raw /dev/raw/raw496 /dev/sdch8
chown oracle:dba /dev/raw/raw496
chmod 660 /dev/raw/raw496
raw /dev/raw/raw568 /dev/sdch9
chown oracle:dba /dev/raw/raw568
chmod 660 /dev/raw/raw568
raw /dev/raw/raw640 /dev/sdch10
chown oracle:dba /dev/raw/raw640
chmod 660 /dev/raw/raw640
raw /dev/raw/raw712 /dev/sdch11
chown oracle:dba /dev/raw/raw712
chmod 660 /dev/raw/raw712
raw /dev/raw/raw784 /dev/sdch12
chown oracle:dba /dev/raw/raw784
chmod 660 /dev/raw/raw784
raw /dev/raw/raw856 /dev/sdch13
chown oracle:dba /dev/raw/raw856
chmod 660 /dev/raw/raw856
raw /dev/raw/raw928 /dev/sdch14
chown oracle:dba /dev/raw/raw928
chmod 660 /dev/raw/raw928
raw /dev/raw/raw1000 /dev/sdch15
chown oracle:dba /dev/raw/raw1000
chmod 660 /dev/raw/raw1000
raw /dev/raw/raw65 /dev/sdcj1
chown oracle:dba /dev/raw/raw65
chmod 660 /dev/raw/raw65
raw /dev/raw/raw137 /dev/sdcj2
chown oracle:dba /dev/raw/raw137
chmod 660 /dev/raw/raw137
raw /dev/raw/raw209 /dev/sdcj3
chown oracle:dba /dev/raw/raw209
chmod 660 /dev/raw/raw209
raw /dev/raw/raw281 /dev/sdcj5
chown oracle:dba /dev/raw/raw281
chmod 660 /dev/raw/raw281
raw /dev/raw/raw353 /dev/sdcj6
chown oracle:dba /dev/raw/raw353
chmod 660 /dev/raw/raw353
raw /dev/raw/raw425 /dev/sdcj7
chown oracle:dba /dev/raw/raw425
chmod 660 /dev/raw/raw425
raw /dev/raw/raw497 /dev/sdcj8
chown oracle:dba /dev/raw/raw497
chmod 660 /dev/raw/raw497
raw /dev/raw/raw569 /dev/sdcj9
chown oracle:dba /dev/raw/raw569
chmod 660 /dev/raw/raw569
raw /dev/raw/raw641 /dev/sdcj10
chown oracle:dba /dev/raw/raw641
chmod 660 /dev/raw/raw641
raw /dev/raw/raw713 /dev/sdcj11
chown oracle:dba /dev/raw/raw713
chmod 660 /dev/raw/raw713
raw /dev/raw/raw785 /dev/sdcj12
chown oracle:dba /dev/raw/raw785
chmod 660 /dev/raw/raw785
raw /dev/raw/raw857 /dev/sdcj13
chown oracle:dba /dev/raw/raw857
chmod 660 /dev/raw/raw857
raw /dev/raw/raw929 /dev/sdcj14
chown oracle:dba /dev/raw/raw929
chmod 660 /dev/raw/raw929
raw /dev/raw/raw1001 /dev/sdcj15
chown oracle:dba /dev/raw/raw1001
chmod 660 /dev/raw/raw1001
```


Copyright © 2004 NEC Corporation

```
raw /dev/raw/raw66 /dev/sdck1
chown oracle:dba /dev/raw/raw66
chmod 660 /dev/raw/raw66
raw /dev/raw/raw138 /dev/sdck2
chown oracle:dba /dev/raw/raw138
chmod 660 /dev/raw/raw138
raw /dev/raw/raw210 /dev/sdck3
chown oracle:dba /dev/raw/raw210
chmod 660 /dev/raw/raw210
raw /dev/raw/raw282 /dev/sdck5
chown oracle:dba /dev/raw/raw282
chmod 660 /dev/raw/raw282
raw /dev/raw/raw354 /dev/sdck6
chown oracle:dba /dev/raw/raw354
chmod 660 /dev/raw/raw354
raw /dev/raw/raw426 /dev/sdck7
chown oracle:dba /dev/raw/raw426
chmod 660 /dev/raw/raw426
raw /dev/raw/raw498 /dev/sdck8
chown oracle:dba /dev/raw/raw498
chmod 660 /dev/raw/raw498
raw /dev/raw/raw570 /dev/sdck9
chown oracle:dba /dev/raw/raw570
chmod 660 /dev/raw/raw570
raw /dev/raw/raw642 /dev/sdck10
chown oracle:dba /dev/raw/raw642
chmod 660 /dev/raw/raw642
raw /dev/raw/raw714 /dev/sdck11
chown oracle:dba /dev/raw/raw714
chmod 660 /dev/raw/raw714
raw /dev/raw/raw786 /dev/sdck12
chown oracle:dba /dev/raw/raw786
chmod 660 /dev/raw/raw786
raw /dev/raw/raw858 /dev/sdck13
chown oracle:dba /dev/raw/raw858
chmod 660 /dev/raw/raw858
raw /dev/raw/raw930 /dev/sdck14
chown oracle:dba /dev/raw/raw930
chmod 660 /dev/raw/raw930
raw /dev/raw/raw1002 /dev/sdck15
chown oracle:dba /dev/raw/raw1002
chmod 660 /dev/raw/raw1002
raw /dev/raw/raw67 /dev/sdcm1
chown oracle:dba /dev/raw/raw67
chmod 660 /dev/raw/raw67
raw /dev/raw/raw139 /dev/sdcm2
chown oracle:dba /dev/raw/raw139
chmod 660 /dev/raw/raw139
raw /dev/raw/raw211 /dev/sdcm3
chown oracle:dba /dev/raw/raw211
chmod 660 /dev/raw/raw211
raw /dev/raw/raw283 /dev/sdcm5
chown oracle:dba /dev/raw/raw283
chmod 660 /dev/raw/raw283
raw /dev/raw/raw355 /dev/sdcm6
chown oracle:dba /dev/raw/raw355
chmod 660 /dev/raw/raw355
raw /dev/raw/raw427 /dev/sdcm7
chown oracle:dba /dev/raw/raw427
chmod 660 /dev/raw/raw427
raw /dev/raw/raw499 /dev/sdcm8
chown oracle:dba /dev/raw/raw499
chmod 660 /dev/raw/raw499
raw /dev/raw/raw571 /dev/sdcm9
chown oracle:dba /dev/raw/raw571
chmod 660 /dev/raw/raw571
raw /dev/raw/raw643 /dev/sdcm10
chown oracle:dba /dev/raw/raw643
chmod 660 /dev/raw/raw643
raw /dev/raw/raw715 /dev/sdcm11
chown oracle:dba /dev/raw/raw715
chmod 660 /dev/raw/raw715
raw /dev/raw/raw787 /dev/sdcm12
chown oracle:dba /dev/raw/raw787
chmod 660 /dev/raw/raw787
raw /dev/raw/raw859 /dev/sdcm13
chown oracle:dba /dev/raw/raw859
chmod 660 /dev/raw/raw859
raw /dev/raw/raw931 /dev/sdcm14
chown oracle:dba /dev/raw/raw931
chmod 660 /dev/raw/raw931
raw /dev/raw/raw1003 /dev/sdcm15
chown oracle:dba /dev/raw/raw1003
chmod 660 /dev/raw/raw1003
raw /dev/raw/raw68 /dev/sdcn1
chown oracle:dba /dev/raw/raw68
chmod 660 /dev/raw/raw68
raw /dev/raw/raw140 /dev/sdcn2
chown oracle:dba /dev/raw/raw140
chmod 660 /dev/raw/raw140
raw /dev/raw/raw212 /dev/sdcn3
chown oracle:dba /dev/raw/raw212
chmod 660 /dev/raw/raw212
raw /dev/raw/raw284 /dev/sdcn5
chown oracle:dba /dev/raw/raw284
chmod 660 /dev/raw/raw284
raw /dev/raw/raw356 /dev/sdcn6
chown oracle:dba /dev/raw/raw356
chmod 660 /dev/raw/raw356
raw /dev/raw/raw428 /dev/sdcn7
chown oracle:dba /dev/raw/raw428
chmod 660 /dev/raw/raw428
raw /dev/raw/raw500 /dev/sdcn8
chown oracle:dba /dev/raw/raw500
chmod 660 /dev/raw/raw500
raw /dev/raw/raw572 /dev/sdcn9
chown oracle:dba /dev/raw/raw572
chmod 660 /dev/raw/raw572
raw /dev/raw/raw644 /dev/sdcn10
chown oracle:dba /dev/raw/raw644
chmod 660 /dev/raw/raw644
raw /dev/raw/raw716 /dev/sdcn11
```


Copyright © 2004 NEC Corporation

```
chown oracle:dba /dev/raw/raw716
chmod 660 /dev/raw/raw716
raw /dev/raw/raw788 /dev/sdcn12
chown oracle:dba /dev/raw/raw788
chmod 660 /dev/raw/raw788
raw /dev/raw/raw860 /dev/sdcn13
chown oracle:dba /dev/raw/raw860
chmod 660 /dev/raw/raw860
raw /dev/raw/raw932 /dev/sdcn14
chown oracle:dba /dev/raw/raw932
chmod 660 /dev/raw/raw932
raw /dev/raw/raw1004 /dev/sdcn15
chown oracle:dba /dev/raw/raw1004
chmod 660 /dev/raw/raw1004
raw /dev/raw/raw69 /dev/sdcp1
chown oracle:dba /dev/raw/raw69
chmod 660 /dev/raw/raw69
raw /dev/raw/raw141 /dev/sdcp2
chown oracle:dba /dev/raw/raw141
chmod 660 /dev/raw/raw141
raw /dev/raw/raw213 /dev/sdcp3
chown oracle:dba /dev/raw/raw213
chmod 660 /dev/raw/raw213
raw /dev/raw/raw285 /dev/sdcp5
chown oracle:dba /dev/raw/raw285
chmod 660 /dev/raw/raw285
raw /dev/raw/raw357 /dev/sdcp6
chown oracle:dba /dev/raw/raw357
chmod 660 /dev/raw/raw357
raw /dev/raw/raw429 /dev/sdcp7
chown oracle:dba /dev/raw/raw429
chmod 660 /dev/raw/raw429
raw /dev/raw/raw501 /dev/sdcp8
chown oracle:dba /dev/raw/raw501
chmod 660 /dev/raw/raw501
raw /dev/raw/raw573 /dev/sdcp9
chown oracle:dba /dev/raw/raw573
chmod 660 /dev/raw/raw573
raw /dev/raw/raw645 /dev/sdcp10
chown oracle:dba /dev/raw/raw645
chmod 660 /dev/raw/raw645
raw /dev/raw/raw717 /dev/sdcp11
chown oracle:dba /dev/raw/raw717
chmod 660 /dev/raw/raw717
raw /dev/raw/raw789 /dev/sdcp12
chown oracle:dba /dev/raw/raw789
chmod 660 /dev/raw/raw789
raw /dev/raw/raw861 /dev/sdcp13
chown oracle:dba /dev/raw/raw861
chmod 660 /dev/raw/raw861
raw /dev/raw/raw933 /dev/sdcp14
chown oracle:dba /dev/raw/raw933
chmod 660 /dev/raw/raw933
raw /dev/raw/raw1005 /dev/sdcp15
chown oracle:dba /dev/raw/raw1005
chmod 660 /dev/raw/raw1005
raw /dev/raw/raw70 /dev/sdcq1
chown oracle:dba /dev/raw/raw70
chmod 660 /dev/raw/raw70
raw /dev/raw/raw142 /dev/sdcq2
chown oracle:dba /dev/raw/raw142
chmod 660 /dev/raw/raw142
raw /dev/raw/raw214 /dev/sdcq3
chown oracle:dba /dev/raw/raw214
chmod 660 /dev/raw/raw214
raw /dev/raw/raw286 /dev/sdcq5
chown oracle:dba /dev/raw/raw286
chmod 660 /dev/raw/raw286
raw /dev/raw/raw358 /dev/sdcq6
chown oracle:dba /dev/raw/raw358
chmod 660 /dev/raw/raw358
raw /dev/raw/raw430 /dev/sdcq7
chown oracle:dba /dev/raw/raw430
chmod 660 /dev/raw/raw430
raw /dev/raw/raw502 /dev/sdcq8
chown oracle:dba /dev/raw/raw502
chmod 660 /dev/raw/raw502
raw /dev/raw/raw574 /dev/sdcq9
chown oracle:dba /dev/raw/raw574
chmod 660 /dev/raw/raw574
raw /dev/raw/raw646 /dev/sdcq10
chown oracle:dba /dev/raw/raw646
chmod 660 /dev/raw/raw646
raw /dev/raw/raw718 /dev/sdcq11
chown oracle:dba /dev/raw/raw718
chmod 660 /dev/raw/raw718
raw /dev/raw/raw790 /dev/sdcq12
chown oracle:dba /dev/raw/raw790
chmod 660 /dev/raw/raw790
raw /dev/raw/raw862 /dev/sdcq13
chown oracle:dba /dev/raw/raw862
chmod 660 /dev/raw/raw862
raw /dev/raw/raw934 /dev/sdcq14
chown oracle:dba /dev/raw/raw934
chmod 660 /dev/raw/raw934
raw /dev/raw/raw1006 /dev/sdcq15
chown oracle:dba /dev/raw/raw1006
chmod 660 /dev/raw/raw1006
raw /dev/raw/raw71 /dev/sdcs1
chown oracle:dba /dev/raw/raw71
chmod 660 /dev/raw/raw71
raw /dev/raw/raw143 /dev/sdcs2
chown oracle:dba /dev/raw/raw143
chmod 660 /dev/raw/raw143
raw /dev/raw/raw215 /dev/sdcs3
chown oracle:dba /dev/raw/raw215
chmod 660 /dev/raw/raw215
raw /dev/raw/raw287 /dev/sdcs5
chown oracle:dba /dev/raw/raw287
chmod 660 /dev/raw/raw287
raw /dev/raw/raw359 /dev/sdcs6
chown oracle:dba /dev/raw/raw359
```


Copyright © 2004 NEC Corporation

```
chmod 660 /dev/raw/raw359
raw /dev/raw/raw431 /dev/sdcs7
chown oracle:dba /dev/raw/raw431
chmod 660 /dev/raw/raw431
raw /dev/raw/raw503 /dev/sdcs8
chown oracle:dba /dev/raw/raw503
chmod 660 /dev/raw/raw503
raw /dev/raw/raw575 /dev/sdcs9
chown oracle:dba /dev/raw/raw575
chmod 660 /dev/raw/raw575
raw /dev/raw/raw647 /dev/sdcs10
chown oracle:dba /dev/raw/raw647
chmod 660 /dev/raw/raw647
raw /dev/raw/raw719 /dev/sdcs11
chown oracle:dba /dev/raw/raw719
chmod 660 /dev/raw/raw719
raw /dev/raw/raw791 /dev/sdcs12
chown oracle:dba /dev/raw/raw791
chmod 660 /dev/raw/raw791
raw /dev/raw/raw863 /dev/sdcs13
chown oracle:dba /dev/raw/raw863
chmod 660 /dev/raw/raw863
raw /dev/raw/raw935 /dev/sdcs14
chown oracle:dba /dev/raw/raw935
chmod 660 /dev/raw/raw935
raw /dev/raw/raw1007 /dev/sdcs15
chown oracle:dba /dev/raw/raw1007
chmod 660 /dev/raw/raw1007
raw /dev/raw/raw72 /dev/sdct1
chown oracle:dba /dev/raw/raw72
chmod 660 /dev/raw/raw72
raw /dev/raw/raw144 /dev/sdct2
chown oracle:dba /dev/raw/raw144
chmod 660 /dev/raw/raw144
raw /dev/raw/raw216 /dev/sdct3
chown oracle:dba /dev/raw/raw216
chmod 660 /dev/raw/raw216
raw /dev/raw/raw288 /dev/sdct5
chown oracle:dba /dev/raw/raw288
chmod 660 /dev/raw/raw288
raw /dev/raw/raw360 /dev/sdct6
chown oracle:dba /dev/raw/raw360
chmod 660 /dev/raw/raw360
raw /dev/raw/raw432 /dev/sdct7
chown oracle:dba /dev/raw/raw432
chmod 660 /dev/raw/raw432
raw /dev/raw/raw504 /dev/sdct8
chown oracle:dba /dev/raw/raw504
chmod 660 /dev/raw/raw504
raw /dev/raw/raw576 /dev/sdct9
chown oracle:dba /dev/raw/raw576
chmod 660 /dev/raw/raw576
raw /dev/raw/raw648 /dev/sdct10
chown oracle:dba /dev/raw/raw648
chmod 660 /dev/raw/raw648
raw /dev/raw/raw720 /dev/sdct11
chown oracle:dba /dev/raw/raw720
chmod 660 /dev/raw/raw720
raw /dev/raw/raw792 /dev/sdct12
chown oracle:dba /dev/raw/raw792
chmod 660 /dev/raw/raw792
raw /dev/raw/raw864 /dev/sdct13
chown oracle:dba /dev/raw/raw864
chmod 660 /dev/raw/raw864
raw /dev/raw/raw936 /dev/sdct14
chown oracle:dba /dev/raw/raw936
chmod 660 /dev/raw/raw936
raw /dev/raw/raw1008 /dev/sdct15
chown oracle:dba /dev/raw/raw1008
chmod 660 /dev/raw/raw1008
raw /dev/raw/raw73 /dev/sdcv1
chown oracle:dba /dev/raw/raw73
chmod 660 /dev/raw/raw73
raw /dev/raw/raw145 /dev/sdcv2
chown oracle:dba /dev/raw/raw145
chmod 660 /dev/raw/raw145
raw /dev/raw/raw217 /dev/sdcv3
chown oracle:dba /dev/raw/raw217
chmod 660 /dev/raw/raw217
raw /dev/raw/raw289 /dev/sdcv5
chown oracle:dba /dev/raw/raw289
chmod 660 /dev/raw/raw289
raw /dev/raw/raw361 /dev/sdcv6
chown oracle:dba /dev/raw/raw361
chmod 660 /dev/raw/raw361
raw /dev/raw/raw433 /dev/sdcv7
chown oracle:dba /dev/raw/raw433
chmod 660 /dev/raw/raw433
raw /dev/raw/raw505 /dev/sdcv8
chown oracle:dba /dev/raw/raw505
chmod 660 /dev/raw/raw505
raw /dev/raw/raw577 /dev/sdcv9
chown oracle:dba /dev/raw/raw577
chmod 660 /dev/raw/raw577
raw /dev/raw/raw649 /dev/sdcv10
chown oracle:dba /dev/raw/raw649
chmod 660 /dev/raw/raw649
raw /dev/raw/raw721 /dev/sdcv11
chown oracle:dba /dev/raw/raw721
chmod 660 /dev/raw/raw721
raw /dev/raw/raw793 /dev/sdcv12
chown oracle:dba /dev/raw/raw793
chmod 660 /dev/raw/raw793
raw /dev/raw/raw865 /dev/sdcv13
chown oracle:dba /dev/raw/raw865
chmod 660 /dev/raw/raw865
raw /dev/raw/raw937 /dev/sdcv14
chown oracle:dba /dev/raw/raw937
chmod 660 /dev/raw/raw937
raw /dev/raw/raw1009 /dev/sdcv15
chown oracle:dba /dev/raw/raw1009
chmod 660 /dev/raw/raw1009
```


Copyright © 2004 NEC Corporation

```
raw /dev/raw/raw74 /dev/sdcw1
chown oracle:dba /dev/raw/raw74
chmod 660 /dev/raw/raw74
raw /dev/raw/raw146 /dev/sdcw2
chown oracle:dba /dev/raw/raw146
chmod 660 /dev/raw/raw146
raw /dev/raw/raw218 /dev/sdcw3
chown oracle:dba /dev/raw/raw218
chmod 660 /dev/raw/raw218
raw /dev/raw/raw290 /dev/sdcw5
chown oracle:dba /dev/raw/raw290
chmod 660 /dev/raw/raw290
raw /dev/raw/raw362 /dev/sdcw6
chown oracle:dba /dev/raw/raw362
chmod 660 /dev/raw/raw362
raw /dev/raw/raw434 /dev/sdcw7
chown oracle:dba /dev/raw/raw434
chmod 660 /dev/raw/raw434
raw /dev/raw/raw506 /dev/sdcw8
chown oracle:dba /dev/raw/raw506
chmod 660 /dev/raw/raw506
raw /dev/raw/raw578 /dev/sdcw9
chown oracle:dba /dev/raw/raw578
chmod 660 /dev/raw/raw578
raw /dev/raw/raw650 /dev/sdcw10
chown oracle:dba /dev/raw/raw650
chmod 660 /dev/raw/raw650
raw /dev/raw/raw722 /dev/sdcw11
chown oracle:dba /dev/raw/raw722
chmod 660 /dev/raw/raw722
raw /dev/raw/raw794 /dev/sdcw12
chown oracle:dba /dev/raw/raw794
chmod 660 /dev/raw/raw794
raw /dev/raw/raw866 /dev/sdcw13
chown oracle:dba /dev/raw/raw866
chmod 660 /dev/raw/raw866
raw /dev/raw/raw938 /dev/sdcw14
chown oracle:dba /dev/raw/raw938
chmod 660 /dev/raw/raw938
raw /dev/raw/raw1010 /dev/sdcw15
chown oracle:dba /dev/raw/raw1010
chmod 660 /dev/raw/raw1010
raw /dev/raw/raw5 /dev/sdd1
chown oracle:dba /dev/raw/raw5
chmod 660 /dev/raw/raw5
raw /dev/raw/raw77 /dev/sdd2
chown oracle:dba /dev/raw/raw77
chmod 660 /dev/raw/raw77
raw /dev/raw/raw149 /dev/sdd3
chown oracle:dba /dev/raw/raw149
chmod 660 /dev/raw/raw149
raw /dev/raw/raw221 /dev/sdd5
chown oracle:dba /dev/raw/raw221
chmod 660 /dev/raw/raw221
raw /dev/raw/raw293 /dev/sdd6
chown oracle:dba /dev/raw/raw293
chmod 660 /dev/raw/raw293
raw /dev/raw/raw365 /dev/sdd7
chown oracle:dba /dev/raw/raw365
chmod 660 /dev/raw/raw365
raw /dev/raw/raw437 /dev/sdd8
chown oracle:dba /dev/raw/raw437
chmod 660 /dev/raw/raw437
raw /dev/raw/raw509 /dev/sdd9
chown oracle:dba /dev/raw/raw509
chmod 660 /dev/raw/raw509
raw /dev/raw/raw581 /dev/sdd10
chown oracle:dba /dev/raw/raw581
chmod 660 /dev/raw/raw581
raw /dev/raw/raw653 /dev/sdd11
chown oracle:dba /dev/raw/raw653
chmod 660 /dev/raw/raw653
raw /dev/raw/raw725 /dev/sdd12
chown oracle:dba /dev/raw/raw725
chmod 660 /dev/raw/raw725
raw /dev/raw/raw797 /dev/sdd13
chown oracle:dba /dev/raw/raw797
chmod 660 /dev/raw/raw797
raw /dev/raw/raw869 /dev/sdd14
chown oracle:dba /dev/raw/raw869
chmod 660 /dev/raw/raw869
raw /dev/raw/raw941 /dev/sdd15
chown oracle:dba /dev/raw/raw941
chmod 660 /dev/raw/raw941
raw /dev/raw/raw6 /dev/sde1
chown oracle:dba /dev/raw/raw6
chmod 660 /dev/raw/raw6
raw /dev/raw/raw78 /dev/sde2
chown oracle:dba /dev/raw/raw78
chmod 660 /dev/raw/raw78
raw /dev/raw/raw150 /dev/sde3
chown oracle:dba /dev/raw/raw150
chmod 660 /dev/raw/raw150
raw /dev/raw/raw222 /dev/sde5
chown oracle:dba /dev/raw/raw222
chmod 660 /dev/raw/raw222
raw /dev/raw/raw294 /dev/sde6
chown oracle:dba /dev/raw/raw294
chmod 660 /dev/raw/raw294
raw /dev/raw/raw366 /dev/sde7
chown oracle:dba /dev/raw/raw366
chmod 660 /dev/raw/raw366
raw /dev/raw/raw438 /dev/sde8
chown oracle:dba /dev/raw/raw438
chmod 660 /dev/raw/raw438
raw /dev/raw/raw510 /dev/sde9
chown oracle:dba /dev/raw/raw510
chmod 660 /dev/raw/raw510
raw /dev/raw/raw582 /dev/sde10
chown oracle:dba /dev/raw/raw582
chmod 660 /dev/raw/raw582
raw /dev/raw/raw654 /dev/sde11
```


Copyright © 2004 NEC Corporation

```
chown oracle:dba /dev/raw/raw654
chmod 660 /dev/raw/raw654
raw /dev/raw/raw726 /dev/sde12
chown oracle:dba /dev/raw/raw726
chmod 660 /dev/raw/raw726
raw /dev/raw/raw798 /dev/sde13
chown oracle:dba /dev/raw/raw798
chmod 660 /dev/raw/raw798
raw /dev/raw/raw870 /dev/sde14
chown oracle:dba /dev/raw/raw870
chmod 660 /dev/raw/raw870
raw /dev/raw/raw942 /dev/sde15
chown oracle:dba /dev/raw/raw942
chmod 660 /dev/raw/raw942
raw /dev/raw/raw7 /dev/sdcy1
chown oracle:dba /dev/raw/raw7
chmod 660 /dev/raw/raw7
raw /dev/raw/raw79 /dev/sdcy2
chown oracle:dba /dev/raw/raw79
chmod 660 /dev/raw/raw79
raw /dev/raw/raw151 /dev/sdcy3
chown oracle:dba /dev/raw/raw151
chmod 660 /dev/raw/raw151
raw /dev/raw/raw223 /dev/sdcy5
chown oracle:dba /dev/raw/raw223
chmod 660 /dev/raw/raw223
raw /dev/raw/raw295 /dev/sdcy6
chown oracle:dba /dev/raw/raw295
chmod 660 /dev/raw/raw295
raw /dev/raw/raw367 /dev/sdcy7
chown oracle:dba /dev/raw/raw367
chmod 660 /dev/raw/raw367
raw /dev/raw/raw439 /dev/sdcy8
chown oracle:dba /dev/raw/raw439
chmod 660 /dev/raw/raw439
raw /dev/raw/raw511 /dev/sdcy9
chown oracle:dba /dev/raw/raw511
chmod 660 /dev/raw/raw511
raw /dev/raw/raw583 /dev/sdcy10
chown oracle:dba /dev/raw/raw583
chmod 660 /dev/raw/raw583
raw /dev/raw/raw655 /dev/sdcy11
chown oracle:dba /dev/raw/raw655
chmod 660 /dev/raw/raw655
raw /dev/raw/raw727 /dev/sdcy12
chown oracle:dba /dev/raw/raw727
chmod 660 /dev/raw/raw727
raw /dev/raw/raw799 /dev/sdcy13
chown oracle:dba /dev/raw/raw799
chmod 660 /dev/raw/raw799
raw /dev/raw/raw871 /dev/sdcy14
chown oracle:dba /dev/raw/raw871
chmod 660 /dev/raw/raw871
raw /dev/raw/raw943 /dev/sdcy15
chown oracle:dba /dev/raw/raw943
chmod 660 /dev/raw/raw943
raw /dev/raw/raw8 /dev/sdg1
chown oracle:dba /dev/raw/raw8
chmod 660 /dev/raw/raw8
raw /dev/raw/raw80 /dev/sdg2
chown oracle:dba /dev/raw/raw80
chmod 660 /dev/raw/raw80
raw /dev/raw/raw152 /dev/sdg3
chown oracle:dba /dev/raw/raw152
chmod 660 /dev/raw/raw152
raw /dev/raw/raw224 /dev/sdg5
chown oracle:dba /dev/raw/raw224
chmod 660 /dev/raw/raw224
raw /dev/raw/raw296 /dev/sdg6
chown oracle:dba /dev/raw/raw296
chmod 660 /dev/raw/raw296
raw /dev/raw/raw368 /dev/sdg7
chown oracle:dba /dev/raw/raw368
chmod 660 /dev/raw/raw368
raw /dev/raw/raw440 /dev/sdg8
chown oracle:dba /dev/raw/raw440
chmod 660 /dev/raw/raw440
raw /dev/raw/raw512 /dev/sdg9
chown oracle:dba /dev/raw/raw512
chmod 660 /dev/raw/raw512
raw /dev/raw/raw584 /dev/sdg10
chown oracle:dba /dev/raw/raw584
chmod 660 /dev/raw/raw584
raw /dev/raw/raw656 /dev/sdg11
chown oracle:dba /dev/raw/raw656
chmod 660 /dev/raw/raw656
raw /dev/raw/raw728 /dev/sdg12
chown oracle:dba /dev/raw/raw728
chmod 660 /dev/raw/raw728
raw /dev/raw/raw800 /dev/sdg13
chown oracle:dba /dev/raw/raw800
chmod 660 /dev/raw/raw800
raw /dev/raw/raw872 /dev/sdg14
chown oracle:dba /dev/raw/raw872
chmod 660 /dev/raw/raw872
raw /dev/raw/raw944 /dev/sdg15
chown oracle:dba /dev/raw/raw944
chmod 660 /dev/raw/raw944
raw /dev/raw/raw9 /dev/sdh1
chown oracle:dba /dev/raw/raw9
chmod 660 /dev/raw/raw9
raw /dev/raw/raw81 /dev/sdh2
chown oracle:dba /dev/raw/raw81
chmod 660 /dev/raw/raw81
raw /dev/raw/raw153 /dev/sdh3
chown oracle:dba /dev/raw/raw153
chmod 660 /dev/raw/raw153
raw /dev/raw/raw225 /dev/sdh5
chown oracle:dba /dev/raw/raw225
chmod 660 /dev/raw/raw225
raw /dev/raw/raw297 /dev/sdh6
chown oracle:dba /dev/raw/raw297
```


Copyright © 2004 NEC Corporation

```
chmod 660 /dev/raw/raw297
raw /dev/raw/raw369 /dev/sdh7
chown oracle:dba /dev/raw/raw369
chmod 660 /dev/raw/raw369
raw /dev/raw/raw441 /dev/sdh8
chown oracle:dba /dev/raw/raw441
chmod 660 /dev/raw/raw441
raw /dev/raw/raw513 /dev/sdh9
chown oracle:dba /dev/raw/raw513
chmod 660 /dev/raw/raw513
raw /dev/raw/raw585 /dev/sdh10
chown oracle:dba /dev/raw/raw585
chmod 660 /dev/raw/raw585
raw /dev/raw/raw657 /dev/sdh11
chown oracle:dba /dev/raw/raw657
chmod 660 /dev/raw/raw657
raw /dev/raw/raw729 /dev/sdh12
chown oracle:dba /dev/raw/raw729
chmod 660 /dev/raw/raw729
raw /dev/raw/raw801 /dev/sdh13
chown oracle:dba /dev/raw/raw801
chmod 660 /dev/raw/raw801
raw /dev/raw/raw873 /dev/sdh14
chown oracle:dba /dev/raw/raw873
chmod 660 /dev/raw/raw873
raw /dev/raw/raw945 /dev/sdh15
chown oracle:dba /dev/raw/raw945
chmod 660 /dev/raw/raw945
raw /dev/raw/raw10 /dev/sdi1
chown oracle:dba /dev/raw/raw10
chmod 660 /dev/raw/raw10
raw /dev/raw/raw82 /dev/sdi2
chown oracle:dba /dev/raw/raw82
chmod 660 /dev/raw/raw82
raw /dev/raw/raw154 /dev/sdi3
chown oracle:dba /dev/raw/raw154
chmod 660 /dev/raw/raw154
raw /dev/raw/raw226 /dev/sdi5
chown oracle:dba /dev/raw/raw226
chmod 660 /dev/raw/raw226
raw /dev/raw/raw298 /dev/sdi6
chown oracle:dba /dev/raw/raw298
chmod 660 /dev/raw/raw298
raw /dev/raw/raw370 /dev/sdi7
chown oracle:dba /dev/raw/raw370
chmod 660 /dev/raw/raw370
raw /dev/raw/raw442 /dev/sdi8
chown oracle:dba /dev/raw/raw442
chmod 660 /dev/raw/raw442
raw /dev/raw/raw514 /dev/sdi9
chown oracle:dba /dev/raw/raw514
chmod 660 /dev/raw/raw514
raw /dev/raw/raw586 /dev/sdi10
chown oracle:dba /dev/raw/raw586
chmod 660 /dev/raw/raw586
raw /dev/raw/raw658 /dev/sdi11
chown oracle:dba /dev/raw/raw658
chmod 660 /dev/raw/raw658
raw /dev/raw/raw730 /dev/sdi12
chown oracle:dba /dev/raw/raw730
chmod 660 /dev/raw/raw730
raw /dev/raw/raw802 /dev/sdi13
chown oracle:dba /dev/raw/raw802
chmod 660 /dev/raw/raw802
raw /dev/raw/raw874 /dev/sdi14
chown oracle:dba /dev/raw/raw874
chmod 660 /dev/raw/raw874
raw /dev/raw/raw946 /dev/sdi15
chown oracle:dba /dev/raw/raw946
chmod 660 /dev/raw/raw946
raw /dev/raw/raw11 /dev/sdo1
chown oracle:dba /dev/raw/raw11
chmod 660 /dev/raw/raw11
raw /dev/raw/raw83 /dev/sdo2
chown oracle:dba /dev/raw/raw83
chmod 660 /dev/raw/raw83
raw /dev/raw/raw155 /dev/sdo3
chown oracle:dba /dev/raw/raw155
chmod 660 /dev/raw/raw155
raw /dev/raw/raw227 /dev/sdo5
chown oracle:dba /dev/raw/raw227
chmod 660 /dev/raw/raw227
raw /dev/raw/raw299 /dev/sdo6
chown oracle:dba /dev/raw/raw299
chmod 660 /dev/raw/raw299
raw /dev/raw/raw371 /dev/sdo7
chown oracle:dba /dev/raw/raw371
chmod 660 /dev/raw/raw371
raw /dev/raw/raw443 /dev/sdo8
chown oracle:dba /dev/raw/raw443
chmod 660 /dev/raw/raw443
raw /dev/raw/raw515 /dev/sdo9
chown oracle:dba /dev/raw/raw515
chmod 660 /dev/raw/raw515
raw /dev/raw/raw587 /dev/sdo10
chown oracle:dba /dev/raw/raw587
chmod 660 /dev/raw/raw587
raw /dev/raw/raw659 /dev/sdo11
chown oracle:dba /dev/raw/raw659
chmod 660 /dev/raw/raw659
raw /dev/raw/raw731 /dev/sdo12
chown oracle:dba /dev/raw/raw731
chmod 660 /dev/raw/raw731
raw /dev/raw/raw803 /dev/sdo13
chown oracle:dba /dev/raw/raw803
chmod 660 /dev/raw/raw803
raw /dev/raw/raw875 /dev/sdo14
chown oracle:dba /dev/raw/raw875
chmod 660 /dev/raw/raw875
raw /dev/raw/raw947 /dev/sdo15
chown oracle:dba /dev/raw/raw947
chmod 660 /dev/raw/raw947
```


Copyright © 2004 NEC Corporation

```
raw /dev/raw/raw12 /dev/sdp1
chown oracle:dba /dev/raw/raw12
chmod 660 /dev/raw/raw12
raw /dev/raw/raw84 /dev/sdp2
chown oracle:dba /dev/raw/raw84
chmod 660 /dev/raw/raw84
raw /dev/raw/raw156 /dev/sdp3
chown oracle:dba /dev/raw/raw156
chmod 660 /dev/raw/raw156
raw /dev/raw/raw228 /dev/sdp5
chown oracle:dba /dev/raw/raw228
chmod 660 /dev/raw/raw228
raw /dev/raw/raw300 /dev/sdp6
chown oracle:dba /dev/raw/raw300
chmod 660 /dev/raw/raw300
raw /dev/raw/raw372 /dev/sdp7
chown oracle:dba /dev/raw/raw372
chmod 660 /dev/raw/raw372
raw /dev/raw/raw444 /dev/sdp8
chown oracle:dba /dev/raw/raw444
chmod 660 /dev/raw/raw444
raw /dev/raw/raw516 /dev/sdp9
chown oracle:dba /dev/raw/raw516
chmod 660 /dev/raw/raw516
raw /dev/raw/raw588 /dev/sdp10
chown oracle:dba /dev/raw/raw588
chmod 660 /dev/raw/raw588
raw /dev/raw/raw660 /dev/sdp11
chown oracle:dba /dev/raw/raw660
chmod 660 /dev/raw/raw660
raw /dev/raw/raw732 /dev/sdp12
chown oracle:dba /dev/raw/raw732
chmod 660 /dev/raw/raw732
raw /dev/raw/raw804 /dev/sdp13
chown oracle:dba /dev/raw/raw804
chmod 660 /dev/raw/raw804
raw /dev/raw/raw876 /dev/sdp14
chown oracle:dba /dev/raw/raw876
chmod 660 /dev/raw/raw876
raw /dev/raw/raw948 /dev/sdp15
chown oracle:dba /dev/raw/raw948
chmod 660 /dev/raw/raw948
raw /dev/raw/raw13 /dev/sdr1
chown oracle:dba /dev/raw/raw13
chmod 660 /dev/raw/raw13
raw /dev/raw/raw85 /dev/sdr2
chown oracle:dba /dev/raw/raw85
chmod 660 /dev/raw/raw85
raw /dev/raw/raw157 /dev/sdr3
chown oracle:dba /dev/raw/raw157
chmod 660 /dev/raw/raw157
raw /dev/raw/raw229 /dev/sdr5
chown oracle:dba /dev/raw/raw229
chmod 660 /dev/raw/raw229
raw /dev/raw/raw301 /dev/sdr6
chown oracle:dba /dev/raw/raw301
chmod 660 /dev/raw/raw301
raw /dev/raw/raw373 /dev/sdr7
chown oracle:dba /dev/raw/raw373
chmod 660 /dev/raw/raw373
raw /dev/raw/raw445 /dev/sdr8
chown oracle:dba /dev/raw/raw445
chmod 660 /dev/raw/raw445
raw /dev/raw/raw517 /dev/sdr9
chown oracle:dba /dev/raw/raw517
chmod 660 /dev/raw/raw517
raw /dev/raw/raw589 /dev/sdr10
chown oracle:dba /dev/raw/raw589
chmod 660 /dev/raw/raw589
raw /dev/raw/raw661 /dev/sdr11
chown oracle:dba /dev/raw/raw661
chmod 660 /dev/raw/raw661
raw /dev/raw/raw733 /dev/sdr12
chown oracle:dba /dev/raw/raw733
chmod 660 /dev/raw/raw733
raw /dev/raw/raw805 /dev/sdr13
chown oracle:dba /dev/raw/raw805
chmod 660 /dev/raw/raw805
raw /dev/raw/raw877 /dev/sdr14
chown oracle:dba /dev/raw/raw877
chmod 660 /dev/raw/raw877
raw /dev/raw/raw949 /dev/sdr15
chown oracle:dba /dev/raw/raw949
chmod 660 /dev/raw/raw949
raw /dev/raw/raw14 /dev/sds1
chown oracle:dba /dev/raw/raw14
chmod 660 /dev/raw/raw14
raw /dev/raw/raw86 /dev/sds2
chown oracle:dba /dev/raw/raw86
chmod 660 /dev/raw/raw86
raw /dev/raw/raw158 /dev/sds3
chown oracle:dba /dev/raw/raw158
chmod 660 /dev/raw/raw158
raw /dev/raw/raw230 /dev/sds5
chown oracle:dba /dev/raw/raw230
chmod 660 /dev/raw/raw230
raw /dev/raw/raw302 /dev/sds6
chown oracle:dba /dev/raw/raw302
chmod 660 /dev/raw/raw302
raw /dev/raw/raw374 /dev/sds7
chown oracle:dba /dev/raw/raw374
chmod 660 /dev/raw/raw374
raw /dev/raw/raw446 /dev/sds8
chown oracle:dba /dev/raw/raw446
chmod 660 /dev/raw/raw446
raw /dev/raw/raw518 /dev/sds9
chown oracle:dba /dev/raw/raw518
chmod 660 /dev/raw/raw518
raw /dev/raw/raw590 /dev/sds10
chown oracle:dba /dev/raw/raw590
chmod 660 /dev/raw/raw590
raw /dev/raw/raw662 /dev/sds11
```


Copyright © 2004 NEC Corporation

```
chown oracle:dba /dev/raw/raw662
chmod 660 /dev/raw/raw662
raw /dev/raw/raw734 /dev/sds12
chown oracle:dba /dev/raw/raw734
chmod 660 /dev/raw/raw734
raw /dev/raw/raw806 /dev/sds13
chown oracle:dba /dev/raw/raw806
chmod 660 /dev/raw/raw806
raw /dev/raw/raw878 /dev/sds14
chown oracle:dba /dev/raw/raw878
chmod 660 /dev/raw/raw878
raw /dev/raw/raw950 /dev/sds15
chown oracle:dba /dev/raw/raw950
chmod 660 /dev/raw/raw950
raw /dev/raw/raw15 /dev/sdu1
chown oracle:dba /dev/raw/raw15
chmod 660 /dev/raw/raw15
raw /dev/raw/raw87 /dev/sdu2
chown oracle:dba /dev/raw/raw87
chmod 660 /dev/raw/raw87
raw /dev/raw/raw159 /dev/sdu3
chown oracle:dba /dev/raw/raw159
chmod 660 /dev/raw/raw159
raw /dev/raw/raw231 /dev/sdu5
chown oracle:dba /dev/raw/raw231
chmod 660 /dev/raw/raw231
raw /dev/raw/raw303 /dev/sdu6
chown oracle:dba /dev/raw/raw303
chmod 660 /dev/raw/raw303
raw /dev/raw/raw375 /dev/sdu7
chown oracle:dba /dev/raw/raw375
chmod 660 /dev/raw/raw375
raw /dev/raw/raw447 /dev/sdu8
chown oracle:dba /dev/raw/raw447
chmod 660 /dev/raw/raw447
raw /dev/raw/raw519 /dev/sdu9
chown oracle:dba /dev/raw/raw519
chmod 660 /dev/raw/raw519
raw /dev/raw/raw591 /dev/sdu10
chown oracle:dba /dev/raw/raw591
chmod 660 /dev/raw/raw591
raw /dev/raw/raw663 /dev/sdu11
chown oracle:dba /dev/raw/raw663
chmod 660 /dev/raw/raw663
raw /dev/raw/raw735 /dev/sdu12
chown oracle:dba /dev/raw/raw735
chmod 660 /dev/raw/raw735
raw /dev/raw/raw807 /dev/sdu13
chown oracle:dba /dev/raw/raw807
chmod 660 /dev/raw/raw807
raw /dev/raw/raw879 /dev/sdu14
chown oracle:dba /dev/raw/raw879
chmod 660 /dev/raw/raw879
raw /dev/raw/raw951 /dev/sdu15
chown oracle:dba /dev/raw/raw951
chmod 660 /dev/raw/raw951
raw /dev/raw/raw16 /dev/sdv1
chown oracle:dba /dev/raw/raw16
chmod 660 /dev/raw/raw16
raw /dev/raw/raw88 /dev/sdv2
chown oracle:dba /dev/raw/raw88
chmod 660 /dev/raw/raw88
raw /dev/raw/raw160 /dev/sdv3
chown oracle:dba /dev/raw/raw160
chmod 660 /dev/raw/raw160
raw /dev/raw/raw232 /dev/sdv5
chown oracle:dba /dev/raw/raw232
chmod 660 /dev/raw/raw232
raw /dev/raw/raw304 /dev/sdv6
chown oracle:dba /dev/raw/raw304
chmod 660 /dev/raw/raw304
raw /dev/raw/raw376 /dev/sdv7
chown oracle:dba /dev/raw/raw376
chmod 660 /dev/raw/raw376
raw /dev/raw/raw448 /dev/sdv8
chown oracle:dba /dev/raw/raw448
chmod 660 /dev/raw/raw448
raw /dev/raw/raw520 /dev/sdv9
chown oracle:dba /dev/raw/raw520
chmod 660 /dev/raw/raw520
raw /dev/raw/raw592 /dev/sdv10
chown oracle:dba /dev/raw/raw592
chmod 660 /dev/raw/raw592
raw /dev/raw/raw664 /dev/sdv11
chown oracle:dba /dev/raw/raw664
chmod 660 /dev/raw/raw664
raw /dev/raw/raw736 /dev/sdv12
chown oracle:dba /dev/raw/raw736
chmod 660 /dev/raw/raw736
raw /dev/raw/raw808 /dev/sdv13
chown oracle:dba /dev/raw/raw808
chmod 660 /dev/raw/raw808
raw /dev/raw/raw880 /dev/sdv14
chown oracle:dba /dev/raw/raw880
chmod 660 /dev/raw/raw880
raw /dev/raw/raw952 /dev/sdv15
chown oracle:dba /dev/raw/raw952
chmod 660 /dev/raw/raw952
raw /dev/raw/raw17 /dev/sdw1
chown oracle:dba /dev/raw/raw17
chmod 660 /dev/raw/raw17
raw /dev/raw/raw89 /dev/sdw2
chown oracle:dba /dev/raw/raw89
chmod 660 /dev/raw/raw89
raw /dev/raw/raw161 /dev/sdw3
chown oracle:dba /dev/raw/raw161
chmod 660 /dev/raw/raw161
raw /dev/raw/raw233 /dev/sdw5
chown oracle:dba /dev/raw/raw233
chmod 660 /dev/raw/raw233
raw /dev/raw/raw305 /dev/sdw6
chown oracle:dba /dev/raw/raw305
```


Copyright © 2004 NEC Corporation

```
chmod 660 /dev/raw/raw305
raw /dev/raw/raw377 /dev/sdw7
chown oracle:dba /dev/raw/raw377
chmod 660 /dev/raw/raw377
raw /dev/raw/raw449 /dev/sdw8
chown oracle:dba /dev/raw/raw449
chmod 660 /dev/raw/raw449
raw /dev/raw/raw521 /dev/sdw9
chown oracle:dba /dev/raw/raw521
chmod 660 /dev/raw/raw521
raw /dev/raw/raw593 /dev/sdw10
chown oracle:dba /dev/raw/raw593
chmod 660 /dev/raw/raw593
raw /dev/raw/raw665 /dev/sdw11
chown oracle:dba /dev/raw/raw665
chmod 660 /dev/raw/raw665
raw /dev/raw/raw737 /dev/sdw12
chown oracle:dba /dev/raw/raw737
chmod 660 /dev/raw/raw737
raw /dev/raw/raw809 /dev/sdw13
chown oracle:dba /dev/raw/raw809
chmod 660 /dev/raw/raw809
raw /dev/raw/raw881 /dev/sdw14
chown oracle:dba /dev/raw/raw881
chmod 660 /dev/raw/raw881
raw /dev/raw/raw953 /dev/sdw15
chown oracle:dba /dev/raw/raw953
chmod 660 /dev/raw/raw953
raw /dev/raw/raw18 /dev/sdx1
chown oracle:dba /dev/raw/raw18
chmod 660 /dev/raw/raw18
raw /dev/raw/raw90 /dev/sdx2
chown oracle:dba /dev/raw/raw90
chmod 660 /dev/raw/raw90
raw /dev/raw/raw162 /dev/sdx3
chown oracle:dba /dev/raw/raw162
chmod 660 /dev/raw/raw162
raw /dev/raw/raw234 /dev/sdx5
chown oracle:dba /dev/raw/raw234
chmod 660 /dev/raw/raw234
raw /dev/raw/raw306 /dev/sdx6
chown oracle:dba /dev/raw/raw306
chmod 660 /dev/raw/raw306
raw /dev/raw/raw378 /dev/sdx7
chown oracle:dba /dev/raw/raw378
chmod 660 /dev/raw/raw378
raw /dev/raw/raw450 /dev/sdx8
chown oracle:dba /dev/raw/raw450
chmod 660 /dev/raw/raw450
raw /dev/raw/raw522 /dev/sdx9
chown oracle:dba /dev/raw/raw522
chmod 660 /dev/raw/raw522
raw /dev/raw/raw594 /dev/sdx10
chown oracle:dba /dev/raw/raw594
chmod 660 /dev/raw/raw594
raw /dev/raw/raw666 /dev/sdx11
chown oracle:dba /dev/raw/raw666
chmod 660 /dev/raw/raw666
raw /dev/raw/raw738 /dev/sdx12
chown oracle:dba /dev/raw/raw738
chmod 660 /dev/raw/raw738
raw /dev/raw/raw810 /dev/sdx13
chown oracle:dba /dev/raw/raw810
chmod 660 /dev/raw/raw810
raw /dev/raw/raw882 /dev/sdx14
chown oracle:dba /dev/raw/raw882
chmod 660 /dev/raw/raw882
raw /dev/raw/raw954 /dev/sdx15
chown oracle:dba /dev/raw/raw954
chmod 660 /dev/raw/raw954
raw /dev/raw/raw19 /dev/sdy1
chown oracle:dba /dev/raw/raw19
chmod 660 /dev/raw/raw19
raw /dev/raw/raw91 /dev/sdy2
chown oracle:dba /dev/raw/raw91
chmod 660 /dev/raw/raw91
raw /dev/raw/raw163 /dev/sdy3
chown oracle:dba /dev/raw/raw163
chmod 660 /dev/raw/raw163
raw /dev/raw/raw235 /dev/sdy5
chown oracle:dba /dev/raw/raw235
chmod 660 /dev/raw/raw235
raw /dev/raw/raw307 /dev/sdy6
chown oracle:dba /dev/raw/raw307
chmod 660 /dev/raw/raw307
raw /dev/raw/raw379 /dev/sdy7
chown oracle:dba /dev/raw/raw379
chmod 660 /dev/raw/raw379
raw /dev/raw/raw451 /dev/sdy8
chown oracle:dba /dev/raw/raw451
chmod 660 /dev/raw/raw451
raw /dev/raw/raw523 /dev/sdy9
chown oracle:dba /dev/raw/raw523
chmod 660 /dev/raw/raw523
raw /dev/raw/raw595 /dev/sdy10
chown oracle:dba /dev/raw/raw595
chmod 660 /dev/raw/raw595
raw /dev/raw/raw667 /dev/sdy11
chown oracle:dba /dev/raw/raw667
chmod 660 /dev/raw/raw667
raw /dev/raw/raw739 /dev/sdy12
chown oracle:dba /dev/raw/raw739
chmod 660 /dev/raw/raw739
raw /dev/raw/raw811 /dev/sdy13
chown oracle:dba /dev/raw/raw811
chmod 660 /dev/raw/raw811
raw /dev/raw/raw883 /dev/sdy14
chown oracle:dba /dev/raw/raw883
chmod 660 /dev/raw/raw883
raw /dev/raw/raw955 /dev/sdy15
chown oracle:dba /dev/raw/raw955
chmod 660 /dev/raw/raw955
```


Copyright © 2004 NEC Corporation

```
raw /dev/raw/raw20 /dev/sdz1
chown oracle:dba /dev/raw/raw20
chmod 660 /dev/raw/raw20
raw /dev/raw/raw92 /dev/sdz2
chown oracle:dba /dev/raw/raw92
chmod 660 /dev/raw/raw92
raw /dev/raw/raw164 /dev/sdz3
chown oracle:dba /dev/raw/raw164
chmod 660 /dev/raw/raw164
raw /dev/raw/raw236 /dev/sdz5
chown oracle:dba /dev/raw/raw236
chmod 660 /dev/raw/raw236
raw /dev/raw/raw308 /dev/sdz6
chown oracle:dba /dev/raw/raw308
chmod 660 /dev/raw/raw308
raw /dev/raw/raw380 /dev/sdz7
chown oracle:dba /dev/raw/raw380
chmod 660 /dev/raw/raw380
raw /dev/raw/raw452 /dev/sdz8
chown oracle:dba /dev/raw/raw452
chmod 660 /dev/raw/raw452
raw /dev/raw/raw524 /dev/sdz9
chown oracle:dba /dev/raw/raw524
chmod 660 /dev/raw/raw524
raw /dev/raw/raw596 /dev/sdz10
chown oracle:dba /dev/raw/raw596
chmod 660 /dev/raw/raw596
raw /dev/raw/raw668 /dev/sdz11
chown oracle:dba /dev/raw/raw668
chmod 660 /dev/raw/raw668
raw /dev/raw/raw740 /dev/sdz12
chown oracle:dba /dev/raw/raw740
chmod 660 /dev/raw/raw740
raw /dev/raw/raw812 /dev/sdz13
chown oracle:dba /dev/raw/raw812
chmod 660 /dev/raw/raw812
raw /dev/raw/raw884 /dev/sdz14
chown oracle:dba /dev/raw/raw884
chmod 660 /dev/raw/raw884
raw /dev/raw/raw956 /dev/sdz15
chown oracle:dba /dev/raw/raw956
chmod 660 /dev/raw/raw956
raw /dev/raw/raw1012 /dev/sdk1
chown oracle:dba /dev/raw/raw1012
chmod 660 /dev/raw/raw1012
raw /dev/raw/raw1013 /dev/sdk2
chown oracle:dba /dev/raw/raw1013
chmod 660 /dev/raw/raw1013
```

link.sh

```
if [ ! -d /oradata/tpcc50000 ]
then
  mkdir /oradata/tpcc50000
fi

rm -f /oradata/tpcc50000/*
ln -s /dev/raw/raw1 /oradata/tpcc50000/log_1_1
ln -s /dev/raw/raw2 /oradata/tpcc50000/log_1_2
ln -s /dev/raw/raw1012 /oradata/tpcc50000/log_1_3
ln -s /dev/raw/raw1013 /oradata/tpcc50000/log_1_4
ln -s /dev/raw/raw3 /oradata/tpcc50000/stok_0_0
ln -s /dev/raw/raw4 /oradata/tpcc50000/stok_0_1
ln -s /dev/raw/raw5 /oradata/tpcc50000/stok_0_2
ln -s /dev/raw/raw6 /oradata/tpcc50000/stok_0_3
ln -s /dev/raw/raw7 /oradata/tpcc50000/stok_0_4
ln -s /dev/raw/raw8 /oradata/tpcc50000/stok_0_5
ln -s /dev/raw/raw9 /oradata/tpcc50000/stok_0_6
ln -s /dev/raw/raw10 /oradata/tpcc50000/stok_0_7
ln -s /dev/raw/raw11 /oradata/tpcc50000/stok_0_8
ln -s /dev/raw/raw12 /oradata/tpcc50000/stok_0_9
ln -s /dev/raw/raw13 /oradata/tpcc50000/stok_0_10
ln -s /dev/raw/raw14 /oradata/tpcc50000/stok_0_11
ln -s /dev/raw/raw15 /oradata/tpcc50000/stok_0_12
ln -s /dev/raw/raw16 /oradata/tpcc50000/stok_0_13
ln -s /dev/raw/raw17 /oradata/tpcc50000/stok_0_14
ln -s /dev/raw/raw18 /oradata/tpcc50000/stok_0_15
ln -s /dev/raw/raw19 /oradata/tpcc50000/stok_0_16
ln -s /dev/raw/raw20 /oradata/tpcc50000/stok_0_17
ln -s /dev/raw/raw21 /oradata/tpcc50000/stok_0_18
ln -s /dev/raw/raw22 /oradata/tpcc50000/stok_0_19
ln -s /dev/raw/raw23 /oradata/tpcc50000/stok_0_20
ln -s /dev/raw/raw24 /oradata/tpcc50000/stok_0_21
ln -s /dev/raw/raw25 /oradata/tpcc50000/stok_0_22
ln -s /dev/raw/raw26 /oradata/tpcc50000/stok_0_23
ln -s /dev/raw/raw27 /oradata/tpcc50000/stok_0_24
ln -s /dev/raw/raw28 /oradata/tpcc50000/stok_0_25
ln -s /dev/raw/raw29 /oradata/tpcc50000/stok_0_26
ln -s /dev/raw/raw30 /oradata/tpcc50000/stok_0_27
ln -s /dev/raw/raw31 /oradata/tpcc50000/stok_0_28
ln -s /dev/raw/raw32 /oradata/tpcc50000/stok_0_29
ln -s /dev/raw/raw33 /oradata/tpcc50000/stok_0_30
ln -s /dev/raw/raw34 /oradata/tpcc50000/stok_0_31
ln -s /dev/raw/raw35 /oradata/tpcc50000/stok_0_32
ln -s /dev/raw/raw36 /oradata/tpcc50000/stok_0_33
ln -s /dev/raw/raw37 /oradata/tpcc50000/stok_0_34
ln -s /dev/raw/raw38 /oradata/tpcc50000/stok_0_35
```


Copyright © 2004 NEC Corporation

```
ln -s /dev/raw/raw39 /oradata/tpcc50000/stok_0_36
ln -s /dev/raw/raw40 /oradata/tpcc50000/stok_0_37
ln -s /dev/raw/raw41 /oradata/tpcc50000/stok_0_38
ln -s /dev/raw/raw42 /oradata/tpcc50000/stok_0_39
ln -s /dev/raw/raw43 /oradata/tpcc50000/stok_0_40
ln -s /dev/raw/raw44 /oradata/tpcc50000/stok_0_41
ln -s /dev/raw/raw45 /oradata/tpcc50000/stok_0_42
ln -s /dev/raw/raw46 /oradata/tpcc50000/stok_0_43
ln -s /dev/raw/raw47 /oradata/tpcc50000/stok_0_44
ln -s /dev/raw/raw48 /oradata/tpcc50000/stok_0_45
ln -s /dev/raw/raw49 /oradata/tpcc50000/stok_0_46
ln -s /dev/raw/raw50 /oradata/tpcc50000/stok_0_47
ln -s /dev/raw/raw51 /oradata/tpcc50000/stok_0_48
ln -s /dev/raw/raw52 /oradata/tpcc50000/stok_0_49
ln -s /dev/raw/raw53 /oradata/tpcc50000/stok_0_50
ln -s /dev/raw/raw54 /oradata/tpcc50000/stok_0_51
ln -s /dev/raw/raw55 /oradata/tpcc50000/stok_0_52
ln -s /dev/raw/raw56 /oradata/tpcc50000/stok_0_53
ln -s /dev/raw/raw57 /oradata/tpcc50000/stok_0_54
ln -s /dev/raw/raw58 /oradata/tpcc50000/stok_0_55
ln -s /dev/raw/raw59 /oradata/tpcc50000/stok_0_56
ln -s /dev/raw/raw60 /oradata/tpcc50000/stok_0_57
ln -s /dev/raw/raw61 /oradata/tpcc50000/stok_0_58
ln -s /dev/raw/raw62 /oradata/tpcc50000/stok_0_59
ln -s /dev/raw/raw63 /oradata/tpcc50000/stok_0_60
ln -s /dev/raw/raw64 /oradata/tpcc50000/stok_0_61
ln -s /dev/raw/raw65 /oradata/tpcc50000/stok_0_62
ln -s /dev/raw/raw66 /oradata/tpcc50000/stok_0_63
ln -s /dev/raw/raw67 /oradata/tpcc50000/stok_0_64
ln -s /dev/raw/raw68 /oradata/tpcc50000/stok_0_65
ln -s /dev/raw/raw69 /oradata/tpcc50000/stok_0_66
ln -s /dev/raw/raw70 /oradata/tpcc50000/stok_0_67
ln -s /dev/raw/raw71 /oradata/tpcc50000/stok_0_68
ln -s /dev/raw/raw72 /oradata/tpcc50000/stok_0_69
ln -s /dev/raw/raw73 /oradata/tpcc50000/stok_0_70
ln -s /dev/raw/raw74 /oradata/tpcc50000/stok_0_71
ln -s /dev/raw/raw75 /oradata/tpcc50000/stok_0_72
ln -s /dev/raw/raw76 /oradata/tpcc50000/stok_0_73
ln -s /dev/raw/raw77 /oradata/tpcc50000/stok_0_74
ln -s /dev/raw/raw78 /oradata/tpcc50000/stok_0_75
ln -s /dev/raw/raw79 /oradata/tpcc50000/stok_0_76
ln -s /dev/raw/raw80 /oradata/tpcc50000/stok_0_77
ln -s /dev/raw/raw81 /oradata/tpcc50000/stok_0_78
ln -s /dev/raw/raw82 /oradata/tpcc50000/stok_0_79
ln -s /dev/raw/raw83 /oradata/tpcc50000/stok_0_80
ln -s /dev/raw/raw84 /oradata/tpcc50000/stok_0_81
ln -s /dev/raw/raw85 /oradata/tpcc50000/stok_0_82
ln -s /dev/raw/raw86 /oradata/tpcc50000/stok_0_83
ln -s /dev/raw/raw87 /oradata/tpcc50000/stok_0_84
ln -s /dev/raw/raw88 /oradata/tpcc50000/stok_0_85
ln -s /dev/raw/raw89 /oradata/tpcc50000/stok_0_86
ln -s /dev/raw/raw90 /oradata/tpcc50000/stok_0_87
ln -s /dev/raw/raw91 /oradata/tpcc50000/stok_0_88
ln -s /dev/raw/raw92 /oradata/tpcc50000/stok_0_89
ln -s /dev/raw/raw93 /oradata/tpcc50000/stok_0_90
ln -s /dev/raw/raw94 /oradata/tpcc50000/stok_0_91
ln -s /dev/raw/raw95 /oradata/tpcc50000/stok_0_92
ln -s /dev/raw/raw96 /oradata/tpcc50000/stok_0_93
ln -s /dev/raw/raw97 /oradata/tpcc50000/stok_0_94
ln -s /dev/raw/raw98 /oradata/tpcc50000/stok_0_95
ln -s /dev/raw/raw99 /oradata/tpcc50000/stok_0_96
ln -s /dev/raw/raw100 /oradata/tpcc50000/stok_0_97
ln -s /dev/raw/raw101 /oradata/tpcc50000/stok_0_98
ln -s /dev/raw/raw102 /oradata/tpcc50000/stok_0_99
ln -s /dev/raw/raw103 /oradata/tpcc50000/stok_0_100
ln -s /dev/raw/raw104 /oradata/tpcc50000/stok_0_101
ln -s /dev/raw/raw105 /oradata/tpcc50000/stok_0_102
ln -s /dev/raw/raw106 /oradata/tpcc50000/stok_0_103
ln -s /dev/raw/raw107 /oradata/tpcc50000/stok_0_104
ln -s /dev/raw/raw108 /oradata/tpcc50000/stok_0_105
ln -s /dev/raw/raw109 /oradata/tpcc50000/stok_0_106
ln -s /dev/raw/raw110 /oradata/tpcc50000/stok_0_107
ln -s /dev/raw/raw111 /oradata/tpcc50000/stok_0_108
ln -s /dev/raw/raw112 /oradata/tpcc50000/stok_0_109
ln -s /dev/raw/raw113 /oradata/tpcc50000/stok_0_110
ln -s /dev/raw/raw114 /oradata/tpcc50000/stok_0_111
ln -s /dev/raw/raw115 /oradata/tpcc50000/stok_0_112
ln -s /dev/raw/raw116 /oradata/tpcc50000/stok_0_113
ln -s /dev/raw/raw117 /oradata/tpcc50000/stok_0_114
ln -s /dev/raw/raw118 /oradata/tpcc50000/stok_0_115
ln -s /dev/raw/raw119 /oradata/tpcc50000/stok_0_116
ln -s /dev/raw/raw120 /oradata/tpcc50000/stok_0_117
ln -s /dev/raw/raw121 /oradata/tpcc50000/stok_0_118
ln -s /dev/raw/raw122 /oradata/tpcc50000/stok_0_119
ln -s /dev/raw/raw123 /oradata/tpcc50000/stok_0_120
ln -s /dev/raw/raw124 /oradata/tpcc50000/stok_0_121
ln -s /dev/raw/raw125 /oradata/tpcc50000/stok_0_122
ln -s /dev/raw/raw126 /oradata/tpcc50000/stok_0_123
```


Copyright © 2004 NEC Corporation

```
ln -s /dev/raw/raw127 /oradata/tpcc50000/stok_0_124
ln -s /dev/raw/raw128 /oradata/tpcc50000/stok_0_125
ln -s /dev/raw/raw129 /oradata/tpcc50000/stok_0_126
ln -s /dev/raw/raw130 /oradata/tpcc50000/stok_0_127
ln -s /dev/raw/raw131 /oradata/tpcc50000/stok_0_128
ln -s /dev/raw/raw132 /oradata/tpcc50000/stok_0_129
ln -s /dev/raw/raw133 /oradata/tpcc50000/stok_0_130
ln -s /dev/raw/raw134 /oradata/tpcc50000/stok_0_131
ln -s /dev/raw/raw135 /oradata/tpcc50000/stok_0_132
ln -s /dev/raw/raw136 /oradata/tpcc50000/stok_0_133
ln -s /dev/raw/raw137 /oradata/tpcc50000/stok_0_134
ln -s /dev/raw/raw138 /oradata/tpcc50000/stok_0_135
ln -s /dev/raw/raw139 /oradata/tpcc50000/stok_0_136
ln -s /dev/raw/raw140 /oradata/tpcc50000/stok_0_137
ln -s /dev/raw/raw141 /oradata/tpcc50000/stok_0_138
ln -s /dev/raw/raw142 /oradata/tpcc50000/stok_0_139
ln -s /dev/raw/raw143 /oradata/tpcc50000/stok_0_140
ln -s /dev/raw/raw144 /oradata/tpcc50000/stok_0_141
ln -s /dev/raw/raw145 /oradata/tpcc50000/stok_0_142
ln -s /dev/raw/raw146 /oradata/tpcc50000/stok_0_143
ln -s /dev/raw/raw147 /oradata/tpcc50000/stok_0_144
ln -s /dev/raw/raw148 /oradata/tpcc50000/stok_0_145
ln -s /dev/raw/raw149 /oradata/tpcc50000/stok_0_146
ln -s /dev/raw/raw150 /oradata/tpcc50000/stok_0_147
ln -s /dev/raw/raw151 /oradata/tpcc50000/stok_0_148
ln -s /dev/raw/raw152 /oradata/tpcc50000/stok_0_149
ln -s /dev/raw/raw153 /oradata/tpcc50000/stok_0_150
ln -s /dev/raw/raw154 /oradata/tpcc50000/stok_0_151
ln -s /dev/raw/raw155 /oradata/tpcc50000/stok_0_152
ln -s /dev/raw/raw156 /oradata/tpcc50000/stok_0_153
ln -s /dev/raw/raw157 /oradata/tpcc50000/stok_0_154
ln -s /dev/raw/raw158 /oradata/tpcc50000/stok_0_155
ln -s /dev/raw/raw159 /oradata/tpcc50000/stok_0_156
ln -s /dev/raw/raw160 /oradata/tpcc50000/stok_0_157
ln -s /dev/raw/raw161 /oradata/tpcc50000/stok_0_158
ln -s /dev/raw/raw162 /oradata/tpcc50000/stok_0_159
ln -s /dev/raw/raw163 /oradata/tpcc50000/stok_0_160
ln -s /dev/raw/raw164 /oradata/tpcc50000/stok_0_161
ln -s /dev/raw/raw165 /oradata/tpcc50000/stok_0_162
ln -s /dev/raw/raw166 /oradata/tpcc50000/stok_0_163
ln -s /dev/raw/raw167 /oradata/tpcc50000/stok_0_164
ln -s /dev/raw/raw168 /oradata/tpcc50000/stok_0_165
ln -s /dev/raw/raw169 /oradata/tpcc50000/stok_0_166
ln -s /dev/raw/raw170 /oradata/tpcc50000/stok_0_167
ln -s /dev/raw/raw171 /oradata/tpcc50000/stok_0_168
ln -s /dev/raw/raw172 /oradata/tpcc50000/stok_0_169
ln -s /dev/raw/raw173 /oradata/tpcc50000/stok_0_170
ln -s /dev/raw/raw174 /oradata/tpcc50000/stok_0_171
ln -s /dev/raw/raw175 /oradata/tpcc50000/stok_0_172
ln -s /dev/raw/raw176 /oradata/tpcc50000/stok_0_173
ln -s /dev/raw/raw177 /oradata/tpcc50000/stok_0_174
ln -s /dev/raw/raw178 /oradata/tpcc50000/stok_0_175
ln -s /dev/raw/raw179 /oradata/tpcc50000/stok_0_176
ln -s /dev/raw/raw180 /oradata/tpcc50000/stok_0_177
ln -s /dev/raw/raw181 /oradata/tpcc50000/stok_0_178
ln -s /dev/raw/raw182 /oradata/tpcc50000/stok_0_179
ln -s /dev/raw/raw183 /oradata/tpcc50000/stok_0_180
ln -s /dev/raw/raw184 /oradata/tpcc50000/stok_0_181
ln -s /dev/raw/raw185 /oradata/tpcc50000/stok_0_182
ln -s /dev/raw/raw186 /oradata/tpcc50000/stok_0_183
ln -s /dev/raw/raw187 /oradata/tpcc50000/stok_0_184
ln -s /dev/raw/raw188 /oradata/tpcc50000/stok_0_185
ln -s /dev/raw/raw189 /oradata/tpcc50000/stok_0_186
ln -s /dev/raw/raw190 /oradata/tpcc50000/stok_0_187
ln -s /dev/raw/raw191 /oradata/tpcc50000/stok_0_188
ln -s /dev/raw/raw192 /oradata/tpcc50000/stok_0_189
ln -s /dev/raw/raw193 /oradata/tpcc50000/stok_0_190
ln -s /dev/raw/raw194 /oradata/tpcc50000/stok_0_191
ln -s /dev/raw/raw195 /oradata/tpcc50000/stok_0_192
ln -s /dev/raw/raw196 /oradata/tpcc50000/stok_0_193
ln -s /dev/raw/raw197 /oradata/tpcc50000/stok_0_194
ln -s /dev/raw/raw198 /oradata/tpcc50000/stok_0_195
ln -s /dev/raw/raw199 /oradata/tpcc50000/stok_0_196
ln -s /dev/raw/raw200 /oradata/tpcc50000/stok_0_197
ln -s /dev/raw/raw201 /oradata/tpcc50000/stok_0_198
ln -s /dev/raw/raw202 /oradata/tpcc50000/stok_0_199
ln -s /dev/raw/raw203 /oradata/tpcc50000/stok_0_200
ln -s /dev/raw/raw204 /oradata/tpcc50000/stok_0_201
ln -s /dev/raw/raw205 /oradata/tpcc50000/stok_0_202
ln -s /dev/raw/raw206 /oradata/tpcc50000/stok_0_203
ln -s /dev/raw/raw207 /oradata/tpcc50000/stok_0_204
ln -s /dev/raw/raw208 /oradata/tpcc50000/stok_0_205
ln -s /dev/raw/raw209 /oradata/tpcc50000/stok_0_206
ln -s /dev/raw/raw210 /oradata/tpcc50000/stok_0_207
ln -s /dev/raw/raw211 /oradata/tpcc50000/stok_0_208
ln -s /dev/raw/raw212 /oradata/tpcc50000/stok_0_209
ln -s /dev/raw/raw213 /oradata/tpcc50000/stok_0_210
ln -s /dev/raw/raw214 /oradata/tpcc50000/stok_0_211
```


Copyright © 2004 NEC Corporation

```
ln -s /dev/raw/raw215 /oradata/tpcc50000/stok_0_212
ln -s /dev/raw/raw216 /oradata/tpcc50000/stok_0_213
ln -s /dev/raw/raw217 /oradata/tpcc50000/stok_0_214
ln -s /dev/raw/raw218 /oradata/tpcc50000/stok_0_215
ln -s /dev/raw/raw219 /oradata/tpcc50000/stok_0_216
ln -s /dev/raw/raw220 /oradata/tpcc50000/stok_0_217
ln -s /dev/raw/raw221 /oradata/tpcc50000/stok_0_218
ln -s /dev/raw/raw222 /oradata/tpcc50000/stok_0_219
ln -s /dev/raw/raw223 /oradata/tpcc50000/stok_0_220
ln -s /dev/raw/raw224 /oradata/tpcc50000/stok_0_221
ln -s /dev/raw/raw225 /oradata/tpcc50000/stok_0_222
ln -s /dev/raw/raw226 /oradata/tpcc50000/stok_0_223
ln -s /dev/raw/raw227 /oradata/tpcc50000/stok_0_224
ln -s /dev/raw/raw228 /oradata/tpcc50000/stok_0_225
ln -s /dev/raw/raw229 /oradata/tpcc50000/stok_0_226
ln -s /dev/raw/raw230 /oradata/tpcc50000/stok_0_227
ln -s /dev/raw/raw231 /oradata/tpcc50000/stok_0_228
ln -s /dev/raw/raw232 /oradata/tpcc50000/stok_0_229
ln -s /dev/raw/raw233 /oradata/tpcc50000/stok_0_230
ln -s /dev/raw/raw234 /oradata/tpcc50000/stok_0_231
ln -s /dev/raw/raw235 /oradata/tpcc50000/stok_0_232
ln -s /dev/raw/raw236 /oradata/tpcc50000/stok_0_233
ln -s /dev/raw/raw237 /oradata/tpcc50000/stok_0_234
ln -s /dev/raw/raw238 /oradata/tpcc50000/stok_0_235
ln -s /dev/raw/raw239 /oradata/tpcc50000/stok_0_236
ln -s /dev/raw/raw240 /oradata/tpcc50000/stok_0_237
ln -s /dev/raw/raw241 /oradata/tpcc50000/stok_0_238
ln -s /dev/raw/raw242 /oradata/tpcc50000/stok_0_239
ln -s /dev/raw/raw243 /oradata/tpcc50000/stok_0_240
ln -s /dev/raw/raw244 /oradata/tpcc50000/stok_0_241
ln -s /dev/raw/raw245 /oradata/tpcc50000/stok_0_242
ln -s /dev/raw/raw246 /oradata/tpcc50000/stok_0_243
ln -s /dev/raw/raw247 /oradata/tpcc50000/stok_0_244
ln -s /dev/raw/raw248 /oradata/tpcc50000/stok_0_245
ln -s /dev/raw/raw249 /oradata/tpcc50000/stok_0_246
ln -s /dev/raw/raw250 /oradata/tpcc50000/stok_0_247
ln -s /dev/raw/raw251 /oradata/tpcc50000/stok_0_248
ln -s /dev/raw/raw252 /oradata/tpcc50000/stok_0_249
ln -s /dev/raw/raw253 /oradata/tpcc50000/stok_0_250
ln -s /dev/raw/raw254 /oradata/tpcc50000/stok_0_251
ln -s /dev/raw/raw255 /oradata/tpcc50000/stok_0_252
ln -s /dev/raw/raw256 /oradata/tpcc50000/stok_0_253
ln -s /dev/raw/raw257 /oradata/tpcc50000/stok_0_254
ln -s /dev/raw/raw258 /oradata/tpcc50000/stok_0_255
ln -s /dev/raw/raw259 /oradata/tpcc50000/stok_0_256
ln -s /dev/raw/raw260 /oradata/tpcc50000/stok_0_257
ln -s /dev/raw/raw261 /oradata/tpcc50000/stok_0_258
ln -s /dev/raw/raw262 /oradata/tpcc50000/stok_0_259
ln -s /dev/raw/raw263 /oradata/tpcc50000/stok_0_260
ln -s /dev/raw/raw264 /oradata/tpcc50000/stok_0_261
ln -s /dev/raw/raw265 /oradata/tpcc50000/stok_0_262
ln -s /dev/raw/raw266 /oradata/tpcc50000/stok_0_263
ln -s /dev/raw/raw267 /oradata/tpcc50000/stok_0_264
ln -s /dev/raw/raw268 /oradata/tpcc50000/stok_0_265
ln -s /dev/raw/raw269 /oradata/tpcc50000/stok_0_266
ln -s /dev/raw/raw270 /oradata/tpcc50000/stok_0_267
ln -s /dev/raw/raw271 /oradata/tpcc50000/stok_0_268
ln -s /dev/raw/raw272 /oradata/tpcc50000/stok_0_269
ln -s /dev/raw/raw273 /oradata/tpcc50000/stok_0_270
ln -s /dev/raw/raw274 /oradata/tpcc50000/stok_0_271
ln -s /dev/raw/raw275 /oradata/tpcc50000/stok_0_272
ln -s /dev/raw/raw276 /oradata/tpcc50000/stok_0_273
ln -s /dev/raw/raw277 /oradata/tpcc50000/stok_0_274
ln -s /dev/raw/raw278 /oradata/tpcc50000/stok_0_275
ln -s /dev/raw/raw279 /oradata/tpcc50000/stok_0_276
ln -s /dev/raw/raw280 /oradata/tpcc50000/stok_0_277
ln -s /dev/raw/raw281 /oradata/tpcc50000/stok_0_278
ln -s /dev/raw/raw282 /oradata/tpcc50000/stok_0_279
ln -s /dev/raw/raw283 /oradata/tpcc50000/stok_0_280
ln -s /dev/raw/raw284 /oradata/tpcc50000/stok_0_281
ln -s /dev/raw/raw285 /oradata/tpcc50000/stok_0_282
ln -s /dev/raw/raw286 /oradata/tpcc50000/stok_0_283
ln -s /dev/raw/raw287 /oradata/tpcc50000/stok_0_284
ln -s /dev/raw/raw288 /oradata/tpcc50000/stok_0_285
ln -s /dev/raw/raw289 /oradata/tpcc50000/stok_0_286
ln -s /dev/raw/raw290 /oradata/tpcc50000/stok_0_287
ln -s /dev/raw/raw291 /oradata/tpcc50000/cust_0_0
ln -s /dev/raw/raw292 /oradata/tpcc50000/cust_0_1
ln -s /dev/raw/raw293 /oradata/tpcc50000/cust_0_2
ln -s /dev/raw/raw294 /oradata/tpcc50000/cust_0_3
ln -s /dev/raw/raw295 /oradata/tpcc50000/cust_0_4
ln -s /dev/raw/raw296 /oradata/tpcc50000/cust_0_5
ln -s /dev/raw/raw297 /oradata/tpcc50000/cust_0_6
ln -s /dev/raw/raw298 /oradata/tpcc50000/cust_0_7
ln -s /dev/raw/raw299 /oradata/tpcc50000/cust_0_8
ln -s /dev/raw/raw300 /oradata/tpcc50000/cust_0_9
ln -s /dev/raw/raw301 /oradata/tpcc50000/cust_0_10
ln -s /dev/raw/raw302 /oradata/tpcc50000/cust_0_11
```


Copyright © 2004 NEC Corporation

```
ln -s /dev/raw/raw303 /oradata/tpcc50000/cust_0_12
ln -s /dev/raw/raw304 /oradata/tpcc50000/cust_0_13
ln -s /dev/raw/raw305 /oradata/tpcc50000/cust_0_14
ln -s /dev/raw/raw306 /oradata/tpcc50000/cust_0_15
ln -s /dev/raw/raw307 /oradata/tpcc50000/cust_0_16
ln -s /dev/raw/raw308 /oradata/tpcc50000/cust_0_17
ln -s /dev/raw/raw309 /oradata/tpcc50000/cust_0_18
ln -s /dev/raw/raw310 /oradata/tpcc50000/cust_0_19
ln -s /dev/raw/raw311 /oradata/tpcc50000/cust_0_20
ln -s /dev/raw/raw312 /oradata/tpcc50000/cust_0_21
ln -s /dev/raw/raw313 /oradata/tpcc50000/cust_0_22
ln -s /dev/raw/raw314 /oradata/tpcc50000/cust_0_23
ln -s /dev/raw/raw315 /oradata/tpcc50000/cust_0_24
ln -s /dev/raw/raw316 /oradata/tpcc50000/cust_0_25
ln -s /dev/raw/raw317 /oradata/tpcc50000/cust_0_26
ln -s /dev/raw/raw318 /oradata/tpcc50000/cust_0_27
ln -s /dev/raw/raw319 /oradata/tpcc50000/cust_0_28
ln -s /dev/raw/raw320 /oradata/tpcc50000/cust_0_29
ln -s /dev/raw/raw321 /oradata/tpcc50000/cust_0_30
ln -s /dev/raw/raw322 /oradata/tpcc50000/cust_0_31
ln -s /dev/raw/raw323 /oradata/tpcc50000/cust_0_32
ln -s /dev/raw/raw324 /oradata/tpcc50000/cust_0_33
ln -s /dev/raw/raw325 /oradata/tpcc50000/cust_0_34
ln -s /dev/raw/raw326 /oradata/tpcc50000/cust_0_35
ln -s /dev/raw/raw327 /oradata/tpcc50000/cust_0_36
ln -s /dev/raw/raw328 /oradata/tpcc50000/cust_0_37
ln -s /dev/raw/raw329 /oradata/tpcc50000/cust_0_38
ln -s /dev/raw/raw330 /oradata/tpcc50000/cust_0_39
ln -s /dev/raw/raw331 /oradata/tpcc50000/cust_0_40
ln -s /dev/raw/raw332 /oradata/tpcc50000/cust_0_41
ln -s /dev/raw/raw333 /oradata/tpcc50000/cust_0_42
ln -s /dev/raw/raw334 /oradata/tpcc50000/cust_0_43
ln -s /dev/raw/raw335 /oradata/tpcc50000/cust_0_44
ln -s /dev/raw/raw336 /oradata/tpcc50000/cust_0_45
ln -s /dev/raw/raw337 /oradata/tpcc50000/cust_0_46
ln -s /dev/raw/raw338 /oradata/tpcc50000/cust_0_47
ln -s /dev/raw/raw339 /oradata/tpcc50000/cust_0_48
ln -s /dev/raw/raw340 /oradata/tpcc50000/cust_0_49
ln -s /dev/raw/raw341 /oradata/tpcc50000/cust_0_50
ln -s /dev/raw/raw342 /oradata/tpcc50000/cust_0_51
ln -s /dev/raw/raw343 /oradata/tpcc50000/cust_0_52
ln -s /dev/raw/raw344 /oradata/tpcc50000/cust_0_53
ln -s /dev/raw/raw345 /oradata/tpcc50000/cust_0_54
ln -s /dev/raw/raw346 /oradata/tpcc50000/cust_0_55
ln -s /dev/raw/raw347 /oradata/tpcc50000/cust_0_56
ln -s /dev/raw/raw348 /oradata/tpcc50000/cust_0_57
ln -s /dev/raw/raw349 /oradata/tpcc50000/cust_0_58
ln -s /dev/raw/raw350 /oradata/tpcc50000/cust_0_59
ln -s /dev/raw/raw351 /oradata/tpcc50000/cust_0_60
ln -s /dev/raw/raw352 /oradata/tpcc50000/cust_0_61
ln -s /dev/raw/raw353 /oradata/tpcc50000/cust_0_62
ln -s /dev/raw/raw354 /oradata/tpcc50000/cust_0_63
ln -s /dev/raw/raw355 /oradata/tpcc50000/cust_0_64
ln -s /dev/raw/raw356 /oradata/tpcc50000/cust_0_65
ln -s /dev/raw/raw357 /oradata/tpcc50000/cust_0_66
ln -s /dev/raw/raw358 /oradata/tpcc50000/cust_0_67
ln -s /dev/raw/raw359 /oradata/tpcc50000/cust_0_68
ln -s /dev/raw/raw360 /oradata/tpcc50000/cust_0_69
ln -s /dev/raw/raw361 /oradata/tpcc50000/cust_0_70
ln -s /dev/raw/raw362 /oradata/tpcc50000/cust_0_71
ln -s /dev/raw/raw363 /oradata/tpcc50000/cust_0_72
ln -s /dev/raw/raw364 /oradata/tpcc50000/cust_0_73
ln -s /dev/raw/raw365 /oradata/tpcc50000/cust_0_74
ln -s /dev/raw/raw366 /oradata/tpcc50000/cust_0_75
ln -s /dev/raw/raw367 /oradata/tpcc50000/cust_0_76
ln -s /dev/raw/raw368 /oradata/tpcc50000/cust_0_77
ln -s /dev/raw/raw369 /oradata/tpcc50000/cust_0_78
ln -s /dev/raw/raw370 /oradata/tpcc50000/cust_0_79
ln -s /dev/raw/raw371 /oradata/tpcc50000/cust_0_80
ln -s /dev/raw/raw372 /oradata/tpcc50000/cust_0_81
ln -s /dev/raw/raw373 /oradata/tpcc50000/cust_0_82
ln -s /dev/raw/raw374 /oradata/tpcc50000/cust_0_83
ln -s /dev/raw/raw375 /oradata/tpcc50000/cust_0_84
ln -s /dev/raw/raw376 /oradata/tpcc50000/cust_0_85
ln -s /dev/raw/raw377 /oradata/tpcc50000/cust_0_86
ln -s /dev/raw/raw378 /oradata/tpcc50000/cust_0_87
ln -s /dev/raw/raw379 /oradata/tpcc50000/cust_0_88
ln -s /dev/raw/raw380 /oradata/tpcc50000/cust_0_89
ln -s /dev/raw/raw381 /oradata/tpcc50000/cust_0_90
ln -s /dev/raw/raw382 /oradata/tpcc50000/cust_0_91
ln -s /dev/raw/raw383 /oradata/tpcc50000/cust_0_92
ln -s /dev/raw/raw384 /oradata/tpcc50000/cust_0_93
ln -s /dev/raw/raw385 /oradata/tpcc50000/cust_0_94
ln -s /dev/raw/raw386 /oradata/tpcc50000/cust_0_95
ln -s /dev/raw/raw387 /oradata/tpcc50000/cust_0_96
ln -s /dev/raw/raw388 /oradata/tpcc50000/cust_0_97
ln -s /dev/raw/raw389 /oradata/tpcc50000/cust_0_98
ln -s /dev/raw/raw390 /oradata/tpcc50000/cust_0_99
```


Copyright © 2004 NEC Corporation

```
ln -s /dev/raw/raw391 /oradata/tpcc50000/cust_0_100
ln -s /dev/raw/raw392 /oradata/tpcc50000/cust_0_101
ln -s /dev/raw/raw393 /oradata/tpcc50000/cust_0_102
ln -s /dev/raw/raw394 /oradata/tpcc50000/cust_0_103
ln -s /dev/raw/raw395 /oradata/tpcc50000/cust_0_104
ln -s /dev/raw/raw396 /oradata/tpcc50000/cust_0_105
ln -s /dev/raw/raw397 /oradata/tpcc50000/cust_0_106
ln -s /dev/raw/raw398 /oradata/tpcc50000/cust_0_107
ln -s /dev/raw/raw399 /oradata/tpcc50000/cust_0_108
ln -s /dev/raw/raw400 /oradata/tpcc50000/cust_0_109
ln -s /dev/raw/raw401 /oradata/tpcc50000/cust_0_110
ln -s /dev/raw/raw402 /oradata/tpcc50000/cust_0_111
ln -s /dev/raw/raw403 /oradata/tpcc50000/cust_0_112
ln -s /dev/raw/raw404 /oradata/tpcc50000/cust_0_113
ln -s /dev/raw/raw405 /oradata/tpcc50000/cust_0_114
ln -s /dev/raw/raw406 /oradata/tpcc50000/cust_0_115
ln -s /dev/raw/raw407 /oradata/tpcc50000/cust_0_116
ln -s /dev/raw/raw408 /oradata/tpcc50000/cust_0_117
ln -s /dev/raw/raw409 /oradata/tpcc50000/cust_0_118
ln -s /dev/raw/raw410 /oradata/tpcc50000/cust_0_119
ln -s /dev/raw/raw411 /oradata/tpcc50000/cust_0_120
ln -s /dev/raw/raw412 /oradata/tpcc50000/cust_0_121
ln -s /dev/raw/raw413 /oradata/tpcc50000/cust_0_122
ln -s /dev/raw/raw414 /oradata/tpcc50000/cust_0_123
ln -s /dev/raw/raw415 /oradata/tpcc50000/cust_0_124
ln -s /dev/raw/raw416 /oradata/tpcc50000/cust_0_125
ln -s /dev/raw/raw417 /oradata/tpcc50000/cust_0_126
ln -s /dev/raw/raw418 /oradata/tpcc50000/cust_0_127
ln -s /dev/raw/raw419 /oradata/tpcc50000/cust_0_128
ln -s /dev/raw/raw420 /oradata/tpcc50000/cust_0_129
ln -s /dev/raw/raw421 /oradata/tpcc50000/cust_0_130
ln -s /dev/raw/raw422 /oradata/tpcc50000/cust_0_131
ln -s /dev/raw/raw423 /oradata/tpcc50000/cust_0_132
ln -s /dev/raw/raw424 /oradata/tpcc50000/cust_0_133
ln -s /dev/raw/raw425 /oradata/tpcc50000/cust_0_134
ln -s /dev/raw/raw426 /oradata/tpcc50000/cust_0_135
ln -s /dev/raw/raw427 /oradata/tpcc50000/cust_0_136
ln -s /dev/raw/raw428 /oradata/tpcc50000/cust_0_137
ln -s /dev/raw/raw429 /oradata/tpcc50000/cust_0_138
ln -s /dev/raw/raw430 /oradata/tpcc50000/cust_0_139
ln -s /dev/raw/raw431 /oradata/tpcc50000/cust_0_140
ln -s /dev/raw/raw432 /oradata/tpcc50000/cust_0_141
ln -s /dev/raw/raw433 /oradata/tpcc50000/cust_0_142
ln -s /dev/raw/raw434 /oradata/tpcc50000/cust_0_143
ln -s /dev/raw/raw435 /oradata/tpcc50000/cust_0_144
ln -s /dev/raw/raw436 /oradata/tpcc50000/cust_0_145
ln -s /dev/raw/raw437 /oradata/tpcc50000/cust_0_146
ln -s /dev/raw/raw438 /oradata/tpcc50000/cust_0_147
ln -s /dev/raw/raw439 /oradata/tpcc50000/cust_0_148
ln -s /dev/raw/raw440 /oradata/tpcc50000/cust_0_149
ln -s /dev/raw/raw441 /oradata/tpcc50000/cust_0_150
ln -s /dev/raw/raw442 /oradata/tpcc50000/cust_0_151
ln -s /dev/raw/raw443 /oradata/tpcc50000/cust_0_152
ln -s /dev/raw/raw444 /oradata/tpcc50000/cust_0_153
ln -s /dev/raw/raw445 /oradata/tpcc50000/cust_0_154
ln -s /dev/raw/raw446 /oradata/tpcc50000/cust_0_155
ln -s /dev/raw/raw447 /oradata/tpcc50000/cust_0_156
ln -s /dev/raw/raw448 /oradata/tpcc50000/cust_0_157
ln -s /dev/raw/raw449 /oradata/tpcc50000/cust_0_158
ln -s /dev/raw/raw450 /oradata/tpcc50000/cust_0_159
ln -s /dev/raw/raw451 /oradata/tpcc50000/cust_0_160
ln -s /dev/raw/raw452 /oradata/tpcc50000/cust_0_161
ln -s /dev/raw/raw453 /oradata/tpcc50000/cust_0_162
ln -s /dev/raw/raw454 /oradata/tpcc50000/cust_0_163
ln -s /dev/raw/raw455 /oradata/tpcc50000/cust_0_164
ln -s /dev/raw/raw456 /oradata/tpcc50000/cust_0_165
ln -s /dev/raw/raw457 /oradata/tpcc50000/cust_0_166
ln -s /dev/raw/raw458 /oradata/tpcc50000/cust_0_167
ln -s /dev/raw/raw459 /oradata/tpcc50000/cust_0_168
ln -s /dev/raw/raw460 /oradata/tpcc50000/cust_0_169
ln -s /dev/raw/raw461 /oradata/tpcc50000/cust_0_170
ln -s /dev/raw/raw462 /oradata/tpcc50000/cust_0_171
ln -s /dev/raw/raw463 /oradata/tpcc50000/cust_0_172
ln -s /dev/raw/raw464 /oradata/tpcc50000/cust_0_173
ln -s /dev/raw/raw465 /oradata/tpcc50000/cust_0_174
ln -s /dev/raw/raw466 /oradata/tpcc50000/cust_0_175
ln -s /dev/raw/raw467 /oradata/tpcc50000/cust_0_176
ln -s /dev/raw/raw468 /oradata/tpcc50000/cust_0_177
ln -s /dev/raw/raw469 /oradata/tpcc50000/cust_0_178
ln -s /dev/raw/raw470 /oradata/tpcc50000/cust_0_179
ln -s /dev/raw/raw471 /oradata/tpcc50000/cust_0_180
ln -s /dev/raw/raw472 /oradata/tpcc50000/cust_0_181
ln -s /dev/raw/raw473 /oradata/tpcc50000/cust_0_182
ln -s /dev/raw/raw474 /oradata/tpcc50000/cust_0_183
ln -s /dev/raw/raw475 /oradata/tpcc50000/cust_0_184
ln -s /dev/raw/raw476 /oradata/tpcc50000/cust_0_185
ln -s /dev/raw/raw477 /oradata/tpcc50000/cust_0_186
ln -s /dev/raw/raw478 /oradata/tpcc50000/cust_0_187
```


Copyright © 2004 NEC Corporation

```
ln -s /dev/raw/raw479 /oradata/tpcc50000/cust_0_188
ln -s /dev/raw/raw480 /oradata/tpcc50000/cust_0_189
ln -s /dev/raw/raw481 /oradata/tpcc50000/cust_0_190
ln -s /dev/raw/raw482 /oradata/tpcc50000/cust_0_191
ln -s /dev/raw/raw483 /oradata/tpcc50000/cust_0_192
ln -s /dev/raw/raw484 /oradata/tpcc50000/cust_0_193
ln -s /dev/raw/raw485 /oradata/tpcc50000/cust_0_194
ln -s /dev/raw/raw486 /oradata/tpcc50000/cust_0_195
ln -s /dev/raw/raw487 /oradata/tpcc50000/cust_0_196
ln -s /dev/raw/raw488 /oradata/tpcc50000/cust_0_197
ln -s /dev/raw/raw489 /oradata/tpcc50000/cust_0_198
ln -s /dev/raw/raw490 /oradata/tpcc50000/cust_0_199
ln -s /dev/raw/raw491 /oradata/tpcc50000/cust_0_200
ln -s /dev/raw/raw492 /oradata/tpcc50000/cust_0_201
ln -s /dev/raw/raw493 /oradata/tpcc50000/cust_0_202
ln -s /dev/raw/raw494 /oradata/tpcc50000/cust_0_203
ln -s /dev/raw/raw495 /oradata/tpcc50000/cust_0_204
ln -s /dev/raw/raw496 /oradata/tpcc50000/cust_0_205
ln -s /dev/raw/raw497 /oradata/tpcc50000/cust_0_206
ln -s /dev/raw/raw498 /oradata/tpcc50000/cust_0_207
ln -s /dev/raw/raw499 /oradata/tpcc50000/cust_0_208
ln -s /dev/raw/raw500 /oradata/tpcc50000/cust_0_209
ln -s /dev/raw/raw501 /oradata/tpcc50000/cust_0_210
ln -s /dev/raw/raw502 /oradata/tpcc50000/cust_0_211
ln -s /dev/raw/raw503 /oradata/tpcc50000/cust_0_212
ln -s /dev/raw/raw504 /oradata/tpcc50000/cust_0_213
ln -s /dev/raw/raw505 /oradata/tpcc50000/cust_0_214
ln -s /dev/raw/raw506 /oradata/tpcc50000/cust_0_215
ln -s /dev/raw/raw507 /oradata/tpcc50000/hist_0_0
ln -s /dev/raw/raw508 /oradata/tpcc50000/hist_0_1
ln -s /dev/raw/raw509 /oradata/tpcc50000/hist_0_2
ln -s /dev/raw/raw510 /oradata/tpcc50000/hist_0_3
ln -s /dev/raw/raw511 /oradata/tpcc50000/hist_0_4
ln -s /dev/raw/raw512 /oradata/tpcc50000/hist_0_5
ln -s /dev/raw/raw513 /oradata/tpcc50000/hist_0_6
ln -s /dev/raw/raw514 /oradata/tpcc50000/hist_0_7
ln -s /dev/raw/raw515 /oradata/tpcc50000/hist_0_8
ln -s /dev/raw/raw516 /oradata/tpcc50000/hist_0_9
ln -s /dev/raw/raw517 /oradata/tpcc50000/hist_0_10
ln -s /dev/raw/raw518 /oradata/tpcc50000/hist_0_11
ln -s /dev/raw/raw519 /oradata/tpcc50000/hist_0_12
ln -s /dev/raw/raw520 /oradata/tpcc50000/hist_0_13
ln -s /dev/raw/raw521 /oradata/tpcc50000/hist_0_14
ln -s /dev/raw/raw522 /oradata/tpcc50000/hist_0_15
ln -s /dev/raw/raw523 /oradata/tpcc50000/hist_0_16
ln -s /dev/raw/raw524 /oradata/tpcc50000/hist_0_17
ln -s /dev/raw/raw525 /oradata/tpcc50000/hist_0_18
ln -s /dev/raw/raw526 /oradata/tpcc50000/hist_0_19
ln -s /dev/raw/raw527 /oradata/tpcc50000/hist_0_20
ln -s /dev/raw/raw528 /oradata/tpcc50000/hist_0_21
ln -s /dev/raw/raw529 /oradata/tpcc50000/hist_0_22
ln -s /dev/raw/raw530 /oradata/tpcc50000/hist_0_23
ln -s /dev/raw/raw531 /oradata/tpcc50000/hist_0_24
ln -s /dev/raw/raw532 /oradata/tpcc50000/hist_0_25
ln -s /dev/raw/raw533 /oradata/tpcc50000/hist_0_26
ln -s /dev/raw/raw534 /oradata/tpcc50000/hist_0_27
ln -s /dev/raw/raw535 /oradata/tpcc50000/hist_0_28
ln -s /dev/raw/raw536 /oradata/tpcc50000/hist_0_29
ln -s /dev/raw/raw537 /oradata/tpcc50000/hist_0_30
ln -s /dev/raw/raw538 /oradata/tpcc50000/hist_0_31
ln -s /dev/raw/raw539 /oradata/tpcc50000/hist_0_32
ln -s /dev/raw/raw540 /oradata/tpcc50000/hist_0_33
ln -s /dev/raw/raw541 /oradata/tpcc50000/hist_0_34
ln -s /dev/raw/raw542 /oradata/tpcc50000/hist_0_35
ln -s /dev/raw/raw543 /oradata/tpcc50000/hist_0_36
ln -s /dev/raw/raw544 /oradata/tpcc50000/hist_0_37
ln -s /dev/raw/raw545 /oradata/tpcc50000/hist_0_38
ln -s /dev/raw/raw546 /oradata/tpcc50000/hist_0_39
ln -s /dev/raw/raw547 /oradata/tpcc50000/hist_0_40
ln -s /dev/raw/raw548 /oradata/tpcc50000/hist_0_41
ln -s /dev/raw/raw549 /oradata/tpcc50000/hist_0_42
ln -s /dev/raw/raw550 /oradata/tpcc50000/hist_0_43
ln -s /dev/raw/raw551 /oradata/tpcc50000/hist_0_44
ln -s /dev/raw/raw552 /oradata/tpcc50000/hist_0_45
ln -s /dev/raw/raw553 /oradata/tpcc50000/hist_0_46
ln -s /dev/raw/raw554 /oradata/tpcc50000/hist_0_47
ln -s /dev/raw/raw555 /oradata/tpcc50000/hist_0_48
ln -s /dev/raw/raw556 /oradata/tpcc50000/hist_0_49
ln -s /dev/raw/raw557 /oradata/tpcc50000/hist_0_50
ln -s /dev/raw/raw558 /oradata/tpcc50000/hist_0_51
ln -s /dev/raw/raw559 /oradata/tpcc50000/hist_0_52
ln -s /dev/raw/raw560 /oradata/tpcc50000/hist_0_53
ln -s /dev/raw/raw561 /oradata/tpcc50000/hist_0_54
ln -s /dev/raw/raw562 /oradata/tpcc50000/hist_0_55
ln -s /dev/raw/raw563 /oradata/tpcc50000/hist_0_56
ln -s /dev/raw/raw564 /oradata/tpcc50000/hist_0_57
ln -s /dev/raw/raw565 /oradata/tpcc50000/hist_0_58
ln -s /dev/raw/raw566 /oradata/tpcc50000/hist_0_59
```


Copyright © 2004 NEC Corporation

```
ln -s /dev/raw/raw567 /oradata/tpcc50000/hist_0_60
ln -s /dev/raw/raw568 /oradata/tpcc50000/hist_0_61
ln -s /dev/raw/raw569 /oradata/tpcc50000/hist_0_62
ln -s /dev/raw/raw570 /oradata/tpcc50000/hist_0_63
ln -s /dev/raw/raw571 /oradata/tpcc50000/hist_0_64
ln -s /dev/raw/raw572 /oradata/tpcc50000/hist_0_65
ln -s /dev/raw/raw573 /oradata/tpcc50000/hist_0_66
ln -s /dev/raw/raw574 /oradata/tpcc50000/hist_0_67
ln -s /dev/raw/raw575 /oradata/tpcc50000/hist_0_68
ln -s /dev/raw/raw576 /oradata/tpcc50000/hist_0_69
ln -s /dev/raw/raw577 /oradata/tpcc50000/hist_0_70
ln -s /dev/raw/raw578 /oradata/tpcc50000/hist_0_71
ln -s /dev/raw/raw579 /oradata/tpcc50000/ordr_0_0
ln -s /dev/raw/raw580 /oradata/tpcc50000/ordr_0_1
ln -s /dev/raw/raw581 /oradata/tpcc50000/ordr_0_2
ln -s /dev/raw/raw582 /oradata/tpcc50000/ordr_0_3
ln -s /dev/raw/raw583 /oradata/tpcc50000/ordr_0_4
ln -s /dev/raw/raw584 /oradata/tpcc50000/ordr_0_5
ln -s /dev/raw/raw585 /oradata/tpcc50000/ordr_0_6
ln -s /dev/raw/raw586 /oradata/tpcc50000/ordr_0_7
ln -s /dev/raw/raw587 /oradata/tpcc50000/ordr_0_8
ln -s /dev/raw/raw588 /oradata/tpcc50000/ordr_0_9
ln -s /dev/raw/raw589 /oradata/tpcc50000/ordr_0_10
ln -s /dev/raw/raw590 /oradata/tpcc50000/ordr_0_11
ln -s /dev/raw/raw591 /oradata/tpcc50000/ordr_0_12
ln -s /dev/raw/raw592 /oradata/tpcc50000/ordr_0_13
ln -s /dev/raw/raw593 /oradata/tpcc50000/ordr_0_14
ln -s /dev/raw/raw594 /oradata/tpcc50000/ordr_0_15
ln -s /dev/raw/raw595 /oradata/tpcc50000/ordr_0_16
ln -s /dev/raw/raw596 /oradata/tpcc50000/ordr_0_17
ln -s /dev/raw/raw597 /oradata/tpcc50000/ordr_0_18
ln -s /dev/raw/raw598 /oradata/tpcc50000/ordr_0_19
ln -s /dev/raw/raw599 /oradata/tpcc50000/ordr_0_20
ln -s /dev/raw/raw600 /oradata/tpcc50000/ordr_0_21
ln -s /dev/raw/raw601 /oradata/tpcc50000/ordr_0_22
ln -s /dev/raw/raw602 /oradata/tpcc50000/ordr_0_23
ln -s /dev/raw/raw603 /oradata/tpcc50000/ordr_0_24
ln -s /dev/raw/raw604 /oradata/tpcc50000/ordr_0_25
ln -s /dev/raw/raw605 /oradata/tpcc50000/ordr_0_26
ln -s /dev/raw/raw606 /oradata/tpcc50000/ordr_0_27
ln -s /dev/raw/raw607 /oradata/tpcc50000/ordr_0_28
ln -s /dev/raw/raw608 /oradata/tpcc50000/ordr_0_29
ln -s /dev/raw/raw609 /oradata/tpcc50000/ordr_0_30
ln -s /dev/raw/raw610 /oradata/tpcc50000/ordr_0_31
ln -s /dev/raw/raw611 /oradata/tpcc50000/ordr_0_32
ln -s /dev/raw/raw612 /oradata/tpcc50000/ordr_0_33
ln -s /dev/raw/raw613 /oradata/tpcc50000/ordr_0_34
ln -s /dev/raw/raw614 /oradata/tpcc50000/ordr_0_35
ln -s /dev/raw/raw615 /oradata/tpcc50000/ordr_0_36
ln -s /dev/raw/raw616 /oradata/tpcc50000/ordr_0_37
ln -s /dev/raw/raw617 /oradata/tpcc50000/ordr_0_38
ln -s /dev/raw/raw618 /oradata/tpcc50000/ordr_0_39
ln -s /dev/raw/raw619 /oradata/tpcc50000/ordr_0_40
ln -s /dev/raw/raw620 /oradata/tpcc50000/ordr_0_41
ln -s /dev/raw/raw621 /oradata/tpcc50000/ordr_0_42
ln -s /dev/raw/raw622 /oradata/tpcc50000/ordr_0_43
ln -s /dev/raw/raw623 /oradata/tpcc50000/ordr_0_44
ln -s /dev/raw/raw624 /oradata/tpcc50000/ordr_0_45
ln -s /dev/raw/raw625 /oradata/tpcc50000/ordr_0_46
ln -s /dev/raw/raw626 /oradata/tpcc50000/ordr_0_47
ln -s /dev/raw/raw627 /oradata/tpcc50000/ordr_0_48
ln -s /dev/raw/raw628 /oradata/tpcc50000/ordr_0_49
ln -s /dev/raw/raw629 /oradata/tpcc50000/ordr_0_50
ln -s /dev/raw/raw630 /oradata/tpcc50000/ordr_0_51
ln -s /dev/raw/raw631 /oradata/tpcc50000/ordr_0_52
ln -s /dev/raw/raw632 /oradata/tpcc50000/ordr_0_53
ln -s /dev/raw/raw633 /oradata/tpcc50000/ordr_0_54
ln -s /dev/raw/raw634 /oradata/tpcc50000/ordr_0_55
ln -s /dev/raw/raw635 /oradata/tpcc50000/ordr_0_56
ln -s /dev/raw/raw636 /oradata/tpcc50000/ordr_0_57
ln -s /dev/raw/raw637 /oradata/tpcc50000/ordr_0_58
ln -s /dev/raw/raw638 /oradata/tpcc50000/ordr_0_59
ln -s /dev/raw/raw639 /oradata/tpcc50000/ordr_0_60
ln -s /dev/raw/raw640 /oradata/tpcc50000/ordr_0_61
ln -s /dev/raw/raw641 /oradata/tpcc50000/ordr_0_62
ln -s /dev/raw/raw642 /oradata/tpcc50000/ordr_0_63
ln -s /dev/raw/raw643 /oradata/tpcc50000/ordr_0_64
ln -s /dev/raw/raw644 /oradata/tpcc50000/ordr_0_65
ln -s /dev/raw/raw645 /oradata/tpcc50000/ordr_0_66
ln -s /dev/raw/raw646 /oradata/tpcc50000/ordr_0_67
ln -s /dev/raw/raw647 /oradata/tpcc50000/ordr_0_68
ln -s /dev/raw/raw648 /oradata/tpcc50000/ordr_0_69
ln -s /dev/raw/raw649 /oradata/tpcc50000/ordr_0_70
ln -s /dev/raw/raw650 /oradata/tpcc50000/ordr_0_71
ln -s /dev/raw/raw651 /oradata/tpcc50000/icust2_0_0
ln -s /dev/raw/raw652 /oradata/tpcc50000/icust2_0_1
ln -s /dev/raw/raw653 /oradata/tpcc50000/icust2_0_2
ln -s /dev/raw/raw654 /oradata/tpcc50000/icust2_0_3
```


Copyright © 2004 NEC Corporation

```
ln -s /dev/raw/raw655 /oradata/tpcc50000/icust2_0_4
ln -s /dev/raw/raw656 /oradata/tpcc50000/icust2_0_5
ln -s /dev/raw/raw657 /oradata/tpcc50000/icust2_0_6
ln -s /dev/raw/raw658 /oradata/tpcc50000/icust2_0_7
ln -s /dev/raw/raw659 /oradata/tpcc50000/icust2_0_8
ln -s /dev/raw/raw660 /oradata/tpcc50000/icust2_0_9
ln -s /dev/raw/raw661 /oradata/tpcc50000/icust2_0_10
ln -s /dev/raw/raw662 /oradata/tpcc50000/icust2_0_11
ln -s /dev/raw/raw663 /oradata/tpcc50000/icust2_0_12
ln -s /dev/raw/raw664 /oradata/tpcc50000/icust2_0_13
ln -s /dev/raw/raw665 /oradata/tpcc50000/icust2_0_14
ln -s /dev/raw/raw666 /oradata/tpcc50000/icust2_0_15
ln -s /dev/raw/raw667 /oradata/tpcc50000/icust2_0_16
ln -s /dev/raw/raw668 /oradata/tpcc50000/icust2_0_17
ln -s /dev/raw/raw669 /oradata/tpcc50000/icust2_0_18
ln -s /dev/raw/raw670 /oradata/tpcc50000/icust2_0_19
ln -s /dev/raw/raw671 /oradata/tpcc50000/icust2_0_20
ln -s /dev/raw/raw672 /oradata/tpcc50000/icust2_0_21
ln -s /dev/raw/raw673 /oradata/tpcc50000/icust2_0_22
ln -s /dev/raw/raw674 /oradata/tpcc50000/icust2_0_23
ln -s /dev/raw/raw675 /oradata/tpcc50000/icust2_0_24
ln -s /dev/raw/raw676 /oradata/tpcc50000/icust2_0_25
ln -s /dev/raw/raw677 /oradata/tpcc50000/icust2_0_26
ln -s /dev/raw/raw678 /oradata/tpcc50000/icust2_0_27
ln -s /dev/raw/raw679 /oradata/tpcc50000/icust2_0_28
ln -s /dev/raw/raw680 /oradata/tpcc50000/icust2_0_29
ln -s /dev/raw/raw681 /oradata/tpcc50000/icust2_0_30
ln -s /dev/raw/raw682 /oradata/tpcc50000/icust2_0_31
ln -s /dev/raw/raw683 /oradata/tpcc50000/icust2_0_32
ln -s /dev/raw/raw684 /oradata/tpcc50000/icust2_0_33
ln -s /dev/raw/raw685 /oradata/tpcc50000/icust2_0_34
ln -s /dev/raw/raw686 /oradata/tpcc50000/icust2_0_35
ln -s /dev/raw/raw687 /oradata/tpcc50000/icust2_0_36
ln -s /dev/raw/raw688 /oradata/tpcc50000/icust2_0_37
ln -s /dev/raw/raw689 /oradata/tpcc50000/icust2_0_38
ln -s /dev/raw/raw690 /oradata/tpcc50000/icust2_0_39
ln -s /dev/raw/raw691 /oradata/tpcc50000/icust2_0_40
ln -s /dev/raw/raw692 /oradata/tpcc50000/icust2_0_41
ln -s /dev/raw/raw693 /oradata/tpcc50000/icust2_0_42
ln -s /dev/raw/raw694 /oradata/tpcc50000/icust2_0_43
ln -s /dev/raw/raw695 /oradata/tpcc50000/icust2_0_44
ln -s /dev/raw/raw696 /oradata/tpcc50000/icust2_0_45
ln -s /dev/raw/raw697 /oradata/tpcc50000/icust2_0_46
ln -s /dev/raw/raw698 /oradata/tpcc50000/icust2_0_47
ln -s /dev/raw/raw699 /oradata/tpcc50000/icust2_0_48
ln -s /dev/raw/raw700 /oradata/tpcc50000/icust2_0_49
ln -s /dev/raw/raw701 /oradata/tpcc50000/icust2_0_50
ln -s /dev/raw/raw702 /oradata/tpcc50000/icust2_0_51
ln -s /dev/raw/raw703 /oradata/tpcc50000/icust2_0_52
ln -s /dev/raw/raw704 /oradata/tpcc50000/icust2_0_53
ln -s /dev/raw/raw705 /oradata/tpcc50000/icust2_0_54
ln -s /dev/raw/raw706 /oradata/tpcc50000/icust2_0_55
ln -s /dev/raw/raw707 /oradata/tpcc50000/icust2_0_56
ln -s /dev/raw/raw708 /oradata/tpcc50000/icust2_0_57
ln -s /dev/raw/raw709 /oradata/tpcc50000/icust2_0_58
ln -s /dev/raw/raw710 /oradata/tpcc50000/icust2_0_59
ln -s /dev/raw/raw711 /oradata/tpcc50000/icust2_0_60
ln -s /dev/raw/raw712 /oradata/tpcc50000/icust2_0_61
ln -s /dev/raw/raw713 /oradata/tpcc50000/icust2_0_62
ln -s /dev/raw/raw714 /oradata/tpcc50000/icust2_0_63
ln -s /dev/raw/raw715 /oradata/tpcc50000/icust2_0_64
ln -s /dev/raw/raw716 /oradata/tpcc50000/icust2_0_65
ln -s /dev/raw/raw717 /oradata/tpcc50000/icust2_0_66
ln -s /dev/raw/raw718 /oradata/tpcc50000/icust2_0_67
ln -s /dev/raw/raw719 /oradata/tpcc50000/icust2_0_68
ln -s /dev/raw/raw720 /oradata/tpcc50000/icust2_0_69
ln -s /dev/raw/raw721 /oradata/tpcc50000/icust2_0_70
ln -s /dev/raw/raw722 /oradata/tpcc50000/icust2_0_71
ln -s /dev/raw/raw723 /oradata/tpcc50000/iordr2_0_0
ln -s /dev/raw/raw724 /oradata/tpcc50000/iordr2_0_1
ln -s /dev/raw/raw725 /oradata/tpcc50000/iordr2_0_2
ln -s /dev/raw/raw726 /oradata/tpcc50000/iordr2_0_3
ln -s /dev/raw/raw727 /oradata/tpcc50000/iordr2_0_4
ln -s /dev/raw/raw728 /oradata/tpcc50000/iordr2_0_5
ln -s /dev/raw/raw729 /oradata/tpcc50000/iordr2_0_6
ln -s /dev/raw/raw730 /oradata/tpcc50000/iordr2_0_7
ln -s /dev/raw/raw731 /oradata/tpcc50000/iordr2_0_8
ln -s /dev/raw/raw732 /oradata/tpcc50000/iordr2_0_9
ln -s /dev/raw/raw733 /oradata/tpcc50000/iordr2_0_10
ln -s /dev/raw/raw734 /oradata/tpcc50000/iordr2_0_11
ln -s /dev/raw/raw735 /oradata/tpcc50000/iordr2_0_12
ln -s /dev/raw/raw736 /oradata/tpcc50000/iordr2_0_13
ln -s /dev/raw/raw737 /oradata/tpcc50000/iordr2_0_14
ln -s /dev/raw/raw738 /oradata/tpcc50000/iordr2_0_15
ln -s /dev/raw/raw739 /oradata/tpcc50000/iordr2_0_16
ln -s /dev/raw/raw740 /oradata/tpcc50000/iordr2_0_17
ln -s /dev/raw/raw741 /oradata/tpcc50000/iordr2_0_18
ln -s /dev/raw/raw742 /oradata/tpcc50000/iordr2_0_19
```


Copyright © 2004 NEC Corporation

```
ln -s /dev/raw/raw743 /oradata/tpcc50000/iordr2_0_20
ln -s /dev/raw/raw744 /oradata/tpcc50000/iordr2_0_21
ln -s /dev/raw/raw745 /oradata/tpcc50000/iordr2_0_22
ln -s /dev/raw/raw746 /oradata/tpcc50000/iordr2_0_23
ln -s /dev/raw/raw747 /oradata/tpcc50000/iordr2_0_24
ln -s /dev/raw/raw748 /oradata/tpcc50000/iordr2_0_25
ln -s /dev/raw/raw749 /oradata/tpcc50000/iordr2_0_26
ln -s /dev/raw/raw750 /oradata/tpcc50000/iordr2_0_27
ln -s /dev/raw/raw751 /oradata/tpcc50000/iordr2_0_28
ln -s /dev/raw/raw752 /oradata/tpcc50000/iordr2_0_29
ln -s /dev/raw/raw753 /oradata/tpcc50000/iordr2_0_30
ln -s /dev/raw/raw754 /oradata/tpcc50000/iordr2_0_31
ln -s /dev/raw/raw755 /oradata/tpcc50000/iordr2_0_32
ln -s /dev/raw/raw756 /oradata/tpcc50000/iordr2_0_33
ln -s /dev/raw/raw757 /oradata/tpcc50000/iordr2_0_34
ln -s /dev/raw/raw758 /oradata/tpcc50000/iordr2_0_35
ln -s /dev/raw/raw759 /oradata/tpcc50000/iordr2_0_36
ln -s /dev/raw/raw760 /oradata/tpcc50000/iordr2_0_37
ln -s /dev/raw/raw761 /oradata/tpcc50000/iordr2_0_38
ln -s /dev/raw/raw762 /oradata/tpcc50000/iordr2_0_39
ln -s /dev/raw/raw763 /oradata/tpcc50000/iordr2_0_40
ln -s /dev/raw/raw764 /oradata/tpcc50000/iordr2_0_41
ln -s /dev/raw/raw765 /oradata/tpcc50000/iordr2_0_42
ln -s /dev/raw/raw766 /oradata/tpcc50000/iordr2_0_43
ln -s /dev/raw/raw767 /oradata/tpcc50000/iordr2_0_44
ln -s /dev/raw/raw768 /oradata/tpcc50000/iordr2_0_45
ln -s /dev/raw/raw769 /oradata/tpcc50000/iordr2_0_46
ln -s /dev/raw/raw770 /oradata/tpcc50000/iordr2_0_47
ln -s /dev/raw/raw771 /oradata/tpcc50000/iordr2_0_48
ln -s /dev/raw/raw772 /oradata/tpcc50000/iordr2_0_49
ln -s /dev/raw/raw773 /oradata/tpcc50000/iordr2_0_50
ln -s /dev/raw/raw774 /oradata/tpcc50000/iordr2_0_51
ln -s /dev/raw/raw775 /oradata/tpcc50000/iordr2_0_52
ln -s /dev/raw/raw776 /oradata/tpcc50000/iordr2_0_53
ln -s /dev/raw/raw777 /oradata/tpcc50000/iordr2_0_54
ln -s /dev/raw/raw778 /oradata/tpcc50000/iordr2_0_55
ln -s /dev/raw/raw779 /oradata/tpcc50000/iordr2_0_56
ln -s /dev/raw/raw780 /oradata/tpcc50000/iordr2_0_57
ln -s /dev/raw/raw781 /oradata/tpcc50000/iordr2_0_58
ln -s /dev/raw/raw782 /oradata/tpcc50000/iordr2_0_59
ln -s /dev/raw/raw783 /oradata/tpcc50000/iordr2_0_60
ln -s /dev/raw/raw784 /oradata/tpcc50000/iordr2_0_61
ln -s /dev/raw/raw785 /oradata/tpcc50000/iordr2_0_62
ln -s /dev/raw/raw786 /oradata/tpcc50000/iordr2_0_63
ln -s /dev/raw/raw787 /oradata/tpcc50000/iordr2_0_64
ln -s /dev/raw/raw788 /oradata/tpcc50000/iordr2_0_65
ln -s /dev/raw/raw789 /oradata/tpcc50000/iordr2_0_66
ln -s /dev/raw/raw790 /oradata/tpcc50000/iordr2_0_67
ln -s /dev/raw/raw791 /oradata/tpcc50000/iordr2_0_68
ln -s /dev/raw/raw792 /oradata/tpcc50000/iordr2_0_69
ln -s /dev/raw/raw793 /oradata/tpcc50000/iordr2_0_70
ln -s /dev/raw/raw794 /oradata/tpcc50000/iordr2_0_71
ln -s /dev/raw/raw795 /oradata/tpcc50000/reserve_0_0
ln -s /dev/raw/raw796 /oradata/tpcc50000/reserve_0_1
ln -s /dev/raw/raw797 /oradata/tpcc50000/reserve_0_2
ln -s /dev/raw/raw798 /oradata/tpcc50000/reserve_0_3
ln -s /dev/raw/raw799 /oradata/tpcc50000/reserve_0_4
ln -s /dev/raw/raw800 /oradata/tpcc50000/reserve_0_5
ln -s /dev/raw/raw801 /oradata/tpcc50000/reserve_0_6
ln -s /dev/raw/raw802 /oradata/tpcc50000/reserve_0_7
ln -s /dev/raw/raw803 /oradata/tpcc50000/reserve_0_8
ln -s /dev/raw/raw804 /oradata/tpcc50000/reserve_0_9
ln -s /dev/raw/raw805 /oradata/tpcc50000/reserve_0_10
ln -s /dev/raw/raw806 /oradata/tpcc50000/reserve_0_11
ln -s /dev/raw/raw807 /oradata/tpcc50000/reserve_0_12
ln -s /dev/raw/raw808 /oradata/tpcc50000/reserve_0_13
ln -s /dev/raw/raw809 /oradata/tpcc50000/reserve_0_14
ln -s /dev/raw/raw810 /oradata/tpcc50000/reserve_0_15
ln -s /dev/raw/raw811 /oradata/tpcc50000/reserve_0_16
ln -s /dev/raw/raw812 /oradata/tpcc50000/reserve_0_17
ln -s /dev/raw/raw813 /oradata/tpcc50000/reserve_0_18
ln -s /dev/raw/raw814 /oradata/tpcc50000/reserve_0_19
ln -s /dev/raw/raw815 /oradata/tpcc50000/reserve_0_20
ln -s /dev/raw/raw816 /oradata/tpcc50000/reserve_0_21
ln -s /dev/raw/raw817 /oradata/tpcc50000/reserve_0_22
ln -s /dev/raw/raw818 /oradata/tpcc50000/reserve_0_23
ln -s /dev/raw/raw819 /oradata/tpcc50000/reserve_0_24
ln -s /dev/raw/raw820 /oradata/tpcc50000/reserve_0_25
ln -s /dev/raw/raw821 /oradata/tpcc50000/reserve_0_26
ln -s /dev/raw/raw822 /oradata/tpcc50000/reserve_0_27
ln -s /dev/raw/raw823 /oradata/tpcc50000/reserve_0_28
ln -s /dev/raw/raw824 /oradata/tpcc50000/reserve_0_29
ln -s /dev/raw/raw825 /oradata/tpcc50000/reserve_0_30
ln -s /dev/raw/raw826 /oradata/tpcc50000/reserve_0_31
ln -s /dev/raw/raw827 /oradata/tpcc50000/reserve_0_32
ln -s /dev/raw/raw828 /oradata/tpcc50000/reserve_0_33
ln -s /dev/raw/raw829 /oradata/tpcc50000/reserve_0_34
ln -s /dev/raw/raw830 /oradata/tpcc50000/reserve_0_35
```


Copyright © 2004 NEC Corporation

```
ln -s /dev/raw/raw831 /oradata/tpcc50000/reserve_0_36
ln -s /dev/raw/raw832 /oradata/tpcc50000/reserve_0_37
ln -s /dev/raw/raw833 /oradata/tpcc50000/reserve_0_38
ln -s /dev/raw/raw834 /oradata/tpcc50000/reserve_0_39
ln -s /dev/raw/raw835 /oradata/tpcc50000/reserve_0_40
ln -s /dev/raw/raw836 /oradata/tpcc50000/reserve_0_41
ln -s /dev/raw/raw837 /oradata/tpcc50000/reserve_0_42
ln -s /dev/raw/raw838 /oradata/tpcc50000/reserve_0_43
ln -s /dev/raw/raw839 /oradata/tpcc50000/reserve_0_44
ln -s /dev/raw/raw840 /oradata/tpcc50000/reserve_0_45
ln -s /dev/raw/raw841 /oradata/tpcc50000/reserve_0_46
ln -s /dev/raw/raw842 /oradata/tpcc50000/reserve_0_47
ln -s /dev/raw/raw843 /oradata/tpcc50000/reserve_0_48
ln -s /dev/raw/raw844 /oradata/tpcc50000/reserve_0_49
ln -s /dev/raw/raw845 /oradata/tpcc50000/reserve_0_50
ln -s /dev/raw/raw846 /oradata/tpcc50000/reserve_0_51
ln -s /dev/raw/raw847 /oradata/tpcc50000/reserve_0_52
ln -s /dev/raw/raw848 /oradata/tpcc50000/reserve_0_53
ln -s /dev/raw/raw849 /oradata/tpcc50000/reserve_0_54
ln -s /dev/raw/raw850 /oradata/tpcc50000/reserve_0_55
ln -s /dev/raw/raw851 /oradata/tpcc50000/reserve_0_56
ln -s /dev/raw/raw852 /oradata/tpcc50000/reserve_0_57
ln -s /dev/raw/raw853 /oradata/tpcc50000/reserve_0_58
ln -s /dev/raw/raw854 /oradata/tpcc50000/reserve_0_59
ln -s /dev/raw/raw855 /oradata/tpcc50000/reserve_0_60
ln -s /dev/raw/raw856 /oradata/tpcc50000/reserve_0_61
ln -s /dev/raw/raw857 /oradata/tpcc50000/reserve_0_62
ln -s /dev/raw/raw858 /oradata/tpcc50000/reserve_0_63
ln -s /dev/raw/raw859 /oradata/tpcc50000/reserve_0_64
ln -s /dev/raw/raw860 /oradata/tpcc50000/reserve_0_65
ln -s /dev/raw/raw861 /oradata/tpcc50000/reserve_0_66
ln -s /dev/raw/raw862 /oradata/tpcc50000/reserve_0_67
ln -s /dev/raw/raw863 /oradata/tpcc50000/reserve_0_68
ln -s /dev/raw/raw864 /oradata/tpcc50000/reserve_0_69
ln -s /dev/raw/raw865 /oradata/tpcc50000/reserve_0_70
ln -s /dev/raw/raw866 /oradata/tpcc50000/reserve_0_71
ln -s /dev/raw/raw867 /oradata/tpcc50000/icust1_0_0
ln -s /dev/raw/raw868 /oradata/tpcc50000/istok_0_0
ln -s /dev/raw/raw869 /oradata/tpcc50000/icust1_0_1
ln -s /dev/raw/raw870 /oradata/tpcc50000/istok_0_1
ln -s /dev/raw/raw871 /oradata/tpcc50000/icust1_0_2
ln -s /dev/raw/raw872 /oradata/tpcc50000/istok_0_2
ln -s /dev/raw/raw873 /oradata/tpcc50000/icust1_0_3
ln -s /dev/raw/raw874 /oradata/tpcc50000/istok_0_3
ln -s /dev/raw/raw875 /oradata/tpcc50000/icust1_0_4
ln -s /dev/raw/raw876 /oradata/tpcc50000/istok_0_4
ln -s /dev/raw/raw877 /oradata/tpcc50000/icust1_0_5
ln -s /dev/raw/raw878 /oradata/tpcc50000/istok_0_5
ln -s /dev/raw/raw879 /oradata/tpcc50000/icust1_0_6
ln -s /dev/raw/raw880 /oradata/tpcc50000/istok_0_6
ln -s /dev/raw/raw881 /oradata/tpcc50000/icust1_0_7
ln -s /dev/raw/raw882 /oradata/tpcc50000/istok_0_7
ln -s /dev/raw/raw883 /oradata/tpcc50000/icust1_0_8
ln -s /dev/raw/raw884 /oradata/tpcc50000/istok_0_8
ln -s /dev/raw/raw885 /oradata/tpcc50000/icust1_0_9
ln -s /dev/raw/raw886 /oradata/tpcc50000/istok_0_9
ln -s /dev/raw/raw887 /oradata/tpcc50000/icust1_0_10
ln -s /dev/raw/raw888 /oradata/tpcc50000/istok_0_10
ln -s /dev/raw/raw889 /oradata/tpcc50000/icust1_0_11
ln -s /dev/raw/raw890 /oradata/tpcc50000/istok_0_11
ln -s /dev/raw/raw891 /oradata/tpcc50000/icust1_0_12
ln -s /dev/raw/raw892 /oradata/tpcc50000/istok_0_12
ln -s /dev/raw/raw893 /oradata/tpcc50000/icust1_0_13
ln -s /dev/raw/raw894 /oradata/tpcc50000/istok_0_13
ln -s /dev/raw/raw895 /oradata/tpcc50000/icust1_0_14
ln -s /dev/raw/raw896 /oradata/tpcc50000/istok_0_14
ln -s /dev/raw/raw897 /oradata/tpcc50000/icust1_0_15
ln -s /dev/raw/raw898 /oradata/tpcc50000/istok_0_15
ln -s /dev/raw/raw899 /oradata/tpcc50000/icust1_0_16
ln -s /dev/raw/raw900 /oradata/tpcc50000/istok_0_16
ln -s /dev/raw/raw901 /oradata/tpcc50000/icust1_0_17
ln -s /dev/raw/raw902 /oradata/tpcc50000/istok_0_17
ln -s /dev/raw/raw903 /oradata/tpcc50000/icust1_0_18
ln -s /dev/raw/raw904 /oradata/tpcc50000/istok_0_18
ln -s /dev/raw/raw905 /oradata/tpcc50000/icust1_0_19
ln -s /dev/raw/raw906 /oradata/tpcc50000/istok_0_19
ln -s /dev/raw/raw907 /oradata/tpcc50000/icust1_0_20
ln -s /dev/raw/raw908 /oradata/tpcc50000/istok_0_20
ln -s /dev/raw/raw909 /oradata/tpcc50000/icust1_0_21
ln -s /dev/raw/raw910 /oradata/tpcc50000/istok_0_21
ln -s /dev/raw/raw911 /oradata/tpcc50000/icust1_0_22
ln -s /dev/raw/raw912 /oradata/tpcc50000/istok_0_22
ln -s /dev/raw/raw913 /oradata/tpcc50000/icust1_0_23
ln -s /dev/raw/raw914 /oradata/tpcc50000/istok_0_23
ln -s /dev/raw/raw915 /oradata/tpcc50000/icust1_0_24
ln -s /dev/raw/raw916 /oradata/tpcc50000/istok_0_24
ln -s /dev/raw/raw917 /oradata/tpcc50000/icust1_0_25
ln -s /dev/raw/raw918 /oradata/tpcc50000/istok_0_25
```


Copyright © 2004 NEC Corporation

```
ln -s /dev/raw/raw919 /oradata/tpcc50000/icust1_0_26
ln -s /dev/raw/raw920 /oradata/tpcc50000/istok_0_26
ln -s /dev/raw/raw921 /oradata/tpcc50000/icust1_0_27
ln -s /dev/raw/raw922 /oradata/tpcc50000/istok_0_27
ln -s /dev/raw/raw923 /oradata/tpcc50000/icust1_0_28
ln -s /dev/raw/raw924 /oradata/tpcc50000/istok_0_28
ln -s /dev/raw/raw925 /oradata/tpcc50000/icust1_0_29
ln -s /dev/raw/raw926 /oradata/tpcc50000/istok_0_29
ln -s /dev/raw/raw927 /oradata/tpcc50000/icust1_0_30
ln -s /dev/raw/raw928 /oradata/tpcc50000/istok_0_30
ln -s /dev/raw/raw929 /oradata/tpcc50000/icust1_0_31
ln -s /dev/raw/raw930 /oradata/tpcc50000/istok_0_31
ln -s /dev/raw/raw931 /oradata/tpcc50000/icust1_0_32
ln -s /dev/raw/raw932 /oradata/tpcc50000/istok_0_32
ln -s /dev/raw/raw933 /oradata/tpcc50000/icust1_0_33
ln -s /dev/raw/raw934 /oradata/tpcc50000/istok_0_33
ln -s /dev/raw/raw935 /oradata/tpcc50000/icust1_0_34
ln -s /dev/raw/raw936 /oradata/tpcc50000/istok_0_34
ln -s /dev/raw/raw937 /oradata/tpcc50000/icust1_0_35
ln -s /dev/raw/raw938 /oradata/tpcc50000/istok_0_35
ln -s /dev/raw/raw939 /oradata/tpcc50000/nodr_0_0
ln -s /dev/raw/raw940 /oradata/tpcc50000/temp_0_0
ln -s /dev/raw/raw941 /oradata/tpcc50000/empt_0_1
ln -s /dev/raw/raw942 /oradata/tpcc50000/temp_0_1
ln -s /dev/raw/raw943 /oradata/tpcc50000/nodr_0_1
ln -s /dev/raw/raw944 /oradata/tpcc50000/temp_0_2
ln -s /dev/raw/raw945 /oradata/tpcc50000/empt_0_3
ln -s /dev/raw/raw946 /oradata/tpcc50000/temp_0_3
ln -s /dev/raw/raw947 /oradata/tpcc50000/nodr_0_2
ln -s /dev/raw/raw948 /oradata/tpcc50000/temp_0_4
ln -s /dev/raw/raw949 /oradata/tpcc50000/empt_0_5
ln -s /dev/raw/raw950 /oradata/tpcc50000/temp_0_5
ln -s /dev/raw/raw951 /oradata/tpcc50000/cont_0
ln -s /dev/raw/raw952 /oradata/tpcc50000/temp_0_6
ln -s /dev/raw/raw953 /oradata/tpcc50000/empt_0_7
ln -s /dev/raw/raw954 /oradata/tpcc50000/temp_0_7
ln -s /dev/raw/raw955 /oradata/tpcc50000/cont_1
ln -s /dev/raw/raw956 /oradata/tpcc50000/temp_0_8
ln -s /dev/raw/raw957 /oradata/tpcc50000/empt_0_9
ln -s /dev/raw/raw958 /oradata/tpcc50000/temp_0_9
ln -s /dev/raw/raw959 /oradata/tpcc50000/system
ln -s /dev/raw/raw960 /oradata/tpcc50000/temp_0_10
ln -s /dev/raw/raw961 /oradata/tpcc50000/empt_0_11
ln -s /dev/raw/raw962 /oradata/tpcc50000/temp_0_11
ln -s /dev/raw/raw963 /oradata/tpcc50000/sysaux
ln -s /dev/raw/raw964 /oradata/tpcc50000/temp_0_12
ln -s /dev/raw/raw965 /oradata/tpcc50000/empt_0_13
ln -s /dev/raw/raw966 /oradata/tpcc50000/temp_0_13
ln -s /dev/raw/raw967 /oradata/tpcc50000/undo_ts
ln -s /dev/raw/raw968 /oradata/tpcc50000/temp_0_14
ln -s /dev/raw/raw969 /oradata/tpcc50000/empt_0_15
ln -s /dev/raw/raw970 /oradata/tpcc50000/temp_0_15
ln -s /dev/raw/raw971 /oradata/tpcc50000/ware_0_1
ln -s /dev/raw/raw972 /oradata/tpcc50000/temp_0_16
ln -s /dev/raw/raw973 /oradata/tpcc50000/empt_0_17
ln -s /dev/raw/raw974 /oradata/tpcc50000/temp_0_17
ln -s /dev/raw/raw975 /oradata/tpcc50000/dist_0_1
ln -s /dev/raw/raw976 /oradata/tpcc50000/temp_0_18
ln -s /dev/raw/raw977 /oradata/tpcc50000/empt_0_19
ln -s /dev/raw/raw978 /oradata/tpcc50000/temp_0_19
ln -s /dev/raw/raw979 /oradata/tpcc50000/item_0_1
ln -s /dev/raw/raw980 /oradata/tpcc50000/temp_0_20
ln -s /dev/raw/raw981 /oradata/tpcc50000/empt_0_21
ln -s /dev/raw/raw982 /oradata/tpcc50000/temp_0_21
ln -s /dev/raw/raw983 /oradata/tpcc50000/iware_0_1
ln -s /dev/raw/raw984 /oradata/tpcc50000/temp_0_22
ln -s /dev/raw/raw985 /oradata/tpcc50000/empt_0_23
ln -s /dev/raw/raw986 /oradata/tpcc50000/temp_0_23
ln -s /dev/raw/raw987 /oradata/tpcc50000/idist_0_1
ln -s /dev/raw/raw988 /oradata/tpcc50000/temp_0_24
ln -s /dev/raw/raw989 /oradata/tpcc50000/empt_0_25
ln -s /dev/raw/raw990 /oradata/tpcc50000/temp_0_25
ln -s /dev/raw/raw991 /oradata/tpcc50000/iitem_0_1
ln -s /dev/raw/raw992 /oradata/tpcc50000/temp_0_26
ln -s /dev/raw/raw993 /oradata/tpcc50000/empt_0_27
ln -s /dev/raw/raw994 /oradata/tpcc50000/temp_0_27
ln -s /dev/raw/raw995 /oradata/tpcc50000/sp_0_1
ln -s /dev/raw/raw996 /oradata/tpcc50000/temp_0_28
ln -s /dev/raw/raw997 /oradata/tpcc50000/empt_0_29
ln -s /dev/raw/raw998 /oradata/tpcc50000/temp_0_29
ln -s /dev/raw/raw999 /oradata/tpcc50000/empt_0_30
ln -s /dev/raw/raw1000 /oradata/tpcc50000/temp_0_30
ln -s /dev/raw/raw1001 /oradata/tpcc50000/empt_0_31
ln -s /dev/raw/raw1002 /oradata/tpcc50000/temp_0_31
ln -s /dev/raw/raw1003 /oradata/tpcc50000/empt_0_32
ln -s /dev/raw/raw1004 /oradata/tpcc50000/temp_0_32
ln -s /dev/raw/raw1005 /oradata/tpcc50000/empt_0_33
ln -s /dev/raw/raw1006 /oradata/tpcc50000/temp_0_33
```


Copyright © 2004 NEC Corporation

```
ln -s /dev/raw/raw1007 /oradata/tpcc50000/empt_0_34
ln -s /dev/raw/raw1008 /oradata/tpcc50000/temp_0_34
ln -s /dev/raw/raw1009 /oradata/tpcc50000/empt_0_35
ln -s /dev/raw/raw1010 /oradata/tpcc50000/temp_0_35
```

createtable_cust.sql

```
/* created automatically by /home/oracle/031127/tpcc-
kit50000/scripts/buildcreatetable.sh Fri Nov 28 15:32:06 JST 2003 */
set timing on
    set sqlblanklines on
    spool createtable_cust.log
    set echo on
      drop cluster custcluster including tables ;

create cluster custcluster (
  c_id number
, c_d_id number
, c_w_id number
  )
  single table
  hashkeys 1500000000
  hash is ( (c_id * ( 50000 * 10 ) + c_w_id * 10 + c_d_id) )
  size 205
  pctfree 0  initrans 3
  storage ( buffer_pool recycle )
  parallel (degree 32)
  tablespace cust_0;

create table cust (
  c_id number
, c_d_id number
, c_w_id number
, c_discount number
, c_credit char(2)
, c_last varchar2(16)
, c_first varchar2(16)
, c_credit_lim number
, c_balance binary_double
, c_ytd_payment binary_float
, c_payment_cnt binary_float
, c_delivery_cnt binary_float
, 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 char(500)
)
cluster custcluster (
  c_id
, c_d_id
, c_w_id
);
    set echo off
    spool off
    exit sql.sqlcode;
```

createtable_hist.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreatetable.sh Fri Nov 28 18:29:32 JST 2003 */
set timing on
    set sqlblanklines on
    spool createtable_hist.log
    set echo on
      drop table hist ;

create table hist (
  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 binary_float
, h_data varchar2(24)
)
  pctfree 5  initrans 4
  storage ( buffer_pool recycle )
  tablespace hist_0 ;
    set echo off
    spool off
    exit sql.sqlcode;
```

createtable_nord.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreatetable.sh Fri Nov 28 18:29:37 JST 2003 */
set timing on
    set sqlblanklines on
    spool createtable_nord.log
    set echo on
        drop cluster nordcluster_queue including tables ;

  create cluster nordcluster_queue (
    no_w_id number
, no_d_id number
, no_o_id number SORT
    )

    hashkeys 500000
    hash is ( (no_w_id - 1) * 10 + no_d_id - 1 )
    size 190
    tablespace nord_0;

  create table nord (
    no_w_id number
```


Copyright © 2004 NEC Corporation

```
, no_d_id number
, no_o_id number sort
    , constraint nord_uk primary key ( no_w_id
, no_d_id
, no_o_id )
  )
  cluster nordcluster_queue (
    no_w_id
, no_d_id
, no_o_id
  );
    set echo off
    spool off
    exit sql.sqlcode;
```

createtable_nord.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreatetable.sh Fri Nov 28 18:29:37 JST 2003 */
set timing on
    set sqlblanklines on
    spool createtable_nord.log
    set echo on
        drop cluster nordcluster_queue including tables ;

  create cluster nordcluster_queue (
    no_w_id number
, no_d_id number
, no_o_id number SORT
    )

    hashkeys 500000
    hash is ( (no_w_id - 1) * 10 + no_d_id - 1 )
    size 190
    tablespace nord_0;

  create table nord (
    no_w_id number
, no_d_id number
, no_o_id number sort
    , constraint nord_uk primary key ( no_w_id
, no_d_id
, no_o_id )
  )
  cluster nordcluster_queue (
    no_w_id
, no_d_id
, no_o_id
  );
    set echo off
    spool off
    exit sql.sqlcode;
```

createtable_ordr.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreatetable.sh Fri Nov 28 18:29:35 JST 2003 */
set timing on
    set sqlblanklines on
    spool createtable_ordr.log
    set echo on
        drop cluster ordrcluster_queue including tables ;

  create cluster ordrcluster_queue (
    o_w_id number
, o_d_id number
, o_id number SORT
, o_number number SORT
    )

    hashkeys 500000
    hash is ( (o_w_id - 1) * 10 + o_d_id - 1 )
    size 1490
    tablespace ordr_0;

  create table ordr (
    o_id number sort
, 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
    , constraint ordr_uk primary key ( o_w_id
, o_d_id
, o_id )
  )
  cluster ordrcluster_queue (
    o_w_id
, o_d_id
, o_id
  );
    set echo off
    spool off
    exit sql.sqlcode;
```

createtable_ordl.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreatetable.sh Fri Nov 28 18:29:36 JST 2003 */
set timing on
    set sqlblanklines on
    spool createtable_ordl.log
    set echo on
        create table ordl (
```


Copyright © 2004 NEC Corporation

```
    ol_w_id number
, ol_d_id number
, ol_o_id number sort
, ol_number number sort
, ol_i_id number
, ol_delivery_d date
, ol_amount binary_float
, ol_supply_w_id number
, ol_quantity binary_float
, ol_dist_info char(24)
  , constraint ordl_uk primary key (ol_w_id, ol_d_id, ol_o_id,
ol_number  )) CLUSTER ordrcluster_queue(ol_w_id, ol_d_id, ol_o_id,
ol_number) ;
    set echo off
    spool off
    exit sql.sqlcode;
```

createtable_stok.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreatetable.sh Fri Nov 28 18:29:33 JST 2003 */
set timing on
    set sqlblanklines on
    spool createtable_stok.log
    set echo on
      drop cluster stokcluster including tables ;

create cluster stokcluster (
  s_i_id number
, s_w_id number
  )
  single table
  hashkeys 5000000000
  hash is ( (s_i_id * 50000 + s_w_id) )
  size 350
  pctfree 0  initrans 3
  storage ( buffer_pool keep )
  tablespace stok_0;

create table stok (
  s_i_id number
, s_w_id number
, s_quantity binary_float
, s_ytd binary_double
, s_order_cnt binary_float
, s_remote_cnt binary_float
, 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 stokcluster (
  s_i_id
, s_w_id
);
    set echo off
    spool off
    exit sql.sqlcode;
```

createtable_dist.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreatetable.sh Fri Nov 28 18:29:31 JST 2003 */
set timing on
    set sqlblanklines on
    spool createtable_dist.log
    set echo on
      drop cluster distcluster including tables ;

create cluster distcluster (
  d_id number
, d_w_id number
  )
  single table
  hashkeys 500000
  hash is ( ((d_w_id * 10) + d_id) )
  size 1448
    initrans 4
  storage ( buffer_pool default )
  tablespace dist_0;

create table dist (
  d_id number
, d_w_id number
, d_ytd binary_double
, d_next_o_id number
, d_tax 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 distcluster (
  d_id
, d_w_id
);
    set echo off
    spool off
```


Copyright © 2004 NEC Corporation

```
    exit sql.sqlcode;
```

createtable_item.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreatetable.sh Fri Nov 28 18:29:34 JST 2003 */
set timing on
    set sqlblanklines on
    spool createtable_item.log
    set echo on
      drop cluster itemcluster including tables ;

create cluster itemcluster (
  i_id number(6,0)
  )
  single table
  hashkeys 100000
  hash is ( (i_id) )
  size 120
  pctfree 0  initrans 3
  storage ( buffer_pool keep )
  tablespace item_0;

create table item (
  i_id number(6,0)
, i_name varchar2(24)
, i_price binary_float
, i_data varchar2(50)
, i_im_id number
)
cluster itemcluster (
  i_id
);
    set echo off
    spool off
    exit sql.sqlcode;
```

createtable_ware.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreatetable.sh Fri Nov 28 18:29:28 JST 2003 */
set timing on
    set sqlblanklines on
    spool createtable_ware.log
    set echo on
      drop cluster warecluster including tables ;

create cluster warecluster (
  w_id number
  )
  single table
  hashkeys 50000
  hash is ( (w_id - 1) )
  size 1448
    initrans 2
  storage ( buffer_pool default )
  tablespace ware_0;

create table ware (
  w_id number
, w_ytd binary_double
, w_tax number
, 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 warecluster (
  w_id
);
    set echo off
    spool off
    exit sql.sqlcode;
```

createindex_iware.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreateindex.sh Fri Nov 28 18:29:40 JST 2003 */
set timing on
    set sqlblanklines on
    spool createindex_iware.log ;
    set echo on ;
    drop index iware ;
      create unique index iware on ware ( w_id )
  pctfree 1  initrans 3
  storage ( buffer_pool default )
  parallel 64
  compute statistics
  tablespace iware_0 ;
    set echo off
    spool off
    exit sql.sqlcode;
```

createindex_iitem.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreateindex.sh Fri Nov 28 18:29:43 JST 2003 */
set timing on
    set sqlblanklines on
    spool createindex_iitem.log ;
```


Copyright © 2004 NEC Corporation

```
    set echo on ;
    drop index iitem ;
      create unique index iitem on item ( i_id )
  pctfree 5  initrans 4
  storage ( buffer_pool default )
  parallel 64
  compute statistics
  tablespace iitem_0 ;
    set echo off
    spool off
    exit sql.sqlcode;
```

createindex_idist.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreateindex.sh Fri Nov 28 18:29:42 JST 2003 */
set timing on
    set sqlblanklines on
    spool createindex_idist.log ;
    set echo on ;
    drop index idist ;
      create unique index idist on dist ( d_w_id
, d_id )
  pctfree 5  initrans 3
  storage ( buffer_pool default )
  parallel 64
  compute statistics
  tablespace idist_0 ;
    set echo off
    spool off
    exit sql.sqlcode;
```

createindex_iordl.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreateindex.sh Fri Nov 28 18:29:45 JST 2003 */
set timing on
  exit 0;
```

createindex_icust1.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreateindex.sh Fri Nov 28 18:29:41 JST 2003 */
set timing on
    set sqlblanklines on
    spool createindex_icust1.log ;
    set echo on ;
    drop index icust1 ;
      create unique index icust1 on cust ( c_w_id
, c_d_id
, c_id )
  pctfree 1  initrans 3
  storage ( buffer_pool default )
  parallel 64
  compute statistics
  tablespace icust1_0 ;
    set echo off
    spool off
    exit sql.sqlcode;
```

createindex_icust2.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreateindex.sh Fri Nov 28 18:29:41 JST 2003 */
set timing on
    set sqlblanklines on
    spool createindex_icust2.log ;
    set echo on ;
    drop index icust2 ;
      create unique index icust2 on cust ( c_last
, c_w_id
, c_d_id
, c_first
, c_id )
  pctfree 1  initrans 3
  storage ( buffer_pool default )
  parallel 64
  compute statistics
  tablespace icust2_0 ;
    set echo off
    spool off
    exit sql.sqlcode;
```

createindex_inord.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreateindex.sh Fri Nov 28 18:29:45 JST 2003 */
set timing on
  exit 0;
```

createindex_iordr1.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreateindex.sh Fri Nov 28 18:29:44 JST 2003 */
set timing on
  exit 0;
```

createindex_iordr2.sql

Copyright © 2004 NEC Corporation

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreateindex.sh Fri Nov 28 18:29:44 JST 2003 */
set timing on
    set sqlblanklines on
    spool createindex_iordr2.log ;
    set echo on ;
    drop index iordr2 ;
      create unique index iordr2 on ordr ( o_c_id
, o_d_id
, o_w_id
, o_id )
  pctfree 25  initrans 4
  storage ( buffer_pool default )
  parallel 64
  compute statistics
  tablespace iordr2_0 ;
    set echo off
    spool off
    exit sql.sqlcode;
```

createindex_istok.sql

```
/* created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreateindex.sh Fri Nov 28 18:29:43
JST 2003 */
set timing on
    set sqlblanklines on
    spool createindex_istok.log ;
    set echo on ;
    drop index istok ;
      create unique index istok on stok ( s_i_id
, s_w_id )
  pctfree 1  initrans 3
  storage ( buffer_pool default )
  parallel 64
  compute statistics
  tablespace istok_0 ;
    set echo off
    spool off
    exit sql.sqlcode;
```

dml.sql

```
REM=====================================================================+
REM        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
REM                   OPEN SYSTEMS PERFORMANCE GROUP                  |
REM                         All Rights Reserved                       |
REM=====================================================================+
REM FILENAME
REM     dml.sql
REM DESCRIPTION
REM     Disable table locks for TPC-C tables.
REM USAGE
REM     sqlplus tpcc/tpcc dml.sql
REM=====================================================================

connect tpcc/tpcc;
set echo on;

   alter table ware disable table lock;
   alter table dist disable table lock;
   alter table cust disable table lock;
   alter table hist disable table lock;
   alter table item disable table lock;
   alter table stok disable table lock;
   alter table ordr disable table lock;
   alter table nord disable table lock;
   alter table ordl disable table lock;

set echo off;

connect $oracle_dba/$oracle_dba_password;
```

driver.sh

```
#!/bin/sh

. ./stepenv.sh

if expr $# \< 1 > /dev/null; then
  echo "$0 <starting stepname> <optional: only>"
  echo OR use:
  echo "$0 buildcreate  - to build the database creation scripts"
  echo "$0 create       - to create the database (after buildcreate)"
  echo "$0 steps        - to list individual steps"
  exit 1
fi

if expr x$1 = xsteps > /dev/null; then
  echo stepnames are from creation scripts: $tpcc_create_steps
  echo
  echo or running steps: $tpcc_steps
  echo "use the 'only' option to only do that step (otherwise all steps
after will also be executed.)"
  echo "   (e.g.  $0 listfiles only)"
  echo "use the 'through' option to do a sequence of steps
(inclusively.)"
  echo "   (e.g.  $0 shutdowndb through startupdb-p_build)"
  exit 1
fi
```


Copyright © 2004 NEC Corporation

```
startstep=$1
controlcmd=$2
endstep=$3

# Aliases for special steps
if test $startstep = buildcreate; then
  startstep=`echo $tpcc_create_steps | cut -d' ' -f1`
fi

if test $startstep = create; then
  startstep=`echo $tpcc_steps | cut -d' ' -f1`
fi

if test "x$controlcmd" = x; then
  endstep=
  # Since endstep is null it won't match any other steps, so we keep
going.
elif test "x$controlcmd" = xonly; then
  controlcmd=only
  # this is allowed
elif test "x$controlcmd" = xthrough; then
  actualstep=f
  for step in $tpcc_create_steps $tpcc_steps ; do
    if test "x$step" = "x$endstep"; then
      actualstep=t
    fi
  done
  if test $actualstep = f; then
    echo "Invalid step $endstep.  Use $0 steps  to show steps."
    exit 1
  fi
else
  echo "Invalid syntax.  Use $0 by itself for help."
  exit 1
fi

echo Starting from step: $startstep

dostep=f
for step in $tpcc_create_steps $tpcc_steps ; do
  if expr $step = $startstep > /dev/null; then
    dostep=t
  fi

  if expr $dostep = t > /dev/null; then
    echo $step
    cd $tpcc_bench
    $tpcc_scripts/`echo $step | cut -d- -f1`.sh `echo $step | sed -e's/-
*$/-/' | cut -d- -f2- | sed -e's/-/ /g'`
    lasterror=$?
    cd $tpcc_bench
    if test -n "`find $tpcc_bench/scripts -name '*.log'`"; then
       mv -f *.log `find $tpcc_bench/scripts -name '*.log'`
$tpcc_bench/log/
    else
       mv -f *.log $tpcc_bench/log/
    fi

    if expr $lasterror != 0 > /dev/null; then
      if expr $lasterror != 99 > /dev/null; then
        echo Step $step failed.  Stopping driver.
        exit 1
      else
        echo Step $step has completed and requested stop.  Stopping
driver.
        exit 0
      fi
    fi
    if test "x$controlcmd" = xonly; then
      exit 0
    fi
    if test "x$endstep" = "x$step"; then
      echo The driver reached the last desired step.  Stopping driver.
      exit 0
    fi
  fi
done

if expr $dostep = f > /dev/null; then
  echo No such step: $1
fi
```

stepenv.sh

```
# forces any env variables we set to be exported
set -a
tpcc_kit=t
tpcc_bench=$PWD
tpcc_scripts=$tpcc_bench/scripts
tpcc_require=$tpcc_scripts/require_vars.sh
tpcc_lcm=$tpcc_scripts/lcm.sh
tpcc_tokilobytes=$tpcc_scripts/tokilobytes.sh
tpcc_fromkilobytes=$tpcc_scripts/fromkilobytes.sh
tpcc_estsize=$tpcc_scripts/estsize.sh
tpcc_notneg=$tpcc_scripts/notneg.sh
tpcc_isneg=$tpcc_scripts/isneg.sh


# need a better way to check for bc, may
# resort to checking each directory in path
# if this doesn't work
#11/7/02 - alex.ni this is causing too many problems
#because systems have bc in some odd place.  typically
#mangled cygwin installs w/ mksnt/cygwin mixes
#if test -x /usr/bin/bc -o -x /bin/bc; then
tpcc_bcexpr=$tpcc_scripts/bcexpr.sh
#else
#tpcc_bcexpr=expr
```


Copyright © 2004 NEC Corporation

```
#fi


# the ksh version is a bit faster, so we want
# to use it if we have ksh.  Otherwise we have
# a compatible version.
#if test -x /bin/ksh; then
#tpcc_createts=$tpcc_scripts/createts.ksh
#else
tpcc_createts=$tpcc_scripts/createts.sh
#fi

tpcc_tabledata=$tpcc_scripts/tabledata.sh
tpcc_load=$tpcc_bench/benchrun/bin/tpccload.exe
tpcc_createtablespaces=$tpcc_scripts/createtablespaces.sh

##
tpcc_sqlplus=cat
tpcc_sqlplus_args='/nolog'
tpcc_internal_connect='connect / as sysdba'
tpcc_user_pass='tpcc/tpcc'
tpcc_dba_user_pass='system/manager'
oracle_dba=system
oracle_dba_password=manager
tpcc_sqlplus_args=
tpcc_user_pass=
tpcc_sqlplus=sqlplus
tpcc_user_pass='tpcc/tpcc'

# import options generated by gui
. ${tpcc_bench}/options.sh

#8gb oracle filesize limit (in k)
tpcc_fsize_limit_k=8388608
#2gb - 1k oracle extent limit (in k)
tpcc_extent_limit_k=2097151

# Runlen calculations should be in hours, but
# this was the old calculation, which assumed
# minutes, and also 8 times:
# tpcc_runlen=`$tpcc_bcexpr 8 \* 60 \* $tpcc_runlen`
# we just want to keep the value as it is.
tpcc_kilo_bytes=1024
tpcc_system_size=200M
#tpcc_logfile_size=`$tpcc_bcexpr 20 + \( $tpcc_scale \)`

if test $tpcc_np -gt 1 ; then
  # 4.69k per commit * 2.1 commit per TPMC ~ 9.85K
  # 9.85k * 30 minutes * 12.5 TPMC per Warehouse = 3693
  tpcc_logfile_size=`$tpcc_bcexpr \( $tpcc_scale \* 3693 \) /
$tpcc_kilo_bytes`
else
  # 2.4k per commit * 2.1 commit per TPMC ~ 5k
  # 5k * 30 minutes * 12.5 TPMC per Warehouse = 1875
  tpcc_logfile_size=`$tpcc_bcexpr \( $tpcc_scale \* 1875 \) /
$tpcc_kilo_bytes`
fi

if test $tpcc_logfile_size -lt 1024; then
  tpcc_logfile_size=1024
fi
tpcc_logfile_size="${tpcc_logfile_size}M"

tpcc_undo_size=`$tpcc_bcexpr 2 \* $tpcc_scale`
if test $tpcc_undo_size -gt 8096; then
  tpcc_undo_size=8096
fi
if test $tpcc_undo_size -lt 512; then
  tpcc_undo_size=512
fi
tpcc_undo_size="${tpcc_undo_size}M"

tpcc_undo_bs=8K

tpcc_statspack_size=`$tpcc_bcexpr 1 \* $tpcc_scale`
if test $tpcc_statspack_size -gt 2048; then
  tpcc_statspack_size=2048
fi
tpcc_statspack_size="${tpcc_statspack_size}M"

tpcc_sysaux_size=120M

# fixed table params

#table list (note temp is always at the end since it may use numbers
from other tables, and it's not included in these lists)
tpcc_table_list='ware cust dist hist stok item ordr ordl nord'
tpcc_index_list='iware icust1 icust2 idist istok iitem iordr1 iordr2
iordl inord'
#for these I use average row length, calculated from multi-blocksize
stats.
#we figure out how many new rows we will gain in a run (in
createtablesspaces.sh)
#and add that much to the base tablespace size.
tpcc_hist_growth=51
tpcc_ordr_growth=35
tpcc_nord_growth=regular
#tpcc_ordl_growth=660
tpcc_ordl_growth=900

#i started indices at 1/10th... need an exact figure
tpcc_iordr1_growth=20
tpcc_iordr2_growth=20
tpcc_iordl_growth=66
tpcc_inord_growth=2

tpcc_item_growth=0
tpcc_iitem_growth=0
tpcc_temp_growth=0

tpcc_cust_growth=regular
tpcc_icust1_growth=regular
```


Copyright © 2004 NEC Corporation

```
tpcc_icust2_growth=regular

tpcc_stok_growth=regular
tpcc_istok_growth=regular

tpcc_ware_growth=regular
tpcc_iware_growth=regular

tpcc_dist_growth=regular
tpcc_idist_growth=regular

# minimum size of temp tablespace
tpcc_tempts_min=10240

# for Linux, set appropriate tablespace heuristics
# to set high io tables to have 64 files, and minimize
# others.
if expr $tpcc_os = linux > /dev/null; then
#  for table in $tpcc_table_list $tpcc_index_list temp; do
#    eval "tpcc_${table}_tsfileinc=1"
#  done
  if test $tpcc_numfiles = 0 ; then
    tpcc_numfiles=256
  fi
  tpcc_os=unix

#  tpcc_stok_tsfileinc=64
#  tpcc_cust_tsfileinc=64
#  tpcc_iordl2_tsfileinc=16
#  tpcc_icust2_tsfileinc=16
#  tpcc_iordl_tsfileinc=16
else
#in case someone changes out of linux, and the shell is stuck
  for table in $tpcc_table_list $tpcc_index_list temp; do
    eval "tpcc_${table}_tsfileinc="
  done
fi
  tpcc_stok_tsfileinc=
  tpcc_cust_tsfileinc=
  tpcc_iordl2_tsfileinc=
  tpcc_icust2_tsfileinc=
  tpcc_iordl_tsfileinc=
#fi


# import local options
. ${tpcc_bench}/localoptions.sh

if expr `echo x$tpcc_no_options` = xt > /dev/null; then
  echo Please modify ${tpcc_bench}/localoptions.sh to configure the
generator.
  exit 1
fi

tpcc_fixordrordl=${tpcc_genscripts_dir}/loadfixordrordl.sh
tpcc_updateordrordl=${tpcc_scripts}/updateordrordl.sh

#tp- get table param.  (that is, $tpcc_tablename_tableparam)
tp(){
  eval echo \""\$tpcc_$1_$2"\"
}

# automatically generated variables
if expr `echo $tpcc_version | cut -b1` = t > /dev/null; then
  tpcc_auto_undo=t
else
  tpcc_auto_undo=f
fi
if expr `echo $tpcc_version | cut -b2` = t > /dev/null; then
  tpcc_autospace_avail=t
else
  tpcc_autospace_avail=f
fi
if expr `echo $tpcc_version | cut -b3` = t > /dev/null; then
  tpcc_queue_avail=t
  tpcc_use_sysaux=t
else
  tpcc_queue_avail=f
  tpcc_use_sysaux=f
fi

# for NT, ORACLE does not like $variables in sql scripts, so we must
# hardcode these things for it.
if test x$tpcc_os = xnt; then
  tpcc_hardcode=t
else
  tpcc_hardcode=f
fi

# if this is unset we need to make sure it's something anyway
if test x$tpcc_defbs = x; then
  tpcc_defbs=2
fi

# used for loading program
if test x$tpcc_hash_overflow = xt; then
  tpcc_hash_overflow=t
else
  unset tpcc_hash_overflow
fi
if test x$tpcc_overflow = xt; then
  tpcc_hash_overflow=t
else
  unset tpcc_hash_overflow
fi

tpcc_create_steps="buildtpccflags buildcreatets buildcreatedb \
buildcreatetable-ware buildcreatetable-cust buildcreatetable-dist
buildcreatetable-hist buildcreatetable-stok buildcreatetable-item
buildcreatetable-ordr buildcreatetable-ordl buildcreatetable-nord \
buildloadware buildloaddist buildloaditem buildloadhist buildloadnord
buildloadordrordl buildloadcust buildloadstok \
```


Copyright © 2004 NEC Corporation

```
buildcreateindex-iware buildcreateindex-icust1 buildcreateindex-icust2
buildcreateindex-idist buildcreateindex-istok buildcreateindex-iitem
buildcreateindex-iordr1 buildcreateindex-iordr2 buildcreateindex-iordl
buildcreateindex-inord \
buildstoreprocsql buildspacestats listfiles
"

# remove runscript-loadfixordrordl - shuang, 030626

tpcc_steps="runsqllocal-createdb shutdowndb startupdb-p_build createuser
ddview runscript-createts assigntemp \
  runsql-createtable_ware runsql-createtable_cust runsql-
createtable_dist runsql-createtable_hist runsql-createtable_stok runsql-
createtable_item runsql-createtable_ordr runsql-createtable_ordl runsql-
createtable_nord \
runscript-loadware runscript-loaddist runscript-loaditem runscript-
loadhist runscript-loadnord runscript-loadordrordl runscript-loadcust
runscript-loadstok \
analyze runsql-createindex_iware runsql-createindex_icust1 runsql-
createindex_icust2 runsql-createindex_idist runsql-createindex_istok
runsql-createindex_iitem runsql-createindex_iordr1 runsql-
createindex_iordr2 runsql-createindex_iordl runsql-createindex_inord \
createstats createstoredprocs createspacestats createmisc"

tpcc_total_files=524

# no longer automatically exports env variables
set +a

# check for problems with configuration
badconf=
for table in $tpcc_table_list; do
  if expr `tp $table imp` = queue > /dev/null; then
    if expr $tpcc_queue_avail = f > /dev/null; then
        echo Table $table may not be a queue, since queues are
        echo are unavailable in the selected Oracle version.
        badconf=t
    fi
  fi
  if expr $tpcc_autospace_avail = f \& `tp $table autospace` = t >
/dev/null; then
    echo Table $table may not use bitmapped space management
    echo since it is not available in the selected Oracle version.
    badconf=t
  fi
done

if test -n "$badconf"; then
  exit 1
fi

# make sure we have everything
if $tpcc_require ORACLE_SID \
 tpcc_tokilobytes tpcc_createts tpcc_lcm\
 tpcc_sqlplus tpcc_internal_connect\
 tpcc_np tpcc_cpu tpcc_os tpcc_runlen tpcc_ldrive tpcc_scale
tpcc_disks_location tpcc_auto_undo tpcc_tempts_min\
 tpcc_system_size tpcc_logfile_size\
 tpcc_undo_size tpcc_undo_bs\
 oracle_dba oracle_dba_password tpcc_dba_user_pass
then exit 1; fi

if test x$tpcc_hardcode != xt; then
  tpcc_disks_location=${tpcc_disks_location}/
#  tpcc_sql_dir='$tpcc_sql_dir'
#  tpcc_statspack_size='$tpcc_statspack_size'
#  tpcc_genscripts_dir='$tpcc_genscripts_dir'
fi
```

createdb.sql

```
/* created automatically by  /home/oracle/tpcc-
kit50000/scripts/buildcreatedb.sh Fri Nov 28 18:29:27 JST 2003 */
spool createdb.log

set echo on

shutdown abort

startup pfile=p_create.ora nomount
create database tpcc
  controlfile reuse
  maxinstances 1
  datafile
   '/oradata/tpcc50000/system_1' size 200M reuse
  logfile '/oradata/tpcc50000/log_1_1' size 120000M reuse,
          '/oradata/tpcc50000/log_1_2' size 120000M reuse
  sysaux datafile '/oradata/tpcc50000/aux.df' size 120M reuse ;




create undo tablespace undo_1 datafile
  '/oradata/tpcc50000/roll1' size 8096M reuse blocksize 8K;

set echo off
exit sql.sqlcode
```

addfile.sh

```
#!/bin/sh
# $1 = tablespace name
# $2 = filename
# $3 = size
# $4 = temporary ts (1) or not (0)
# global variable $tpcc_listfiles, does not execute sql
```


Copyright © 2004 NEC Corporation

```
if expr x$tpcc_listfiles = xt > /dev/null; then
  echo $2 $3 >> $tpcc_bench/files.dat
  exit 0
fi

if expr $4 = 1 > /dev/null; then
  altersql="alter tablespace $1 add tempfile '$2' size $3 reuse;"
else
  altersql="alter tablespace $1 add datafile '$2' size $3 reuse
autoextend on;"
fi

$tpcc_sqlplus $tpcc_user_pass <<!
   spool addfile_$1.log
   set echo on
  $altersql
   set echo off
   spool off
   exit ;
!
```

addts.sh

```
#!/bin/sh
# $1 = tablespace name
# $2 = filename
# $3 = size
# $4 = uniform size
# $5 = block size
# $6 = temporary ts (1) or not (0)
# $7 = bitmapped manage (t) or not (f) or (d) for dictionary
# global variable $tpcc_listfiles, does not execute sql

if expr x$tpcc_listfiles = xt > /dev/null; then
  echo $2 $3 >> $tpcc_bench/files.dat
  exit 0
fi

if expr $5 = auto > /dev/null; then
  bssql=
else
  bssql="blocksize $5"
fi

if expr $6 = 1 > /dev/null; then
  createsql="create temporary tablespace $1 tempfile '$2' size $3 reuse
extent management local uniform size $4;"
else
  if expr x$7 = xt > /dev/null; then
    createsql="create tablespace $1 datafile '$2' size $3 reuse extent
management local uniform size $4 segment space management auto $bssql
nologging ;"
  else
    if expr x$7 = xd > /dev/null; then
      createsql="create tablespace $1 datafile '$2' size $3 reuse extent
management dictionary nologging $bssql;"
    else
      createsql="create tablespace $1 datafile '$2' size $3 reuse extent
management local uniform size $4 segment space management manual $bssql
nologging ;"
    fi
  fi
fi

$tpcc_sqlplus $tpcc_user_pass <<!
   spool createts_$1.log
   set echo on
   drop tablespace $1 including contents;
 $createsql
   set echo off
   spool off
   exit ;
!
```

options.sh

```
tpcc_os='linux'
tpcc_version='ttt'
tpcc_ldrive='72'
tpcc_scale='50000'
tpcc_np='1'
tpcc_cpu='32'
tpcc_memsize='250000'
tpcc_runlen='25'
tpcc_compress='f'
tpcc_overflow='t'
tpcc_defbs='2'
tpcc_ieee_number='t'
tpcc_numfiles='-1'

tpcc_cust_imp='cluster'
tpcc_cust_size='calc'
tpcc_cust_ext='calc'
tpcc_cust_nf='calc'
tpcc_cust_bs='auto'
tpcc_cust_used='-1'
tpcc_cust_free='0'
tpcc_cust_trans='3'
tpcc_cust_autospace='t'
tpcc_cust_flg='43'
tpcc_cust_fl='22'
tpcc_cust_rsize='auto'
tpcc_cust_hkey='auto'
tpcc_cust_hash='auto'
tpcc_cust_bpool='recycle'
tpcc_cust_indices=3-2-1-

tpcc_dist_imp='cluster'
```


Copyright © 2004 NEC Corporation

```
tpcc_dist_size='calc'
tpcc_dist_ext='calc'
tpcc_dist_nf='calc'
tpcc_dist_bs='auto'
tpcc_dist_used='-1'
tpcc_dist_free='-1'
tpcc_dist_trans='4'
tpcc_dist_autospace='t'
tpcc_dist_flg='43'
tpcc_dist_fl='22'
tpcc_dist_rsize='auto'
tpcc_dist_hkey='auto'
tpcc_dist_hash='auto'
tpcc_dist_bpool='default'
tpcc_dist_indices=2-1-

tpcc_hist_imp='table'
tpcc_hist_size='1791'
tpcc_hist_ext='calc'
tpcc_hist_nf='calc'
tpcc_hist_bs='4k'
tpcc_hist_used='-1'
tpcc_hist_free='5'
tpcc_hist_trans='4'
tpcc_hist_autospace='t'
tpcc_hist_flg='43'
tpcc_hist_fl='22'
tpcc_hist_rsize='auto'
tpcc_hist_hkey='auto'
tpcc_hist_hash='auto'
tpcc_hist_bpool='recycle'
tpcc_hist_indices=no

tpcc_item_imp='cluster'
tpcc_item_size='calc'
tpcc_item_ext='calc'
tpcc_item_nf='calc'
tpcc_item_bs='auto'
tpcc_item_used='-1'
tpcc_item_free='0'
tpcc_item_trans='3'
tpcc_item_autospace='t'
tpcc_item_flg='43'
tpcc_item_fl='22'
tpcc_item_rsize='auto'
tpcc_item_hkey='auto'
tpcc_item_hash='auto'
tpcc_item_bpool='keep'
tpcc_item_indices=1-

tpcc_nord_imp='queue'
tpcc_nord_size='178'
tpcc_nord_ext='calc'
tpcc_nord_nf='calc'
tpcc_nord_bs='auto'
tpcc_nord_used='-1'
tpcc_nord_free='5'
tpcc_nord_trans='4'
tpcc_nord_autospace='t'
tpcc_nord_flg='43'
tpcc_nord_fl='22'
tpcc_nord_rsize='auto'
tpcc_nord_hkey='auto'
tpcc_nord_hash='auto'
tpcc_nord_bpool='default'
tpcc_nord_indices=1-2-3-

tpcc_ordl_imp='queue'
tpcc_ordl_size='21775'
tpcc_ordl_ext='calc'
tpcc_ordl_nf='calc'
tpcc_ordl_bs='16K'
tpcc_ordl_used='-1'
tpcc_ordl_free='5'
tpcc_ordl_trans='4'
tpcc_ordl_autospace='t'
tpcc_ordl_flg='43'
tpcc_ordl_fl='22'
tpcc_ordl_rsize='auto'
tpcc_ordl_hkey='auto'
tpcc_ordl_hash='auto'
tpcc_ordl_bpool='default'
tpcc_ordl_indices=1-2-3-4-

tpcc_ordr_imp='queue'
tpcc_ordr_size='1206'
tpcc_ordr_ext='calc'
tpcc_ordr_nf='calc'
tpcc_ordr_bs='16K'
tpcc_ordr_used='-1'
tpcc_ordr_free='5'
tpcc_ordr_trans='4'
tpcc_ordr_autospace='t'
tpcc_ordr_flg='43'
tpcc_ordr_fl='22'
tpcc_ordr_rsize='auto'
tpcc_ordr_hkey='auto'
tpcc_ordr_hash='auto'
tpcc_ordr_bpool='default'
tpcc_ordr_indices=2-3-1-

tpcc_stok_imp='cluster'
tpcc_stok_size='calc'
tpcc_stok_ext='calc'
tpcc_stok_nf='calc'
tpcc_stok_bs='auto'
tpcc_stok_used='-1'
tpcc_stok_free='0'
tpcc_stok_trans='3'
tpcc_stok_autospace='t'
tpcc_stok_flg='43'
tpcc_stok_fl='22'
```


Copyright © 2004 NEC Corporation

```
tpcc_stok_rsize='auto'
tpcc_stok_hkey='auto'
tpcc_stok_hash='auto'
tpcc_stok_bpool='keep'
tpcc_stok_indices=1-2-

tpcc_ware_imp='cluster'
tpcc_ware_size='calc'
tpcc_ware_ext='calc'
tpcc_ware_nf='calc'
tpcc_ware_bs='auto'
tpcc_ware_used='-1'
tpcc_ware_free='-1'
tpcc_ware_trans='2'
tpcc_ware_autospace='t'
tpcc_ware_flg='43'
tpcc_ware_fl='22'
tpcc_ware_rsize='auto'
tpcc_ware_hkey='auto'
tpcc_ware_hash='auto'
tpcc_ware_bpool='default'
tpcc_ware_indices=1-

tpcc_icust1_imp='index'
tpcc_icust1_size='736'
tpcc_icust1_ext='calc'
tpcc_icust1_nf='calc'
tpcc_icust1_bs='auto'
tpcc_icust1_used='-1'
tpcc_icust1_free='1'
tpcc_icust1_trans='3'
tpcc_icust1_autospace='t'
tpcc_icust1_flg='43'
tpcc_icust1_fl='22'
tpcc_icust1_rsize='auto'
tpcc_icust1_hkey='auto'
tpcc_icust1_hash='auto'
tpcc_icust1_bpool='default'
tpcc_icust1_indices=3-2-1-

tpcc_icust2_imp='index'
tpcc_icust2_size='4591'
tpcc_icust2_ext='calc'
tpcc_icust2_nf='calc'
tpcc_icust2_bs='auto'
tpcc_icust2_used='-1'
tpcc_icust2_free='1'
tpcc_icust2_trans='3'
tpcc_icust2_autospace='t'
tpcc_icust2_flg='43'
tpcc_icust2_fl='22'
tpcc_icust2_rsize='auto'
tpcc_icust2_hkey='auto'
tpcc_icust2_hash='auto'
tpcc_icust2_bpool='default'
tpcc_icust2_indices=6-3-2-7-1-

tpcc_idist_imp='index'
tpcc_idist_size='4'
tpcc_idist_ext='calc'
tpcc_idist_nf='calc'
tpcc_idist_bs='auto'
tpcc_idist_used='-1'
tpcc_idist_free='5'
tpcc_idist_trans='3'
tpcc_idist_autospace='t'
tpcc_idist_flg='43'
tpcc_idist_fl='22'
tpcc_idist_rsize='auto'
tpcc_idist_hkey='auto'
tpcc_idist_hash='auto'
tpcc_idist_bpool='default'
tpcc_idist_indices=2-1-

tpcc_iitem_imp='index'
tpcc_iitem_size='2048'
tpcc_iitem_ext='calc'
tpcc_iitem_nf='calc'
tpcc_iitem_bs='auto'
tpcc_iitem_used='-1'
tpcc_iitem_free='5'
tpcc_iitem_trans='4'
tpcc_iitem_autospace='t'
tpcc_iitem_flg='43'
tpcc_iitem_fl='22'
tpcc_iitem_rsize='auto'
tpcc_iitem_hkey='auto'
tpcc_iitem_hash='auto'
tpcc_iitem_bpool='default'
tpcc_iitem_indices=1-

tpcc_inord_imp='none'
tpcc_inord_size='229'
tpcc_inord_ext='calc'
tpcc_inord_nf='calc'
tpcc_inord_bs='auto'
tpcc_inord_used='-1'
tpcc_inord_free='5'
tpcc_inord_trans='4'
tpcc_inord_autospace='t'
tpcc_inord_flg='43'
tpcc_inord_fl='22'
tpcc_inord_rsize='auto'
tpcc_inord_hkey='auto'
tpcc_inord_hash='auto'
tpcc_inord_bpool='default'
tpcc_inord_indices=1-2-3-

tpcc_iordl_imp='none'
tpcc_iordl_size='8072'
tpcc_iordl_ext='calc'
tpcc_iordl_nf='calc'
```


Copyright © 2004 NEC Corporation

```
tpcc_iordl_bs='auto'
tpcc_iordl_used='-1'
tpcc_iordl_free='5'
tpcc_iordl_trans='4'
tpcc_iordl_autospace='t'
tpcc_iordl_flg='43'
tpcc_iordl_fl='22'
tpcc_iordl_rsize='auto'
tpcc_iordl_hkey='auto'
tpcc_iordl_hash='auto'
tpcc_iordl_bpool='default'
tpcc_iordl_indices=1-2-3-4-

tpcc_iordr1_imp='none'
tpcc_iordr1_size='703'
tpcc_iordr1_ext='calc'
tpcc_iordr1_nf='calc'
tpcc_iordr1_bs='auto'
tpcc_iordr1_used='-1'
tpcc_iordr1_free='1'
tpcc_iordr1_trans='3'
tpcc_iordr1_autospace='t'
tpcc_iordr1_flg='43'
tpcc_iordr1_fl='22'
tpcc_iordr1_rsize='auto'
tpcc_iordr1_hkey='auto'
tpcc_iordr1_hash='auto'
tpcc_iordr1_bpool='default'
tpcc_iordr1_indices=2-3-1-

tpcc_iordr2_imp='index'
tpcc_iordr2_size='1135'
tpcc_iordr2_ext='calc'
tpcc_iordr2_nf='calc'
tpcc_iordr2_bs='auto'
tpcc_iordr2_used='-1'
tpcc_iordr2_free='25'
tpcc_iordr2_trans='4'
tpcc_iordr2_autospace='t'
tpcc_iordr2_flg='43'
tpcc_iordr2_fl='22'
tpcc_iordr2_rsize='auto'
tpcc_iordr2_hkey='auto'
tpcc_iordr2_hash='auto'
tpcc_iordr2_bpool='default'
tpcc_iordr2_indices=2-3-4-1-

tpcc_istok_imp='index'
tpcc_istok_size='2090'
tpcc_istok_ext='calc'
tpcc_istok_nf='calc'
tpcc_istok_bs='auto'
tpcc_istok_used='-1'
tpcc_istok_free='1'
tpcc_istok_trans='3'
tpcc_istok_autospace='t'
tpcc_istok_flg='43'
tpcc_istok_fl='22'
tpcc_istok_rsize='auto'
tpcc_istok_hkey='auto'
tpcc_istok_hash='auto'
tpcc_istok_bpool='default'
tpcc_istok_indices=1-2-

tpcc_iware_imp='index'
tpcc_iware_size='1'
tpcc_iware_ext='calc'
tpcc_iware_nf='calc'
tpcc_iware_bs='auto'
tpcc_iware_used='-1'
tpcc_iware_free='1'
tpcc_iware_trans='3'
tpcc_iware_autospace='t'
tpcc_iware_flg='43'
tpcc_iware_fl='22'
tpcc_iware_rsize='auto'
tpcc_iware_hkey='auto'
tpcc_iware_hash='auto'
tpcc_iware_bpool='default'
tpcc_iware_indices=1-

tpcc_temp_imp='temp'
tpcc_temp_size='16145'
tpcc_temp_ext='calc'
tpcc_temp_nf='calc'
tpcc_temp_bs='auto'
tpcc_temp_used='-1'
tpcc_temp_free='0'
tpcc_temp_trans='3'
tpcc_temp_autospace='t'
tpcc_temp_flg='43'
tpcc_temp_fl='22'
tpcc_temp_rsize='auto'
tpcc_temp_hkey='auto'
tpcc_temp_hash='auto'
tpcc_temp_bpool='default'
tpcc_temp_indices=no
```

localoptions.sh

```
#LOCAL OPTION FILE- You must fill these in
# before the driver will work.


#oracle sid to use for the run
ORACLE_SID=tpcc


#folder location of the database files (or links to raw partitions)
```


Copyright © 2004 NEC Corporation

```
tpcc_disks_location=/oradata/tpcc50000

#FOR NT
#tpcc_disks_location=\\\\.\\



#FOR RAC

#node id
#tpcc_rac_id=1


# How many createts_node*.sh will be run in this node, started from
tpcc_rac_id
# eq. if tpcc_rac_id is 3 and tpcc_rac_createts_count is 2
# createts_node3.sh and createts_node4.sh will be executed

#tpcc_rac_createts_count=1


#locations of various files used in the generation scripts.
#(you can usually leave these alone.)
tpcc_sql_dir=${tpcc_bench}/scripts/sql
tpcc_log_dir=${tpcc_bench}/log
tpcc_genscripts_dir=${tpcc_bench}/scripts/generated


#Once you have filled all the options, comment
#out or delete this line.
#tpcc_no_options=t
```

inittpcc.ora

```
ifile=/home/oracle/tpcc-kit50000/run.ora
```

run.ora

```
aq_tm_processes = 0
compatible = 10.1.0.0.0
control_files = '/oradata/tpcc50000/control_001','/oradata/tpcc50000/control_002'
cursor_space_for_time = TRUE
db_16k_cache_size = 49392123904
db_4k_cache_size = 268435456
db_8k_cache_size = 939524096
db_block_checking = FALSE
db_block_checksum = FALSE
db_block_size = 2048
db_cache_size = 43083890688
db_files = 1100
db_keep_cache_size = 289910292480
db_name = tpcc
db_recycle_cache_size = 82678120448
db_writer_processes = 8
disk_asynch_io = true
dml_locks = 500
fast_start_mttr_target = 0
java_pool_size = 0
log_buffer = 33554432
log_checkpoint_interval = 0
log_checkpoint_timeout = 0
log_checkpoints_to_alert = TRUE
parallel_max_servers = 256
pga_aggregate_target = 0
plsql_optimize_level = 2
processes = 650
recovery_parallelism = 64
replication_dependency_tracking = false
sessions = 1300
shared_pool_size = 15556673536
statistics_level = basic
timed_statistics = false
transactions = 1000
undo_management = AUTO
undo_retention = 1
undo_tablespace = undo_1
query_rewrite_enabled = false
_db_cache_pre_warm = false
_cursor_cache_frame_bind_memory = TRUE
_db_writer_flush_imu = FALSE
_imu_pools = 900
_lgwr_async_io = FALSE
_two_pass = FALSE
_lightweight_hdrs = true
```

p_create.ora

```
compatible = 10.1.0.0.0
db_name = tpcc
control_files =
'/oradata/tpcc50000/control_001','/oradata/tpcc50000/control_002'
```


Copyright © 2004 NEC Corporation

```
db_block_size = 2048
db_cache_size = 45000M
db_8k_cache_size = 20000M
log_buffer = 1048576
db_16k_cache_size = 20000M
undo_management = manual
shared_pool_size = 6625M
plsql_optimize_level=2
db_4k_cache_size = 400M
```

loadware.sh

```
Cd $tpcc_bench
$tpcc_load -M $tpcc_scale -w > loadware.log 2>&1
```

loaddist.sh

```
cd $tpcc_bench
$tpcc_load -M $tpcc_scale -d > loaddist.log 2>&1
```

loaditem.sh

```
cd $tpcc_bench
$tpcc_load -M $tpcc_scale -i > loaditem.log 2>&1
```

loadhist.sh

```
#created automatically by /home/oracle/tpcc-kit50000/scripts/evenload.sh
Fri Nov 28 18:29:37 JST 2003
rm -f loadhist*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 50000 -h  -b 1 -e 781 >> loadhist0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 782 -e 1562 >> loadhist1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 1563 -e 2343 >> loadhist2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 2344 -e 3124 >> loadhist3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 3125 -e 3905 >> loadhist4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 3906 -e 4686 >> loadhist5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 4687 -e 5467 >> loadhist6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 5468 -e 6248 >> loadhist7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 6249 -e 7029 >> loadhist8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 7030 -e 7810 >> loadhist9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 7811 -e 8591 >> loadhist10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 8592 -e 9372 >> loadhist11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 9373 -e 10153 >> loadhist12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 10154 -e 10934 >> loadhist13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 10935 -e 11715 >> loadhist14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 11716 -e 12496 >> loadhist15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 12497 -e 13277 >> loadhist16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 13278 -e 14058 >> loadhist17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 14059 -e 14839 >> loadhist18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 14840 -e 15620 >> loadhist19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 15621 -e 16401 >> loadhist20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 16402 -e 17182 >> loadhist21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 17183 -e 17963 >> loadhist22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 17964 -e 18744 >> loadhist23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 18745 -e 19525 >> loadhist24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 19526 -e 20306 >> loadhist25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 20307 -e 21087 >> loadhist26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 21088 -e 21868 >> loadhist27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 21869 -e 22649 >> loadhist28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 22650 -e 23430 >> loadhist29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 23431 -e 24211 >> loadhist30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 24212 -e 24992 >> loadhist31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 24993 -e 25773 >> loadhist32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 25774 -e 26554 >> loadhist33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 26555 -e 27335 >> loadhist34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 27336 -e 28116 >> loadhist35.log 2>&1 &
```


Copyright © 2004 NEC Corporation

```
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 28117 -e 28897 >> loadhist36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 28898 -e 29678 >> loadhist37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 29679 -e 30459 >> loadhist38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 30460 -e 31240 >> loadhist39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 31241 -e 32021 >> loadhist40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 32022 -e 32802 >> loadhist41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 32803 -e 33583 >> loadhist42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 33584 -e 34364 >> loadhist43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 34365 -e 35145 >> loadhist44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 35146 -e 35926 >> loadhist45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 35927 -e 36707 >> loadhist46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 36708 -e 37488 >> loadhist47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 37489 -e 38270 >> loadhist48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 38271 -e 39052 >> loadhist49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 39053 -e 39834 >> loadhist50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 39835 -e 40616 >> loadhist51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 40617 -e 41398 >> loadhist52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 41399 -e 42180 >> loadhist53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 42181 -e 42962 >> loadhist54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 42963 -e 43744 >> loadhist55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 43745 -e 44526 >> loadhist56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 44527 -e 45308 >> loadhist57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 45309 -e 46090 >> loadhist58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 46091 -e 46872 >> loadhist59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 46873 -e 47654 >> loadhist60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 47655 -e 48436 >> loadhist61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 48437 -e 49218 >> loadhist62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -h  -b 49219 -e 50000 >> loadhist63.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
  wait $curproc
  error=`expr $? + $error`
done
exit `expr $error != 0`
```

loadnord.sh

```
#created automatically by /home/oracle/tpcc-kit50000/scripts/evenload.sh
Fri Nov 28 18:29:38 JST 2003
rm -f loadnord*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 50000 -n  -b 1 -e 781 >> loadnord0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 782 -e 1562 >> loadnord1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 1563 -e 2343 >> loadnord2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 2344 -e 3124 >> loadnord3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 3125 -e 3905 >> loadnord4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 3906 -e 4686 >> loadnord5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 4687 -e 5467 >> loadnord6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 5468 -e 6248 >> loadnord7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 6249 -e 7029 >> loadnord8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 7030 -e 7810 >> loadnord9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 7811 -e 8591 >> loadnord10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 8592 -e 9372 >> loadnord11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 9373 -e 10153 >> loadnord12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 10154 -e 10934 >> loadnord13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 10935 -e 11715 >> loadnord14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 11716 -e 12496 >> loadnord15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 12497 -e 13277 >> loadnord16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 13278 -e 14058 >> loadnord17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 14059 -e 14839 >> loadnord18.log 2>&1 &
allprocs="$allprocs ${!}"
```


Copyright © 2004 NEC Corporation

```
$tpcc_load -M 50000 -n  -b 14840 -e 15620 >> loadnord19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 15621 -e 16401 >> loadnord20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 16402 -e 17182 >> loadnord21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 17183 -e 17963 >> loadnord22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 17964 -e 18744 >> loadnord23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 18745 -e 19525 >> loadnord24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 19526 -e 20306 >> loadnord25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 20307 -e 21087 >> loadnord26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 21088 -e 21868 >> loadnord27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 21869 -e 22649 >> loadnord28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 22650 -e 23430 >> loadnord29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 23431 -e 24211 >> loadnord30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 24212 -e 24992 >> loadnord31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 24993 -e 25773 >> loadnord32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 25774 -e 26554 >> loadnord33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 26555 -e 27335 >> loadnord34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 27336 -e 28116 >> loadnord35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 28117 -e 28897 >> loadnord36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 28898 -e 29678 >> loadnord37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 29679 -e 30459 >> loadnord38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 30460 -e 31240 >> loadnord39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 31241 -e 32021 >> loadnord40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 32022 -e 32802 >> loadnord41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 32803 -e 33583 >> loadnord42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 33584 -e 34364 >> loadnord43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 34365 -e 35145 >> loadnord44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 35146 -e 35926 >> loadnord45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 35927 -e 36707 >> loadnord46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 36708 -e 37488 >> loadnord47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 37489 -e 38270 >> loadnord48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 38271 -e 39052 >> loadnord49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 39053 -e 39834 >> loadnord50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 39835 -e 40616 >> loadnord51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 40617 -e 41398 >> loadnord52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 41399 -e 42180 >> loadnord53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 42181 -e 42962 >> loadnord54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 42963 -e 43744 >> loadnord55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 43745 -e 44526 >> loadnord56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 44527 -e 45308 >> loadnord57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 45309 -e 46090 >> loadnord58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 46091 -e 46872 >> loadnord59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 46873 -e 47654 >> loadnord60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 47655 -e 48436 >> loadnord61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 48437 -e 49218 >> loadnord62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -n  -b 49219 -e 50000 >> loadnord63.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
  wait $curproc
  error=`expr $? + $error`
done
exit `expr $error != 0`
```

loadordrordl.sh

```
#created automatically by /home/oracle/tpcc-kit50000/scripts/evenload.sh
Fri Nov 28 18:29:38 JST 2003
rm -f loadordrordl*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy0.dat -b 1 -e 781 >>
loadordrordl0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy1.dat -b 782 -e 1562
>> loadordrordl1.log 2>&1 &
```


Copyright © 2004 NEC Corporation

```
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy2.dat -b 1563 -e 2343
>> loadordrordl2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy3.dat -b 2344 -e 3124
>> loadordrordl3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy4.dat -b 3125 -e 3905
>> loadordrordl4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy5.dat -b 3906 -e 4686
>> loadordrordl5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy6.dat -b 4687 -e 5467
>> loadordrordl6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy7.dat -b 5468 -e 6248
>> loadordrordl7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy8.dat -b 6249 -e 7029
>> loadordrordl8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy9.dat -b 7030 -e 7810
>> loadordrordl9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy10.dat -b 7811 -e 8591
>> loadordrordl10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy11.dat -b 8592 -e 9372
>> loadordrordl11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy12.dat -b 9373 -e
10153 >> loadordrordl12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy13.dat -b 10154 -e
10934 >> loadordrordl13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy14.dat -b 10935 -e
11715 >> loadordrordl14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy15.dat -b 11716 -e
12496 >> loadordrordl15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy16.dat -b 12497 -e
13277 >> loadordrordl16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy17.dat -b 13278 -e
14058 >> loadordrordl17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy18.dat -b 14059 -e
14839 >> loadordrordl18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy19.dat -b 14840 -e
15620 >> loadordrordl19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy20.dat -b 15621 -e
16401 >> loadordrordl20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy21.dat -b 16402 -e
17182 >> loadordrordl21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy22.dat -b 17183 -e
17963 >> loadordrordl22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy23.dat -b 17964 -e
18744 >> loadordrordl23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy24.dat -b 18745 -e
19525 >> loadordrordl24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy25.dat -b 19526 -e
20306 >> loadordrordl25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy26.dat -b 20307 -e
21087 >> loadordrordl26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy27.dat -b 21088 -e
21868 >> loadordrordl27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy28.dat -b 21869 -e
22649 >> loadordrordl28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy29.dat -b 22650 -e
23430 >> loadordrordl29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy30.dat -b 23431 -e
24211 >> loadordrordl30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy31.dat -b 24212 -e
24992 >> loadordrordl31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy32.dat -b 24993 -e
25773 >> loadordrordl32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy33.dat -b 25774 -e
26554 >> loadordrordl33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy34.dat -b 26555 -e
27335 >> loadordrordl34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy35.dat -b 27336 -e
28116 >> loadordrordl35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy36.dat -b 28117 -e
28897 >> loadordrordl36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy37.dat -b 28898 -e
29678 >> loadordrordl37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy38.dat -b 29679 -e
30459 >> loadordrordl38.log 2>&1 &
```


Copyright © 2004 NEC Corporation

```
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy39.dat -b 30460 -e
31240 >> loadordrordl39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy40.dat -b 31241 -e
32021 >> loadordrordl40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy41.dat -b 32022 -e
32802 >> loadordrordl41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy42.dat -b 32803 -e
33583 >> loadordrordl42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy43.dat -b 33584 -e
34364 >> loadordrordl43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy44.dat -b 34365 -e
35145 >> loadordrordl44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy45.dat -b 35146 -e
35926 >> loadordrordl45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy46.dat -b 35927 -e
36707 >> loadordrordl46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy47.dat -b 36708 -e
37488 >> loadordrordl47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy48.dat -b 37489 -e
38270 >> loadordrordl48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy49.dat -b 38271 -e
39052 >> loadordrordl49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy50.dat -b 39053 -e
39834 >> loadordrordl50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy51.dat -b 39835 -e
40616 >> loadordrordl51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy52.dat -b 40617 -e
41398 >> loadordrordl52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy53.dat -b 41399 -e
42180 >> loadordrordl53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy54.dat -b 42181 -e
42962 >> loadordrordl54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy55.dat -b 42963 -e
43744 >> loadordrordl55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy56.dat -b 43745 -e
44526 >> loadordrordl56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy57.dat -b 44527 -e
45308 >> loadordrordl57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy58.dat -b 45309 -e
46090 >> loadordrordl58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy59.dat -b 46091 -e
46872 >> loadordrordl59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy60.dat -b 46873 -e
47654 >> loadordrordl60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy61.dat -b 47655 -e
48436 >> loadordrordl61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy62.dat -b 48437 -e
49218 >> loadordrordl62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -o ${tpcc_disks_location}dummy63.dat -b 49219 -e
50000 >> loadordrordl63.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
  wait $curproc
  error=`expr $? + $error`
done
exit `expr $error != 0`
```

loadcust.sh

```
#created automatically by /home/oracle/tpcc-kit50000/scripts/evenload.sh
Fri Nov 28 18:29:39 JST 2003
rm -f loadcust*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 50000 -c  -b 1 -e 781 >> loadcust0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 782 -e 1562 >> loadcust1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 1563 -e 2343 >> loadcust2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 2344 -e 3124 >> loadcust3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 3125 -e 3905 >> loadcust4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 3906 -e 4686 >> loadcust5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 4687 -e 5467 >> loadcust6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 5468 -e 6248 >> loadcust7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 6249 -e 7029 >> loadcust8.log 2>&1 &
allprocs="$allprocs ${!}"
```


Copyright © 2004 NEC Corporation

```
$tpcc_load -M 50000 -c  -b 7030 -e 7810 >> loadcust9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 7811 -e 8591 >> loadcust10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 8592 -e 9372 >> loadcust11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 9373 -e 10153 >> loadcust12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 10154 -e 10934 >> loadcust13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 10935 -e 11715 >> loadcust14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 11716 -e 12496 >> loadcust15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 12497 -e 13277 >> loadcust16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 13278 -e 14058 >> loadcust17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 14059 -e 14839 >> loadcust18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 14840 -e 15620 >> loadcust19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 15621 -e 16401 >> loadcust20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 16402 -e 17182 >> loadcust21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 17183 -e 17963 >> loadcust22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 17964 -e 18744 >> loadcust23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 18745 -e 19525 >> loadcust24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 19526 -e 20306 >> loadcust25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 20307 -e 21087 >> loadcust26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 21088 -e 21868 >> loadcust27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 21869 -e 22649 >> loadcust28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 22650 -e 23430 >> loadcust29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 23431 -e 24211 >> loadcust30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 24212 -e 24992 >> loadcust31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 24993 -e 25773 >> loadcust32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 25774 -e 26554 >> loadcust33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 26555 -e 27335 >> loadcust34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 27336 -e 28116 >> loadcust35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 28117 -e 28897 >> loadcust36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 28898 -e 29678 >> loadcust37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 29679 -e 30459 >> loadcust38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 30460 -e 31240 >> loadcust39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 31241 -e 32021 >> loadcust40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 32022 -e 32802 >> loadcust41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 32803 -e 33583 >> loadcust42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 33584 -e 34364 >> loadcust43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 34365 -e 35145 >> loadcust44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 35146 -e 35926 >> loadcust45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 35927 -e 36707 >> loadcust46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 36708 -e 37488 >> loadcust47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 37489 -e 38270 >> loadcust48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 38271 -e 39052 >> loadcust49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 39053 -e 39834 >> loadcust50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 39835 -e 40616 >> loadcust51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 40617 -e 41398 >> loadcust52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 41399 -e 42180 >> loadcust53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 42181 -e 42962 >> loadcust54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 42963 -e 43744 >> loadcust55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 43745 -e 44526 >> loadcust56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 44527 -e 45308 >> loadcust57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 45309 -e 46090 >> loadcust58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 46091 -e 46872 >> loadcust59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 46873 -e 47654 >> loadcust60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 47655 -e 48436 >> loadcust61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 48437 -e 49218 >> loadcust62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -c  -b 49219 -e 50000 >> loadcust63.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
```


Copyright © 2004 NEC Corporation

```
  wait $curproc
  error=`expr $? + $error`
done
exit `expr $error != 0`
```

loadstok.sh

```
#created automatically by /home/oracle/tpcc-kit50000/scripts/evenload.sh
Fri Nov 28 18:29:39 JST 2003
rm -f loadstok*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 50000 -S  -j 1 -k 1562 >> loadstok0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 1563 -k 3124 >> loadstok1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 3125 -k 4686 >> loadstok2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 4687 -k 6248 >> loadstok3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 6249 -k 7810 >> loadstok4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 7811 -k 9372 >> loadstok5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 9373 -k 10934 >> loadstok6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 10935 -k 12496 >> loadstok7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 12497 -k 14058 >> loadstok8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 14059 -k 15620 >> loadstok9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 15621 -k 17182 >> loadstok10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 17183 -k 18744 >> loadstok11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 18745 -k 20306 >> loadstok12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 20307 -k 21868 >> loadstok13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 21869 -k 23430 >> loadstok14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 23431 -k 24992 >> loadstok15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 24993 -k 26554 >> loadstok16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 26555 -k 28116 >> loadstok17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 28117 -k 29678 >> loadstok18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 29679 -k 31240 >> loadstok19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 31241 -k 32802 >> loadstok20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 32803 -k 34364 >> loadstok21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 34365 -k 35926 >> loadstok22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 35927 -k 37488 >> loadstok23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 37489 -k 39050 >> loadstok24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 39051 -k 40612 >> loadstok25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 40613 -k 42174 >> loadstok26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 42175 -k 43736 >> loadstok27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 43737 -k 45298 >> loadstok28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 45299 -k 46860 >> loadstok29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 46861 -k 48422 >> loadstok30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 48423 -k 49984 >> loadstok31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 49985 -k 51547 >> loadstok32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 51548 -k 53110 >> loadstok33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 53111 -k 54673 >> loadstok34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 54674 -k 56236 >> loadstok35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 56237 -k 57799 >> loadstok36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 57800 -k 59362 >> loadstok37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 59363 -k 60925 >> loadstok38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 60926 -k 62488 >> loadstok39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 62489 -k 64051 >> loadstok40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 64052 -k 65614 >> loadstok41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 65615 -k 67177 >> loadstok42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 67178 -k 68740 >> loadstok43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 68741 -k 70303 >> loadstok44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 70304 -k 71866 >> loadstok45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 71867 -k 73429 >> loadstok46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 73430 -k 74992 >> loadstok47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 74993 -k 76555 >> loadstok48.log 2>&1 &
```


Copyright © 2004 NEC Corporation

```
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 76556 -k 78118 >> loadstok49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 78119 -k 79681 >> loadstok50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 79682 -k 81244 >> loadstok51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 81245 -k 82807 >> loadstok52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 82808 -k 84370 >> loadstok53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 84371 -k 85933 >> loadstok54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 85934 -k 87496 >> loadstok55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 87497 -k 89059 >> loadstok56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 89060 -k 90622 >> loadstok57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 90623 -k 92185 >> loadstok58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 92186 -k 93748 >> loadstok59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 93749 -k 95311 >> loadstok60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 95312 -k 96874 >> loadstok61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 96875 -k 98437 >> loadstok62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 50000 -S  -j 98438 -k 100000 >> loadstok63.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
  wait $curproc
  error=`expr $? + $error`
done
exit `expr $error != 0`
```

createts.sh

```
#!/bin/sh
#NOTE - ANY CHANGES MUST BE MADE TO CREATETS.KSH AS WELL.
# createts.sh [name] [no. of file] [no. of partition] [filesize]
[ext_size]
#             [unix/nt] [1: temporary ts / 0: others] [filecount] [no of
cpu]
#            [blocksize] [t: bitmapped / f: manual manage / d:
dictionary ]

name=$1
fileno=$2
noofts=$3
filesize=$4
extsize=$5
ver=$6
isTemp=$7
filecount=$8
para=`expr $9 \* 2`
#blocksize=${10}  sh bug workaround
blocksize=`echo $@ | cut -d' ' -f10`
#autospace=${11} sh bug workaround
autospace=`echo $@ | cut -d' ' -f11`

addts=$tpcc_scripts/addts.sh
addfile=$tpcc_scripts/addfile.sh

if expr "x$tpcc_createts_print" = "xt" > /dev/null ; then
  createtsout=${tpcc_genscripts_dir}/createts_node${tpcc_rac_node}.sh
  fileavg=`expr $fileno / $tpcc_np`

  if test $noofts -gt 1 ; then
    avg_ts_node=`expr $noofts / $tpcc_np`
    if test "x$tpcc_rac_createts_phase" = "x1" ; then
       fileavg=$avg_ts_node
    else
      if test "x$tpcc_rac_createts_phase" = "x2" ; then
       fileavg=`expr $fileavg - $avg_ts_node`
      fi
    fi
  fi

  fileend=`expr $fileavg \* $tpcc_rac_node`
  filestart=`expr $fileend - $fileavg`
  if expr $tpcc_rac_node = $tpcc_np > /dev/null; then
    fileend=$fileno
  fi
fi

if test $ver = unix;
then
  fileaddr=$tpcc_disks_location;
elif test $ver = nt;
then
  fileaddr=\\\\\\.\\\\
fi

filecounter=0
i=0
while test $i -lt $noofts; do

  filecount=`expr $filecount + 1`;
  if expr "x$tpcc_createts_print" = "xt" > /dev/null ; then
    if test "x$tpcc_rac_createts_phase" = "x1" ; then
      if test "x$name" = "xitem" -o "x$name" = "xiitem" -o "x$name" =
"xtemp" -o "x$name" = "xrestbl" ; then
        if test $tpcc_rac_node = 1 ; then
          echo $addts $name\_$i $fileaddr$name\_$i\_0 $filesize $extsize
$blocksize $isTemp $autospace \& >> $createtsout
          rac_count=`expr $rac_count + 1`
```


Copyright © 2004 NEC Corporation

```
          if test "$rac_count" = "$para" ; then
            rac_count=0
            echo wait >> $createtsout
          fi
        fi
      else
        if test $filecounter -ge $filestart -a $filecounter -lt
$fileend ; then
          echo $addts $name\_$i $fileaddr$name\_$i\_0 $filesize $extsize
$blocksize $isTemp $autospace \& >> $createtsout
          rac_count=`expr $rac_count + 1`
          if test "$rac_count" = "$para" ; then
            rac_count=0
            echo wait >> $createtsout
          fi
        fi
      fi
    fi
  else
    $addts $name\_$i $fileaddr$name\_$i\_0 $filesize $extsize $blocksize
$isTemp $autospace \> junk$filecount 2\>\&1 \&;
  fi
  eval "proc$filecount=$!"
  filecounter=`expr $filecounter + 1`

  p=`expr $filecount % $para`;
  if test $p = 0;
  then
    k=`expr $filecount - $para + 1`;
    if test $k -le $8;
    then
      k=`expr $8 + 1`;
    fi
    while test $k -le $filecount ; do
#       wait `eval echo '$proc'$k`
      wait
      eval "proc$k=$?"
      k=`expr $k + 1`;
    done
  fi

  i=`expr $i + 1`;

done

p=`expr $filecount % $para`
if test $p != 0;
then
  k=`expr $filecount - $p + 1`;
  if test $k -le $8;
  then
    k=`expr $8 + 1`;
  fi
  while test $k -le $filecount; do
#    wait `eval echo '$proc'$k`
    wait
    eval "proc$k=$?"
    k=`expr $k + 1`
  done
fi

if test "x$tpcc_createts_print" = "xt" -a "x$tpcc_rac_createts_phase" =
"x1" ; then
  echo $rac_count
  exit 0
fi

if test "x$tpcc_createts_print" = "xt" -a $noofts -gt 1 -a
"x$tpcc_rac_createts_phase" = "x2" ; then
  filecounter=0
fi

filecount=0
fileperts=`expr $fileno / $noofts - 1`
if test $fileperts -gt 0 ;
then
  i=0
  while test $i -lt $noofts ; do
    j=0;
    while test $j -lt $fileperts ;do

      filecount=`expr $filecount + 1`;
      if expr "x$tpcc_createts_print" = "xt" > /dev/null ; then
        if test "x$tpcc_rac_createts_phase" = "x2" ; then
          if test "x$name" = "xitem" -o "x$name" = "xtemp" -o "x$name" =
"xrestbl" ; then
            if test $tpcc_rac_node = 1 ; then
              echo $addfile $name\_$i $fileaddr$name\_$i\_`expr $j + 1`
$filesize $isTemp \& >> $createtsout
              rac_count=`expr $rac_count + 1`
              if test "$rac_count" = "$para" ; then
                rac_count=0
                echo wait >> $createtsout
              fi
            fi
          else
            if test $filecounter -ge $filestart -a $filecounter -lt
$fileend ; then
              echo $addfile $name\_$i $fileaddr$name\_$i\_`expr $j + 1`
$filesize $isTemp \& >> $createtsout
              rac_count=`expr $rac_count + 1`
              if test "$rac_count" = "$para" ; then
                rac_count=0
                echo wait >> $createtsout
              fi
            fi
          fi
        fi
      else
        $addfile $name\_$i $fileaddr$name\_$i\_`expr $j + 1` $filesize
$isTemp \> junk$filecount 2\>\&1 &
      fi
```


Copyright © 2004 NEC Corporation

```
      eval "proc$filecount=$!"

      filecounter=`expr $filecounter + 1`

      p=`expr $filecount % $para`;
      if test $p = 0;
      then
        k=`expr $filecount - $para + 1`;
        if test $k -le $8;
        then
          k=`expr $8 + 1`;
        fi
       while test $k -le $filecount ; do
#          wait `eval echo '$proc'$k`
          wait
          eval "proc$k=$?"
          k=`expr $k + 1`;
        done
      fi

      j=`expr $j + 1`
    done

    i=`expr $i + 1`
  done

  p=`expr $filecount % $para`
  if test $p != 0;
  then
    k=`expr $filecount - $p + 1`;
    if test $k -le $8;
    then
      k=`expr $8 + 1`;
    fi
    while test $k -le $filecount; do
#      wait `eval echo '$proc'$k`
      wait
      eval "proc$k=$?"
      k=`expr $k + 1`
    done
  fi
fi

if test "x$tpcc_createts_print" = "xt" ; then
  echo $rac_count
fi

i=`expr $8 + 1`
proc=0
while test $i -le $filecount ;do
  eval 'process=$proc'"$i"
  proc=`expr $proc + $process`
  i=`expr $i + 1`
done

out=`expr $proc % 127`
# Added wait here for all tablespaces to be created
wait
if test $out -ne 0
then
  exit 1;
else
  exit 0;
fi
```

create_cache_views.sql

```
rem This script creates four views that when queried will return
rem the total number of buffers in the buffer cache and the total
rem number of cloned buffers from each of the database's tablespaces.
rem
rem This assumes that each table and index is in its own tablespace.
rem If this is not the case, another query can be used which uses the
rem database's object tables to decipher the different objects.  However,
rem this query is slower.
rem
rem This script assumes 7.3.x.  If you are using V7.2.x or below, please
rem replace svrmgrl with sqldba lmode=y.
rem
rem Modification History:
rem
rem wbattist      16-Jun-1996     Create two additional views to keep
rem                               track of the number of clones in each
rem                               tablespace.
rem
rem wbattist      24-May-1995     Add the state check for the cbf view
rem                               to ensure that cloned blocks are not
rem                               counted.
rem

connect $oracle_dba/$oracle_dba_password;
set echo on;
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
```


Copyright © 2004 NEC Corporation

```
 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;

set echo off;
```

extent.sql

```
REM            Copyright (c) 1994  Oracle Corp, Belmont, CA          |
REM                   OPEN SYSTEMS PERFORMANCE GROUP                 |
REM                        All Rights Reserved                       |
REM=================================================================+
REM FILENAME
REM       extent.sql
REM DESCRIPTION
REM       List all extents in all the TPCC tablespaces.
REM
REM  Usage: sqlplus 'sys/change_on_install as sysdba' @extent
REM=================================================================*/
      set space 2
      set pagesize 2000
      set echo off
      set termout off
      set verify off
      set feedback off
      spool extent.rpt
      select substr(e.tablespace_name,1,8) tspace,
        substr(segment_name,1,11) segment, substr(segment_type,1,15)
type,
        substr(extent_id,1,5) eid, substr(file_id,1,5) fid, blocks,
        blocks * t.block_size / 1048576 size_MB
      from   dba_extents e, dba_tablespaces t
      where  owner = 'TPCC' AND ( segment_type = 'INDEX' OR
        segment_type = 'INDEX PARTITION' OR segment_type = 'CLUSTER'
        OR segment_type = 'TABLE' OR segment_type = 'TABLE PARTITION')
        AND e.tablespace_name <> 'SYSTEM'
        AND e.tablespace_name = t.tablespace_name
      order by e.tablespace_name, segment_name, extent_id, file_id;

      select substr(e.tablespace_name,1,8) tspace,
        substr(segment_name,1,11) segment,
        sum(blocks) tot_blk, sum(blocks) * t.block_size / 1048576
size_MB
      from   dba_extents e, dba_tablespaces t
      where  owner = 'TPCC' AND ( segment_type = 'INDEX' OR
        segment_type = 'INDEX PARTITION' OR segment_type = 'CLUSTER'
        OR segment_type = 'TABLE' OR segment_type = 'TABLE PARTITION')
        AND e.tablespace_name <> 'SYSTEM'
        AND e.tablespace_name = t.tablespace_name
      group by e.tablespace_name, segment_name, t.block_size
      order by e.tablespace_name, segment_name;
      spool off;
```

pst_c.sql

```
rem
rem
rem ================================================================+
rem            Copyright (c) 1992  Oracle Corp, Belmont, CA          |
rem                   OPEN SYSTEMS PERFORMANCE GROUP                 |
rem                        All Rights Reserved                       |
rem ================================================================+
rem  FILENAME
rem       pst_c.sql
rem  DESCRIPTION
rem       Create Table for OS Specific Process Stats
rem ================================================================*/
rem
rem  Tables for Unix-specific process statistics
rem
rem  Usage: sqlplus internal/internal @pst_c
rem

        connect tpcc/tpcc;
        set echo on;
        DROP TABLE proc_resource;
        DROP TABLE os_stat;

rem
rem  Resource usage for a process.
rem

        CREATE TABLE proc_resource
        (
            config        VARCHAR2(10),
            run           NUMBER,
            proc          NUMBER,
            child         NUMBER,
            user_cpu_ms   NUMBER,
            system_cpu_ms NUMBER,
            maxrss        NUMBER,
            pagein        NUMBER,
            reclaim       NUMBER,
            zerofill      NUMBER,
            pffincr       NUMBER,
            pffdecr       NUMBER,
            swap          NUMBER,
            syscall       NUMBER,
            volcsw        NUMBER,
            involcsw      NUMBER,
            signal        NUMBER,
            lread         NUMBER,
```


Copyright © 2004 NEC Corporation

```
            lwrite        NUMBER,
            bread         NUMBER,
            bwrite        NUMBER,
            phread        NUMBER,
            phwrite       NUMBER
        );

rem
rem  OS statistics.
rem  These results are from the measurement interval only.
rem

        CREATE TABLE os_stat
        (
            config          VARCHAR2(10),
            run             NUMBER,
            hid             NUMBER,
            syscall         NUMBER,
            intr            NUMBER,
            cswitch         NUMBER,
            pagefault       NUMBER,
            usr             NUMBER,
            sys             NUMBER,
            idl             NUMBER,
            wio             NUMBER
        );

        set echo off;
```

space_get.sql

```
REM=================================================================+
REM         Copyright (c) 1995  Oracle Corp, Redwood Shores, CA      |
REM                   OPEN SYSTEMS PERFORMANCE GROUP                 |
REM                        All Rights Reserved                       |
REM=================================================================+
REM FILENAME
REM       space_get.sql
REM DESCRIPTION
REM       Get sizes of tables, indexes and tablespaces.
REM  Usage: sqlplus 'sys/change_on_install as sysdba' @space_get [<tpm>
<# of warehouses>]
REM=================================================================*/

   set echo on;
   delete from tpcc_data;
   delete from tpcc_space;
   delete from tpcc_totspace;

   insert into tpcc_data
   select substr(segment_name,1,18), substr(segment_type,1,15),
          sum(blocks), t.block_size,
          round(sum(blocks) * 0.05), 0,
          sum(blocks) + round(sum(blocks) * 0.05)
   from   dba_extents e, dba_tablespaces t
   where  owner = 'TPCC' AND ( segment_type = 'INDEX' OR
          segment_type = 'INDEX PARTITION' OR segment_type = 'CLUSTER'
          OR segment_type = 'TABLE' OR segment_type = 'TABLE PARTITION')
          AND e.tablespace_name <> 'SYSTEM' AND e.tablespace_name <>
'SP_0'
          AND e.tablespace_name = t.tablespace_name
   group by segment_name, segment_type, t.block_size;

   insert into tpcc_data
      select 'SYSTEM', 'SYS', sum(blocks), t.block_size, 0, 0,
sum(blocks)
      from   dba_data_files f, dba_tablespaces t
      where  f.tablespace_name = 'SYSTEM' and t.tablespace_name =
f.tablespace_name
      group by t.block_size;

   insert into tpcc_data
      select 'SYSAUX', 'SYS', sum(blocks), t.block_size, 0, 0,
sum(blocks)
      from   dba_data_files f, dba_tablespaces t
      where  f.tablespace_name = 'SYSAUX' and t.tablespace_name =
f.tablespace_name
      group by t.block_size;

   insert into tpcc_data
      select 'ROLL_SEG', 'SYS', sum(blocks), t.block_size, 0, 0,
sum(blocks)
      from   dba_data_files f, dba_tablespaces t
      where  f.tablespace_name like '%UNDO_1%' and f.tablespace_name =
t.tablespace_name
      group by f.tablespace_name, t.block_size;

   insert into tpcc_data
      select 'DB_STAT', 'SYS', sum(blocks), t.block_size, 0, 0,
sum(blocks)
      from   dba_data_files f, dba_tablespaces t
      where  f.tablespace_name like '%SP_0%' and f.tablespace_name =
t.tablespace_name
      group by f.tablespace_name, t.block_size;

   update tpcc_data
      set five_pct = 0,
          daily_grow = round(blocks * &&1 / 62.5 / &&2),
          total = blocks + round(blocks * &&1 / 62.5 / &&2)
      where segment = 'HIST' OR segment = 'ORDRCLUSTER_QUEUE' OR
            segment = 'IORDL';

   insert into tpcc_space
   select substr(ex$.name,1,18), sum(sp$.sz_blocks), sp$.block_size, 0,
0, 0, 0
   from
     (select f.tablespace_name , sum(blocks) sz_blocks, t.block_size
block_size
      from dba_data_files f, dba_tablespaces t
```


Copyright © 2004 NEC Corporation

```
      where f.tablespace_name <> 'SYSTEM' and f.tablespace_name =
t.tablespace_name
      group by f.tablespace_name, t.block_size
     ) sp$,
     (select distinct tablespace_name, segment_name name
      from dba_extents
      where owner = 'TPCC'
        and (segment_type = 'CLUSTER' or segment_type = 'TABLE'
          or segment_type = 'TABLE PARTITION' or segment_type = 'INDEX'
          or segment_type = 'INDEX PARTITION')
        and tablespace_name <> 'SYSTEM'
     ) ex$
   where sp$.tablespace_name = ex$.tablespace_name
   group by ex$.name, sp$.block_size;

   insert into tpcc_space
   select substr(f.tablespace_name,1,18), sum(blocks), t.block_size, 0,
0, 0, 0
   from dba_data_files f, dba_tablespaces t
   where (f.tablespace_name = 'SYSTEM' or f.tablespace_name = 'SYSAUX')
         and f.tablespace_name = t.tablespace_name
   group by f.tablespace_name, t.block_size;

   insert into tpcc_space
   select 'ROLL_SEG', sum(blocks), t.block_size, 0, 0, 0, 0
   from dba_data_files f, dba_tablespaces t
   where f.tablespace_name = 'UNDO_1' and f.tablespace_name =
t.tablespace_name
   group by f.tablespace_name, t.block_size;

   insert into tpcc_space
   select 'DB_STAT', sum(blocks), t.block_size, 0, 0, 0, 0
   from dba_data_files f, dba_tablespaces t
   where f.tablespace_name = 'SP_0' and f.tablespace_name =
t.tablespace_name
   group by f.tablespace_name, t.block_size;

   update tpcc_space
      set required =
      (
         select sum(total)
         from   tpcc_data
         where  tpcc_data.segment = tpcc_space.segment
      )
      where segment in
      (
         select segment from tpcc_data
      );

   update tpcc_space
      set static =
      (
         select sum(total)
         from   tpcc_data
         where  tpcc_data.segment = tpcc_space.segment
      )
      where segment in
      (
         select segment from tpcc_data
      );

   update tpcc_space
      set static = 0,
          dynamic =
      (
         select sum(blocks)
         from   tpcc_data
         where  tpcc_data.segment = tpcc_space.segment
      )
      where segment in ('HIST', 'ORDRCLUSTER_QUEUE', 'IORDL');

   update tpcc_space
      set oversize = blocks  - required;


   insert into tpcc_totspace
      select &&1, &&2, sum(static * block_size)/1024, sum(dynamic *
block_size)/1024, sum(oversize * block_size)/1024, 0, 0, 0
      from   tpcc_space;

   update tpcc_totspace
      set daily_grow =
      (
         select sum(daily_grow * block_size)/1024
         from   tpcc_data
      );
   update tpcc_totspace
      set space60 = static + 60 * daily_grow;
   set echo off;
```

space_init.sql

```
REM==================================================================+
REM FILENAME
REM       space_init.sql
REM DESCRIPTION
REM       Create tables for space calculations.
REM  Usage: sqlplus 'sys/change_on_install as sysdba' @space_init.sql
REM==================================================================*/
   set echo on;
   drop table tpcc_data;
   drop table tpcc_space;
   drop table tpcc_totspace;
   create table tpcc_data (
      segment      varchar2(18),
      type         varchar2(15),
      blocks       number,
      block_size   number,
      five_pct     number,
      daily_grow   number,
```


Copyright © 2004 NEC Corporation

```
   total        number
);
create table tpcc_space (
   segment       varchar2(18),
   blocks       number,
   block_size   number,
   required     number,
   static       number,
   dynamic      number,
   oversize     number
);
create table tpcc_totspace (
   tpm          number,
   nware        number,
   static       number,
   dynamic      number,
   oversize     number,
   daily_grow   number,
   daily_spre   number,
   space60      number
);
create unique index itpcc_data on tpcc_data (segment);
create unique index itpcc_space on tpcc_space (segment);
set echo off;
```

space_rpt.sql

```
REM=================================================================+
REM         Copyright (c) 1995  Oracle Corp, Redwood Shores, CA      |
REM                    OPEN SYSTEMS PERFORMANCE GROUP                |
REM                         All Rights Reserved                      |
REM=================================================================+
REM FILENAME
REM       space_rpt.sql
REM DESCRIPTION
REM       Generate space report and save it in space.rpt
REM  Usage: sqlplus 'sys/change_on_install as sysdba' @space_rpt.sql
REM=================================================================*/
   set space 2
   set pagesize 2000
   set echo off
   set termout off
   set verify off
   set feedback off
   set pagesize 60 linesize 120
   spool space.rpt
   select tpm, nware from tpcc_totspace;
   select * from tpcc_data order by segment;
   select * from tpcc_space order by segment;
   select static, dynamic, oversize, daily_grow, daily_spre, space60
      from tpcc_totspace;
   spool off;
```

analyze.sh

```
#!/bin/sh
$tpcc_sqlplus $tpcc_user_pass @${tpcc_sql_dir}/analyze >
$tpcc_log_dir/junk 2>&1

# this one tends to fail if indices aren't made, which is legal, so
# always exit without error.

exit 0
```

analyze.sql

```
spool analyze.log;
set echo on;

connect tpcc/tpcc

execute dbms_stats.GATHER_TABLE_STATS (OWNNAME=>'TPCC', -
                                       TABNAME=>'STOK', -
                                       PARTNAME=>NULL, -
                                       ESTIMATE_PERCENT=>1, -
                                       BLOCK_SAMPLE=>TRUE, -
                                       METHOD_OPT=>'FOR ALL COLUMNS SIZE
1', -
                                       DEGREE=>10, -
                                       GRANULARITY=>'DEFAULT', -
                                       CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS (OWNNAME=>'TPCC', -
                                       TABNAME=>'CUST', -
                                       PARTNAME=>NULL, -
                                       ESTIMATE_PERCENT=>1, -
                                       BLOCK_SAMPLE=>TRUE, -
                                       METHOD_OPT=>'FOR ALL COLUMNS SIZE
1', -
                                       DEGREE=>10, -
                                       GRANULARITY=>'DEFAULT', -
                                       CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS (OWNNAME=>'TPCC', -
                                       TABNAME=>'ORDR', -
                                       PARTNAME=>NULL, -
                                       ESTIMATE_PERCENT=>1, -
                                       BLOCK_SAMPLE=>TRUE, -
                                       METHOD_OPT=>'FOR ALL COLUMNS SIZE
1', -
                                       DEGREE=>10, -
                                       GRANULARITY=>'DEFAULT', -
                                       CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS (OWNNAME=>'TPCC', -
                                       TABNAME=>'ORDL', -
```


Copyright © 2004 NEC Corporation

```
                                    PARTNAME=>NULL, -
                                    ESTIMATE_PERCENT=>1, -
                                    BLOCK_SAMPLE=>TRUE, -
                                    METHOD_OPT=>'FOR ALL COLUMNS SIZE
1', -
                                    DEGREE=>10, -
                                    GRANULARITY=>'DEFAULT', -
                                    CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS (OWNNAME=>'TPCC', -
                                    TABNAME=>'NORD', -
                                    PARTNAME=>NULL, -
                                    ESTIMATE_PERCENT=>1, -
                                    BLOCK_SAMPLE=>TRUE, -
                                    METHOD_OPT=>'FOR ALL COLUMNS SIZE
1', -
                                    DEGREE=>10, -
                                    GRANULARITY=>'DEFAULT', -
                                    CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS(OWNNAME=>'TPCC', -
                                    TABNAME=>'HIST', -
                                    PARTNAME=>NULL, -
                                    ESTIMATE_PERCENT=>1, -
                                    BLOCK_SAMPLE=>TRUE, -
                                    METHOD_OPT=>'FOR ALL COLUMNS SIZE
1', -
                                    DEGREE=>10, -
                                    GRANULARITY=>'DEFAULT', -
                                    CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS(OWNNAME=>'TPCC', -
                                    TABNAME=>'DIST', -
                                    PARTNAME=>NULL, -
                                    ESTIMATE_PERCENT=>1, -
                                    BLOCK_SAMPLE=>TRUE, -
                                    METHOD_OPT=>'FOR ALL COLUMNS SIZE
1', -
                                    DEGREE=>10, -
                                    GRANULARITY=>'DEFAULT', -
                                    CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS(OWNNAME=>'TPCC', -
                                    TABNAME=>'ITEM', -
                                    PARTNAME=>NULL, -
                                    ESTIMATE_PERCENT=>10, -
                                    BLOCK_SAMPLE=>TRUE, -
                                    METHOD_OPT=>'FOR ALL COLUMNS SIZE
1', -
                                    DEGREE=>1, -
                                    GRANULARITY=>'DEFAULT', -
                                    CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS(OWNNAME=>'TPCC', -
                                    TABNAME=>'WARE', -
                                    PARTNAME=>NULL, -
                                    ESTIMATE_PERCENT=>10, -
                                    BLOCK_SAMPLE=>TRUE, -
                                    METHOD_OPT=>'FOR ALL COLUMNS SIZE
1', -
                                    DEGREE=>10, -
                                    GRANULARITY=>'DEFAULT', -
                                    CASCADE=>TRUE);

set echo off;
spool off;

exit sql.sqlcode;
```

createstoredprocs.sh

```
#!/bin/sh
cd $tpcc_genscripts_dir
$tpcc_sqlplus $tpcc_user_pass @${tpcc_genscripts_dir}/createstoredprocs
> junk 2>&1

if test $? -ne 0
then
  exit 1;
else
  exit 0;
fi
```

createstoredprocs.sql

```
spool createstoreprocs.log
@tkvcinin.sql
spool off
exit sql.sqlcode;
```

tkvcinin.sql

```
-- The initnew package for storing variables used in the
-- New Order anonymous block

CREATE OR REPLACE PACKAGE inittpcc
AS
 TYPE intarray IS TABLE OF INTEGER INDEX BY BINARY_INTEGER;
 TYPE distarray IS TABLE OF VARCHAR(24) INDEX BY BINARY_INTEGER;
 nulldate       DATE;
 TYPE rowidarray IS TABLE OF ROWID INDEX BY PLS_INTEGER;
 s_dist                distarray;
```


Copyright © 2004 NEC Corporation

```
 idx1arr               intarray;
 s_remote              intarray;
 dist                   intarray;
 row_id                 rowidarray;
 cust_rowid             rowid;
 dist_name              VARCHAR2(11);
 ware_name              VARCHAR2(11);
 c_num                  PLS_INTEGER;

 PROCEDURE init_no(idxarr intarray);
 PROCEDURE init_del;
 PROCEDURE init_pay;
END inittpcc;
/
show errors;

CREATE OR REPLACE PACKAGE BODY inittpcc AS
  PROCEDURE init_no (idxarr  intarray)
  IS
  BEGIN
        -- initialize null date
   nulldate := TO_DATE('01-01-1811', 'MM-DD-YYYY');
    idx1arr := idxarr;
  END init_no;

  PROCEDURE init_del
  IS
  BEGIN
    FOR i IN 1 .. 10 LOOP
      dist(i) := i;
    END LOOP;
  END init_del;

  PROCEDURE init_pay IS
  BEGIN
    NULL;
  END init_pay;

END inittpcc;
/
show errors
exit
```

createspacestats.sh

```
#!/bin/sh
cd $tpcc_genscripts_dir
$tpcc_sqlplus $tpcc_dba_user_pass @$tpcc_genscripts_dir/createspacestats
> junk 2>&1

if test $? -ne 0
then
  exit 1;
else
  exit 0;
fi
```

createspacestats.sql

```
@space_init
@space_get 400000 50000
@space_rpt
spool off
exit sql.sqlcode;
```

createuser.sh

```
#!/bin/sh

echo Creating user tpcc...
$tpcc_sqlplus $tpcc_dba_user_pass @$tpcc_sql_dir/createuser > junk 2>&1
if test $? -ne 0
then
  exit 1;
else
  exit 0;
fi
```

createuser.sql

```
spool createusertpcc.log;

set echo on;

create user tpcc identified by tpcc;

grant dba to tpcc;

set echo off;
spool off;

exit ;
```

assigntemp.sh

```
spool assigntemp.log;

set echo on;

alter user tpcc temporary tablespace temp_0;

set echo off;
spool off;
```


Copyright © 2004 NEC Corporation

```
exit ;

#!/bin/sh

echo Assigning temporary tablespace to user tpcc...
$tpcc_sqlplus $tpcc_dba_user_pass @$tpcc_sql_dir/assigntemp > junk 2>&1
if test $? -ne 0
then
  exit 1;
else
  exit 0;
fi
```

assigntemp.sql

```
spool assigntemp.log;

set echo on;

alter user tpcc temporary tablespace temp_0;

set echo off;
spool off;

exit ;
```

shutdowndb.sh

```
#!/bin/sh

echo "Shutting down database..."

$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect

spool shutdowndb.log;

set echo on;

alter system switch logfile;
alter system switch logfile;

shutdown immediate;

set echo off;
spool off;

exit
!
```

startupdb.sh

```
#!/bin/sh

echo "Starting up database using $1..."

init_file=${1}.ora

if test $tpcc_np -gt 1 ; then
  init_file=build_init_${tpcc_rac_id}.ora
fi

$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect

spool startdb.log

set echo on

startup pfile=$init_file open

spool off
set echo off
exit sql.sqlcode
!
```

views.sql

```
connect tpcc/tpcc;
set echo on;

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 cust c, ware 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 dist d, ware 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
```


Copyright © 2004 NEC Corporation

```
  from stok s, item i
 where i.i_id = s.s_i_id;

set echo off;
```

ddview.sh

```
#!/bin/sh

$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect

spool ddview.log


REM
REM In an ade/nde view we might need to run standard.sql and dbmsstdx
manually
REM catalog and catproc suppose to take care of it
REM

@$ORACLE_HOME/plsql/admin/standard
@$ORACLE_HOME/rdbms/admin/dbmsstdx

@$ORACLE_HOME/rdbms/admin/catalog
@$ORACLE_HOME/rdbms/admin/catproc

REM
REM In an ade/nde view we might need to run pupbld manually
REM catalog and catproc suppose to take care of it
REM

connect system/manager
REM @$ORACLE_HOME/sqlplus/admin/pupbld


REM
REM Oracle
REM

REM if test $NUMBER_ORACLE_NODE -qt 1
REM then

REM @$ORACLE_HOME/rdbms/admin/catparr

REM fi

spool off
!

#sh $tpcc_scripts/queue.sh
```

createmisc.sh

```
#!/bin/sh

$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect

spool createmisc.log
set echo on;
alter user tpcc temporary tablespace system;
grant execute on dbms_lock to public;
grant execute on dbms_pipe to public;
grant select on v_\$parameter to public;

REM
REM begin plsql_mon.sql
REM

connect tpcc/tpcc;
set echo on;
CREATE OR REPLACE PACKAGE plsql_mon_pack
IS
   PROCEDURE print
   (
      info         VARCHAR2
   );
END;
/
show errors;

CREATE OR REPLACE PACKAGE BODY plsql_mon_pack
IS
   PROCEDURE print
   (
      info         VARCHAR2
   )
   IS
      s            NUMBER;
   BEGIN
      dbms_pipe.pack_message (info);
      s := dbms_pipe.send_message ('plsql_mon');
      IF (s <> 0) THEN
         raise_application_error (-20000, 'Error:' || to_char(s) ||
                                  ' sending on pipe');
      END IF;
   END;
END;
/
show errors;

set echo off;

REM
REM end plsql_mon.sql
REM
```


Copyright © 2004 NEC Corporation

```
REM
REM  begin cre_tab.sql
REM

connect tpcc/tpcc;
set echo on;

drop table temp_o1;
drop table temp_no;
drop table temp_o2;
drop table temp_ol;
drop table tpcc_audit_tab;

create table temp_o1 (
   o_w_id  integer,
   o_d_id  integer,
   o_o_id  integer);

create table temp_no (
   no_w_id integer,
   no_d_id integer,
   no_o_id integer);

create table temp_o2 (
   o_w_id  integer,
   o_d_id  integer,
   o_count integer);

create table temp_ol (
   ol_w_id  integer,
   ol_d_id  integer,
   ol_count integer);

create table tpcc_audit_tab (starttime date);

delete from tpcc_audit_tab;

set echo off;

REM
REM  end cre_tab.sql
REM

REM
REM  begin views.sql
REM

connect tpcc/tpcc;
set echo on;

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 cust c, ware 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 dist d, ware 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 stok s, item i
 where i.i_id = s.s_i_id;

set echo off;

REM
REM  end views.sql
REM


REM
REM  begin dml.sql
REM
connect tpcc/tpcc;
set echo on;

   alter table ware disable table lock;
   alter table dist disable table lock;
   alter table cust disable table lock;
   alter table hist disable table lock;
   alter table item disable table lock;
   alter table stok disable table lock;
   alter table ordr disable table lock;
   alter table nord disable table lock;
   alter table ordl disable table lock;

set echo off;

REM
REM  end dml.sql
REM

REM
REM  begin extent.sql
REM

$SYS_CONNECTION_STRING
```


Copyright © 2004 NEC Corporation

```
@$tpcc_sql_dir/extent

@$tpcc_sql_dir/freeext

exit sql.sqlcode;

!
```

createstats.sh

```
#!/bin/sh

cstat=c_stat
if test $tpcc_np -gt 1 ; then
  cstat=c_stat_rac
fi

$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect

REM
REM create tablespace for statspack user sp begin
REM

spool createstats.log

set echo on
   drop tablespace sp including contents;
   create tablespace sp_0 datafile '${tpcc_disks_location}sp_0' size
$tpcc_statspack_size reuse autoextend on extent management local uniform
size 1M nologging ;
   spool off

REM
REM create tablespace for statspack user sp end
REM

REM
REM  begin now call spcreate to create statspack sp package
REM

$tpcc_internal_connect

define default_tablespace='sp_0'

define temporary_tablespace='temp_0'

@$ORACLE_HOME/rdbms/admin/spcreate
perfstat

REM note that the last thing (after spcreate) is the perfstat password.
REM since we're not worried about security, perfstat will do.

REM
REM tpcc stat table for NT, it is not working so I comment it out
REM shui.lau@oracle.com it is better to use perfmon
REM

@$tpcc_sql_dir/cs_tpcc
@$tpcc_sql_dir/cs_cpu
@$tpcc_sql_dir/cs_os
@$tpcc_sql_dir/cs_proc
@$tpcc_sql_dir/cs_thread

REM
REM tpcc result table for unix and NT
REM

@$tpcc_sql_dir/${cstat}
@$tpcc_sql_dir/pst_c

!
```

createts.sh

```
#created automatically by /home/oracle/tpcc-
kit50000/scripts/buildcreatets.sh Fri Nov 28 18:28:57 JST 2003

# Tablespace ware, ts size 110M (104000K)
# each file 110M (112640K)
# extents 110592K (110592K)
# 1 files

$tpcc_createts ware 1 1       110M 110592K unix 0       0 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for ware failed.  Exiting.
      exit 0
    fi

# Tablespace cust, ts size 1371100M (1404000002K)
# each file 6360M (6512640K)
# extents 282089K (282089K)
# 216 files

$tpcc_createts cust 216 1       6360M 282089K unix 0       1 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for cust failed.  Exiting.
      exit 0
    fi

# Tablespace dist, ts size 1020M (1040000K)
# each file 1030M (1054720K)
# extents 1052672K (1052672K)
# 1 files

$tpcc_createts dist 1 1       1030M 1052672K unix 0       217 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for dist failed.  Exiting.
      exit 0
    fi
```


Copyright © 2004 NEC Corporation

```
# Tablespace hist, ts size 157880M (161662500K)
# each file 2200M (2252800K)
# extents 101329K (101329K)
# 72 files

$tpcc_createts hist 72 1       2200M 101329K unix 0       218 32 4k t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for hist failed.  Exiting.
      exit 0
    fi

# Tablespace stok, ts size 2031250M (2080000000K)
# each file 7060M (7229440K)
# extents 424176K (424176K)
# 288 files

$tpcc_createts stok 288 1       7060M 424176K unix 0       290 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for stok failed.  Exiting.
      exit 0
    fi

# Tablespace item, ts size 20M (15868K)
# each file 30M (30720K)
# extents 28672K (28672K)
# 1 files

$tpcc_createts item 1 1       30M 28672K unix 0       578 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for item failed.  Exiting.
      exit 0
    fi

# Tablespace ordr, ts size 2292910M (2347937500K)
# each file 31860M (32624640K)
# extents 101244K (101244K)
# 72 files

$tpcc_createts ordr 72 1       31860M 101244K unix 0       579 32 16K t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for ordr failed.  Exiting.
      exit 0
    fi

# Tablespace nord, ts size 16930M (17333332K)
# each file 5650M (5785600K)
# extents 1155891K (1155891K)
# 3 files

$tpcc_createts nord 3 1       5650M 1155891K unix 0       651 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for nord failed.  Exiting.
      exit 0
    fi

# Tablespace iware, ts size 70M (62500K)
# each file 70M (71680K)
# extents 33K (33K)
# 1 files

$tpcc_createts iware 1 1       70M 33K unix 0       654 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for iware failed.  Exiting.
      exit 0
    fi

# Tablespace icust1, ts size 35160M (35999999K)
# each file 500M (512000K)
# extents 237K (237K)
# 72 files

$tpcc_createts icust1 36 1       500M 237K unix 0       655 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for icust1 failed.  Exiting.
      exit 0
    fi

# Tablespace icust2, ts size 81130M (83076922K)
# each file 1140M (1167360K)
# extents 541K (541K)
# 72 files

$tpcc_createts icust2 72 1       1140M 541K unix 0       727 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for icust2 failed.  Exiting.
      exit 0
    fi

# Tablespace idist, ts size 250M (250000K)
# each file 250M (256000K)
# extents 118K (118K)
# 1 files

$tpcc_createts idist 1 1       250M 118K unix 0       799 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for idist failed.  Exiting.
      exit 0
    fi

# Tablespace istok, ts size 105470M (108000000K)
# each file 1470M (1505280K)
# extents 698K (698K)
# 72 files

$tpcc_createts istok 36 1       1470M 698K unix 0       800 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for istok failed.  Exiting.
      exit 0
    fi

# Tablespace iitem, ts size 2560K (2560K)
# each file 10M (10240K)
```


Copyright © 2004 NEC Corporation

```
# extents 69K (69K)
# 1 files

$tpcc_createts iitem 1 1       10M 69K unix 0       872 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for iitem failed.  Exiting.
      exit 0
    fi

# Tablespace iordr2, ts size 88320M (90437500K)
# each file 1240M (1269760K)
# extents 589K (589K)
# 72 files

$tpcc_createts iordr2 72 1       1240M 589K unix 0       873 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for iordr2 failed.  Exiting.
      exit 0
    fi

# Tablespace temp, ts size 263680M (270000000K)
# each file 3670M (3758080K)
# extents 196715K (196715K)
# 72 files

$tpcc_createts temp 36 1       3670M 196715K unix 1       945 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for temp failed.  Exiting.
      exit 0
    fi
```

createts.sh.iordr2.bigextent

#created automatically by /home/weshi/1014/nec_kit/tpcc-kit50k/scripts/buildcreatets.sh Thu Jan 8 15:30:36 PST 2004

```
# Tablespace iordr2, ts size 88320M (90437500K)
# each file 1240M (1269760K)
# extents 30M
# 72 files

$tpcc_createts iordr2 72 1       1240M 30M unix 0       873 32 16K t
   if expr $? != 0 > /dev/null; then
     echo Creating tablespace for iordr2 failed.  Exiting.
     exit 0
   fi
```

freeext.sql

```
REM==================================================================+
REM            Copyright (c) 1994  Oracle Corp, Belmont, CA          |
REM                   OPEN SYSTEMS PERFORMANCE GROUP                 |
REM                        All Rights Reserved                       |
REM==================================================================+
REM FILENAME
REM       freeext.sql
REM DESCRIPTION
REM       List all free extents in all the TPCC tablespace
REM
REM  Usage: sqlplus 'sys/change_on_install as sysdba' @freeext
REM==================================================================*/
      set space 2
      set pagesize 2000
      set echo off
      set termout off
      set verify off
      set feedback off
      spool freeextent.rpt
      select substr(e.tablespace_name,1,8) tspace, file_id, block_id,
blocks,
             blocks * t.block_size / 1048576 size_MB
      from dba_free_space e, dba_tablespaces t
      where e.tablespace_name = t.tablespace_name
      order by e.tablespace_name, file_id, block_id;

      select substr(e.tablespace_name,1,8) tspace, sum(blocks) tot_blk,
             sum(blocks) * t.block_size / 1048576 size_MB
      from dba_free_space e, dba_tablespaces t
      where e.tablespace_name = t.tablespace_name
      group by e.tablespace_name, t.block_size
      order by e.tablespace_name;
```

plsql_mon.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       plsql_mon.sql
rem  DESCRIPTION
rem       SQL script to create a stored package for PL/SQL stored
rem       procedures to dump messages.
rem ===================================================================
rem
rem Usage:   sqlplus tpcc/tpcc @plsql_mon
rem
```


Copyright © 2004 NEC Corporation

```
connect tpcc/tpcc;
set echo on;
CREATE OR REPLACE PACKAGE plsql_mon_pack
IS
   PROCEDURE print
   (
      info         VARCHAR2
   );
END;
/
show errors;

CREATE OR REPLACE PACKAGE BODY plsql_mon_pack
IS
   PROCEDURE print
   (
      info         VARCHAR2
   )
   IS
      s            NUMBER;
   BEGIN
      dbms_pipe.pack_message (info);
      s := dbms_pipe.send_message ('plsql_mon');
      IF (s <> 0) THEN
         raise_application_error (-20000, 'Error:' || to_char(s) ||
                                  ' sending on pipe');
      END IF;
   END;
END;
/
show errors;

set echo off;
```

cre_tab.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       cre_tab.sql
rem  DESCRIPTION
rem       Create temporary tables for consistency tests.
rem ================================================================
rem
rem Usage:   sqlplus tpcc/tpcc @cre_tab
rem

connect tpcc/tpcc;
set echo on;

drop table temp_o1;
drop table temp_no;
drop table temp_o2;
drop table temp_ol;
drop table tpcc_audit_tab;

create table temp_o1 (
   o_w_id  integer,
   o_d_id  integer,
   o_o_id  integer);

create table temp_no (
   no_w_id integer,
   no_d_id integer,
   no_o_id integer);

create table temp_o2 (
   o_w_id  integer,
   o_d_id  integer,
   o_count integer);

create table temp_ol (
   ol_w_id  integer,
   ol_d_id  integer,
   ol_count integer);

create table tpcc_audit_tab (starttime date);

delete from tpcc_audit_tab;

set echo off;
```

cs_cpu.sql

```
rem
rem
rem================================================================+
rem        Copyright (c) 1997  Oracle Corp, Redwood Shores, CA       |
rem                        All Rights Reserved                       |
rem================================================================+
rem FILENAME
rem     cs_cpu.sql
rem DESCRIPTION
rem     Create Table for CPU Specific Process Stat
rem===================================================================
rem usage: sqlplus tpcc/tpcc @cs_cpu.sql

connect tpcc/tpcc
set echo on

DROP TABLE pre_cpu_stats;
DROP TABLE post_cpu_stats;
DROP TABLE cpu_stats;
```


Copyright © 2004 NEC Corporation

```
rem
rem  CPU statistics.
rem

        CREATE TABLE cpu_stats
        (
            runname         VARCHAR2(20),
                      cpu_id          NUMBER,
            dpc_cpu         NUMBER,
            interrupt_cpu   NUMBER,
            priv_cpu        NUMBER,
            processor_cpu   NUMBER,
            user_cpu        NUMBER,
            interrupt_rate   NUMBER
        );

rem
rem  Save Begining CPU Stat Values
rem

        CREATE TABLE pre_cpu_stats
        (
            runname         VARCHAR2(20),
                      cpu_id          NUMBER,
            dpc_cpu         NUMBER,
            interrupt_cpu   NUMBER,
            priv_cpu        NUMBER,
            processor_cpu   NUMBER,
            user_cpu        NUMBER,
            interrupt_rate   NUMBER
        );


rem
rem  Save Ending CPU Stat Values
rem

        CREATE TABLE post_cpu_stats
        (
            runname         VARCHAR2(20),
                      cpu_id          NUMBER,
            dpc_cpu         NUMBER,
            interrupt_cpu   NUMBER,
            priv_cpu        NUMBER,
            processor_cpu   NUMBER,
            user_cpu        NUMBER,
            interrupt_rate   NUMBER
        );
commit;
set echo off;
```

cs_os.sql

```
rem
rem
rem        Copyright (c) 1997  Oracle Corp, Redwood Shores, CA       |
rem                        All Rights Reserved                       |
rem
rem FILENAME
rem     cs_os.sql
rem DESCRIPTION
rem     Create Table for OS Specific Process Stat
rem
rem usage: sqlplus tpcc/tpcc @cs_os.sql

connect tpcc/tpcc
set echo on

DROP TABLE pre_os_stats;
DROP TABLE post_os_stats;
DROP TABLE os_stats;

rem
rem  OS statistics.
rem

        CREATE TABLE os_stats
        (
            runname         VARCHAR2(20),
            time            NUMBER,
            syscall         NUMBER,
            intr            NUMBER,
            cswitch         NUMBER,
            freads          NUMBER,
            fwrites         NUMBER,
            fcontrolops     NUMBER,
            priv_cpu        NUMBER,
            user_cpu        NUMBER,
            processor_cpu   NUMBER,
                      interrupt_cpu   NUMBER
        );

rem
rem  Save Begining OS Stat Values
rem

        CREATE TABLE pre_os_stats
        (
            runname         VARCHAR2(20),
            time            NUMBER,
            syscall         NUMBER,
            intr            NUMBER,
            cswitch         NUMBER,
            freads          NUMBER,
            fwrites         NUMBER,
            fcontrolops     NUMBER,
            priv_cpu        NUMBER,
            user_cpu        NUMBER,
            processor_cpu   NUMBER,
```


Copyright © 2004 NEC Corporation

```
                    interrupt_cpu   NUMBER
        );


rem
rem  Save Ending OS Stat Values
rem

        CREATE TABLE post_os_stats
        (
            runname         VARCHAR2(20),
            time            NUMBER,
            syscall         NUMBER,
            intr            NUMBER,
            cswitch         NUMBER,
            freads          NUMBER,
            fwrites         NUMBER,
            fcontrolops     NUMBER,
            priv_cpu        NUMBER,
            user_cpu        NUMBER,
            processor_cpu   NUMBER,
                    interrupt_cpu   NUMBER
        );
commit;
set echo off;
```

cs_thread.sql

```
rem
rem
rem================================================================+
rem        Copyright (c) 1997  Oracle Corp, Redwood Shores, CA       |
rem                        All Rights Reserved                      |
rem================================================================+
rem FILENAME
rem    cs_thread.sql
rem DESCRIPTION
rem    Create Table for thread statistics
rem===================================================================
rem Usage: sqlplus tpcc/tpcc @cs_thread.sql

connect tpcc/tpcc
set echo on

DROP TABLE thread_stats;
DROP TABLE pre_thread_stats;
DROP TABLE post_thread_stats;

rem
rem  Resource usage for a thread.
rem

        CREATE TABLE thread_stats
        (
            runname       VARCHAR2(20),
               thread_id       VARCHAR2(10),
            user_cpu      NUMBER,
            priv_cpu      NUMBER,
            processor_cpu NUMBER,
            ctxswitch     NUMBER
        );


rem
rem  Save Begining Resource Values for a thread.
rem

        CREATE TABLE pre_thread_stats
        (
            runname       VARCHAR2(20),
               thread_id       VARCHAR2(10),
            user_cpu      NUMBER,
            priv_cpu      NUMBER,
            processor_cpu NUMBER,
            ctxswitch     NUMBER
        );


rem
rem  Save Ending Resource Values for a thread.
rem

        CREATE TABLE post_thread_stats
        (
            runname       VARCHAR2(20),
               thread_id       VARCHAR2(10),
            user_cpu      NUMBER,
            priv_cpu      NUMBER,
            processor_cpu NUMBER,
            ctxswitch     NUMBER
        );
commit;
set echo off
```

cs_proc.sql

```
rem
rem
rem================================================================+
rem        Copyright (c) 1997  Oracle Corp, Redwood Shores, CA       |
rem                        All Rights Reserved                      |
rem================================================================+
rem FILENAME
rem    cs_proc.sql
rem DESCRIPTION
rem    Create Table for OS Specific Process Stats
rem===================================================================
```


Copyright © 2004 NEC Corporation

```
rem Usage: sqlplus tpcc/tpcc @cs_proc.sql

connect tpcc/tpcc
set echo on

DROP TABLE process_stats;
DROP TABLE pre_process_stats;
DROP TABLE post_process_stats;

rem
rem  Resource usage for a process.
rem

        CREATE TABLE process_stats
        (
            runname       VARCHAR2(20),
            user_cpu      NUMBER,
            priv_cpu      NUMBER,
            processor_cpu NUMBER,
            pagefaults    NUMBER
        );



rem
rem  Save Begining Resource Values for a process.
rem

        CREATE TABLE pre_process_stats
        (
            runname       VARCHAR2(20),
            user_cpu      NUMBER,
            priv_cpu      NUMBER,
            processor_cpu NUMBER,
            pagefaults    NUMBER
        );


rem
rem  Save Ending Resource Values for a process.
rem

        CREATE TABLE post_process_stats
        (
            runname       VARCHAR2(20),
            user_cpu      NUMBER,
            priv_cpu      NUMBER,
            processor_cpu NUMBER,
            pagefaults    NUMBER
        );
commit;
set echo off
```

cs_tpcc.sql

```
rem
rem
rem==================================================================+
rem        Copyright (c) 1997  Oracle Corp, Redwood Shores, CA       |
rem                         All Rights Reserved                      |
rem==================================================================+
rem FILENAME
rem      cs_tpcc.sq
rem DESCRIPTION
rem      Create tables for saving TPC-C results.
rem===================================================================
rem  Usage: sqlplus user/password @cs_tpcc.sql
rem spool cs_tpcc.log

connect tpcc/tpcc;
set echo on

DROP TABLE tpcc_run_desc;
DROP TABLE tpcc_run_int;
DROP TABLE bench_run_int;
DROP TABLE tpcc_back_res;
DROP TABLE tpcc_user_res;
DROP TABLE bench_user_res;
DROP TABLE tpcc_tpm;
DROP TABLE tpcc_new_res;
DROP TABLE bench_new_res;
DROP TABLE tpcc_pay_res;
DROP TABLE bench_pay_res;
DROP TABLE tpcc_ord_res;
DROP TABLE bench_ord_res;
DROP TABLE tpcc_del_res;
DROP TABLE bench_del_res;
DROP TABLE tpcc_sto_res;
DROP TABLE bench_sto_res;

rem
rem  description of a run
rem
        CREATE TABLE tpcc_run_desc
        (
            run_name        VARCHAR2(20),
            rundate         DATE,
            time            NUMBER,
            rampup          NUMBER,
            rampdown        NUMBER,
            warehouses      NUMBER,
            customers       NUMBER,
            users           NUMBER,
            driver          VARCHAR2(40),
            commnt          VARCHAR2(80)
        );

rem
rem  throughput of new order transactions
```


Copyright © 2004 NEC Corporation

```
rem
       CREATE TABLE tpcc_run_int
       (
           run_name        VARCHAR2(20),
           interval        NUMBER,
           interval_count  NUMBER,
           response_time   NUMBER,
           think_time      NUMBER
       );

rem
rem  throughput of new order transactions
rem
       CREATE TABLE bench_run_int
       (
           run_name        VARCHAR2(20),
           proc_no         NUMBER,
           interval        NUMBER,
           interval_count  NUMBER,
           response_time   NUMBER,
           think_time      NUMBER
       );

rem
rem  Results from delivery servers
rem
       CREATE TABLE tpcc_back_res
       (
           run_name        VARCHAR2(20),
           in_timing_int   NUMBER,
           fast            NUMBER,
           resp_time       NUMBER,
           retries         NUMBER
       );

rem
rem  Aggregate results for all generators.
rem  These results are from the measurement interval only.
rem  These results are used to calculate the TPS rate over
rem  the measurement interval.
rem
       CREATE TABLE tpcc_user_res
       (
           run_name        VARCHAR2(20),
           no_men          NUMBER,
           fast_men        NUMBER,
           in_flight_men   NUMBER,
           retry_men       NUMBER,
           min_time_men    NUMBER,
           max_time_men    NUMBER,
           sum_time_men    NUMBER,
           ninety_per_men  NUMBER,
           think_min_men   NUMBER,
           think_max_men   NUMBER,
           think_sum_men   NUMBER,
           key_min_men     NUMBER,
           key_max_men     NUMBER,
           key_sum_men     NUMBER,
           no_new          NUMBER,
           fast_new        NUMBER,
           in_flight_new   NUMBER,
           retry_new       NUMBER,
           min_time_new    NUMBER,
           max_time_new    NUMBER,
           sum_time_new    NUMBER,
           ninety_per_new  NUMBER,
           think_min_new   NUMBER,
           think_max_new   NUMBER,
           think_sum_new   NUMBER,
           key_min_new     NUMBER,
           key_max_new     NUMBER,
           key_sum_new     NUMBER,
           remote_new      NUMBER,
           rollback_new    NUMBER,
           sum_ol_new      NUMBER,
           remote_ol_new   NUMBER,
           allrollback_new NUMBER,
           no_pay          NUMBER,
           fast_pay        NUMBER,
           in_flight_pay   NUMBER,
           retry_pay       NUMBER,
           min_time_pay    NUMBER,
           max_time_pay    NUMBER,
           sum_time_pay    NUMBER,
           ninety_per_pay  NUMBER,
           think_min_pay   NUMBER,
           think_max_pay   NUMBER,
           think_sum_pay   NUMBER,
           key_min_pay     NUMBER,
           key_max_pay     NUMBER,
           key_sum_pay     NUMBER,
           remote_pay      NUMBER,
           bylast_pay      NUMBER,
           no_ord          NUMBER,
           fast_ord        NUMBER,
           in_flight_ord   NUMBER,
           retry_ord       NUMBER,
           min_time_ord    NUMBER,
           max_time_ord    NUMBER,
           sum_time_ord    NUMBER,
           ninety_per_ord  NUMBER,
           think_min_ord   NUMBER,
           think_max_ord   NUMBER,
           think_sum_ord   NUMBER,
           key_min_ord     NUMBER,
           key_max_ord     NUMBER,
           key_sum_ord     NUMBER,
           bylast_ord      NUMBER,
           no_del          NUMBER,
           fast_del        NUMBER,
           in_flight_del   NUMBER,
           retry_del       NUMBER,
```


Copyright © 2004 NEC Corporation

```
        min_time_del    NUMBER,
        max_time_del    NUMBER,
        sum_time_del    NUMBER,
        ninety_per_del  NUMBER,
        think_min_del   NUMBER,
        think_max_del   NUMBER,
        think_sum_del   NUMBER,
        key_min_del     NUMBER,
        key_max_del     NUMBER,
        key_sum_del     NUMBER,
        no_sto          NUMBER,
        fast_sto        NUMBER,
        in_flight_sto   NUMBER,
        retry_sto       NUMBER,
        min_time_sto    NUMBER,
        max_time_sto    NUMBER,
        sum_time_sto    NUMBER,
        ninety_per_sto  NUMBER,
        think_min_sto   NUMBER,
        think_max_sto   NUMBER,
        think_sum_sto   NUMBER,
        key_min_sto     NUMBER,
        key_max_sto     NUMBER,
        key_sum_sto     NUMBER,
        cpu_time        NUMBER,
        deadlocks       NUMBER
    );

rem
rem  Results from individual generators.
rem  These results are from the measurement interval only.
rem  These results are used to calculate the TPS rate over
rem  the measurement interval.
rem
    CREATE TABLE bench_user_res
    (
        run_name        VARCHAR2(20),
        audit_str       VARCHAR2(10),
        proc_no         NUMBER,
        hid             NUMBER,
        no_men          NUMBER,
        fast_men        NUMBER,
        in_flight_men   NUMBER,
        retry_men       NUMBER,
        min_time_men    NUMBER,
        max_time_men    NUMBER,
        sum_time_men    NUMBER,
        ninety_per_men  NUMBER,
        think_min_men   NUMBER,
        think_max_men   NUMBER,
        think_sum_men   NUMBER,
        key_min_men     NUMBER,
        key_max_men     NUMBER,
        key_sum_men     NUMBER,
        no_new          NUMBER,
        fast_new        NUMBER,
        in_flight_new   NUMBER,
        retry_new       NUMBER,
        min_time_new    NUMBER,
        max_time_new    NUMBER,
        sum_time_new    NUMBER,
        ninety_per_new  NUMBER,
        think_min_new   NUMBER,
        think_max_new   NUMBER,
        think_sum_new   NUMBER,
        key_min_new     NUMBER,
        key_max_new     NUMBER,
        key_sum_new     NUMBER,
        remote_new      NUMBER,
        rollback_new    NUMBER,
        sum_ol_new      NUMBER,
        remote_ol_new   NUMBER,
        allrollback_new NUMBER,
        no_pay          NUMBER,
        fast_pay        NUMBER,
        in_flight_pay   NUMBER,
        retry_pay       NUMBER,
        min_time_pay    NUMBER,
        max_time_pay    NUMBER,
        sum_time_pay    NUMBER,
        ninety_per_pay  NUMBER,
        think_min_pay   NUMBER,
        think_max_pay   NUMBER,
        think_sum_pay   NUMBER,
        key_min_pay     NUMBER,
        key_max_pay     NUMBER,
        key_sum_pay     NUMBER,
        remote_pay      NUMBER,
        bylast_pay      NUMBER,
        no_ord          NUMBER,
        fast_ord        NUMBER,
        in_flight_ord   NUMBER,
        retry_ord       NUMBER,
        min_time_ord    NUMBER,
        max_time_ord    NUMBER,
        sum_time_ord    NUMBER,
        ninety_per_ord  NUMBER,
        think_min_ord   NUMBER,
        think_max_ord   NUMBER,
        think_sum_ord   NUMBER,
        key_min_ord     NUMBER,
        key_max_ord     NUMBER,
        key_sum_ord     NUMBER,
        bylast_ord      NUMBER,
        no_del          NUMBER,
        fast_del        NUMBER,
        in_flight_del   NUMBER,
        retry_del       NUMBER,
        min_time_del    NUMBER,
        max_time_del    NUMBER,
        sum_time_del    NUMBER,
        ninety_per_del  NUMBER,
```


Copyright © 2004 NEC Corporation

```
            think_min_del   NUMBER,
            think_max_del   NUMBER,
            think_sum_del   NUMBER,
            key_min_del     NUMBER,
            key_max_del     NUMBER,
            key_sum_del     NUMBER,
            no_sto          NUMBER,
            fast_sto        NUMBER,
            in_flight_sto   NUMBER,
            retry_sto       NUMBER,
            min_time_sto    NUMBER,
            max_time_sto    NUMBER,
            sum_time_sto    NUMBER,
            ninety_per_sto  NUMBER,
            think_min_sto   NUMBER,
            think_max_sto   NUMBER,
            think_sum_sto   NUMBER,
            key_min_sto     NUMBER,
            key_max_sto     NUMBER,
            key_sum_sto     NUMBER,
            cpu_time        NUMBER,
            deadlocks       NUMBER
        );

rem
rem  Aggregate results for generators on each host.
rem  These results are from the measurement interval only.
rem  These results are used to calculate the TPM rate over
rem  the measurement interval.
rem
        CREATE TABLE tpcc_tpm
        (
            run_name        VARCHAR2(20),
            hid             NUMBER,
            no_new          NUMBER
        );

rem
rem  Aggregate results for new order transactions.
rem  These results are from the measurement interval only.
rem
        CREATE TABLE tpcc_new_res
        (
            run_name        VARCHAR2(20),
            rep1            NUMBER,
            rep2            NUMBER,
            rep3            NUMBER,
            rep4            NUMBER,
            rep5            NUMBER,
            rep6            NUMBER,
            rep7            NUMBER,
            rep8            NUMBER,
            rep9            NUMBER,
            rep10           NUMBER,
            rep11           NUMBER,
            rep12           NUMBER,
            rep13           NUMBER,
            rep14           NUMBER,
            rep15           NUMBER,
            rep16           NUMBER,
            rep17           NUMBER,
            rep18           NUMBER,
            rep19           NUMBER,
            rep20           NUMBER,
            rep21           NUMBER,
            rep22           NUMBER,
            rep23           NUMBER,
            rep24           NUMBER,
            rep25           NUMBER,
            rep26           NUMBER,
            rep27           NUMBER,
            rep28           NUMBER,
            rep29           NUMBER,
            rep30           NUMBER,
            rep31           NUMBER,
            rep32           NUMBER,
            rep33           NUMBER,
            rep34           NUMBER,
            rep35           NUMBER,
            rep36           NUMBER,
            rep37           NUMBER,
            rep38           NUMBER,
            rep39           NUMBER,
            rep40           NUMBER,
            rep41           NUMBER,
            rep42           NUMBER,
            rep43           NUMBER,
            rep44           NUMBER,
            rep45           NUMBER,
            rep46           NUMBER,
            rep47           NUMBER,
            rep48           NUMBER,
            rep49           NUMBER,
            rep50           NUMBER,
            rep51           NUMBER,
            rep52           NUMBER,
            rep53           NUMBER,
            rep54           NUMBER,
            rep55           NUMBER,
            rep56           NUMBER,
            rep57           NUMBER,
            rep58           NUMBER,
            rep59           NUMBER,
            rep60           NUMBER,
            rep61           NUMBER,
            rep62           NUMBER,
            rep63           NUMBER,
            rep64           NUMBER,
            rep65           NUMBER,
            rep66           NUMBER,
            rep67           NUMBER,
            rep68           NUMBER,
```


Copyright © 2004 NEC Corporation

```
rep69           NUMBER,
rep70           NUMBER,
rep71           NUMBER,
rep72           NUMBER,
rep73           NUMBER,
rep74           NUMBER,
rep75           NUMBER,
rep76           NUMBER,
rep77           NUMBER,
rep78           NUMBER,
rep79           NUMBER,
rep80           NUMBER,
rep81           NUMBER,
rep82           NUMBER,
rep83           NUMBER,
rep84           NUMBER,
rep85           NUMBER,
rep86           NUMBER,
rep87           NUMBER,
rep88           NUMBER,
rep89           NUMBER,
rep90           NUMBER,
rep91           NUMBER,
rep92           NUMBER,
rep93           NUMBER,
rep94           NUMBER,
rep95           NUMBER,
rep96           NUMBER,
rep97           NUMBER,
rep98           NUMBER,
rep99           NUMBER,
rep100          NUMBER,
thk1            NUMBER,
thk2            NUMBER,
thk3            NUMBER,
thk4            NUMBER,
thk5            NUMBER,
thk6            NUMBER,
thk7            NUMBER,
thk8            NUMBER,
thk9            NUMBER,
thk10           NUMBER,
thk11           NUMBER,
thk12           NUMBER,
thk13           NUMBER,
thk14           NUMBER,
thk15           NUMBER,
thk16           NUMBER,
thk17           NUMBER,
thk18           NUMBER,
thk19           NUMBER,
thk20           NUMBER,
thk21           NUMBER,
thk22           NUMBER,
thk23           NUMBER,
thk24           NUMBER,
thk25           NUMBER,
key1            NUMBER,
key2            NUMBER,
key3            NUMBER,
key4            NUMBER,
key5            NUMBER,
key6            NUMBER,
key7            NUMBER,
key8            NUMBER,
key9            NUMBER,
key10           NUMBER
        );

rem
rem  Results for new order transactions.
rem  These results are from the measurement interval only.
rem
        CREATE TABLE bench_new_res
        (
            run_name        VARCHAR2(20),
            audit_str       VARCHAR2(10),
            proc_no         NUMBER,
            rep1            NUMBER,
            rep2            NUMBER,
            rep3            NUMBER,
            rep4            NUMBER,
            rep5            NUMBER,
            rep6            NUMBER,
            rep7            NUMBER,
            rep8            NUMBER,
            rep9            NUMBER,
            rep10           NUMBER,
            rep11           NUMBER,
            rep12           NUMBER,
            rep13           NUMBER,
            rep14           NUMBER,
            rep15           NUMBER,
            rep16           NUMBER,
            rep17           NUMBER,
            rep18           NUMBER,
            rep19           NUMBER,
            rep20           NUMBER,
            rep21           NUMBER,
            rep22           NUMBER,
            rep23           NUMBER,
            rep24           NUMBER,
            rep25           NUMBER,
            rep26           NUMBER,
            rep27           NUMBER,
            rep28           NUMBER,
            rep29           NUMBER,
            rep30           NUMBER,
            rep31           NUMBER,
            rep32           NUMBER,
            rep33           NUMBER,
            rep34           NUMBER,
```


Copyright © 2004 NEC Corporation

```
rep35           NUMBER,
rep36           NUMBER,
rep37           NUMBER,
rep38           NUMBER,
rep39           NUMBER,
rep40           NUMBER,
rep41           NUMBER,
rep42           NUMBER,
rep43           NUMBER,
rep44           NUMBER,
rep45           NUMBER,
rep46           NUMBER,
rep47           NUMBER,
rep48           NUMBER,
rep49           NUMBER,
rep50           NUMBER,
rep51           NUMBER,
rep52           NUMBER,
rep53           NUMBER,
rep54           NUMBER,
rep55           NUMBER,
rep56           NUMBER,
rep57           NUMBER,
rep58           NUMBER,
rep59           NUMBER,
rep60           NUMBER,
rep61           NUMBER,
rep62           NUMBER,
rep63           NUMBER,
rep64           NUMBER,
rep65           NUMBER,
rep66           NUMBER,
rep67           NUMBER,
rep68           NUMBER,
rep69           NUMBER,
rep70           NUMBER,
rep71           NUMBER,
rep72           NUMBER,
rep73           NUMBER,
rep74           NUMBER,
rep75           NUMBER,
rep76           NUMBER,
rep77           NUMBER,
rep78           NUMBER,
rep79           NUMBER,
rep80           NUMBER,
rep81           NUMBER,
rep82           NUMBER,
rep83           NUMBER,
rep84           NUMBER,
rep85           NUMBER,
rep86           NUMBER,
rep87           NUMBER,
rep88           NUMBER,
rep89           NUMBER,
rep90           NUMBER,
rep91           NUMBER,
rep92           NUMBER,
rep93           NUMBER,
rep94           NUMBER,
rep95           NUMBER,
rep96           NUMBER,
rep97           NUMBER,
rep98           NUMBER,
rep99           NUMBER,
rep100          NUMBER,
thk1            NUMBER,
thk2            NUMBER,
thk3            NUMBER,
thk4            NUMBER,
thk5            NUMBER,
thk6            NUMBER,
thk7            NUMBER,
thk8            NUMBER,
thk9            NUMBER,
thk10           NUMBER,
thk11           NUMBER,
thk12           NUMBER,
thk13           NUMBER,
thk14           NUMBER,
thk15           NUMBER,
thk16           NUMBER,
thk17           NUMBER,
thk18           NUMBER,
thk19           NUMBER,
thk20           NUMBER,
thk21           NUMBER,
thk22           NUMBER,
thk23           NUMBER,
thk24           NUMBER,
thk25           NUMBER,
key1            NUMBER,
key2            NUMBER,
key3            NUMBER,
key4            NUMBER,
key5            NUMBER,
key6            NUMBER,
key7            NUMBER,
key8            NUMBER,
key9            NUMBER,
key10           NUMBER
        );

rem
rem  Aggregate results for payment transactions.
rem  These results are from the measurement interval only.
rem
        CREATE TABLE tpcc_pay_res
        (
            run_name        VARCHAR2(20),
            rep1            NUMBER,
            rep2            NUMBER,
```


Copyright © 2004 NEC Corporation

```
rep3            NUMBER,
rep4            NUMBER,
rep5            NUMBER,
rep6            NUMBER,
rep7            NUMBER,
rep8            NUMBER,
rep9            NUMBER,
rep10           NUMBER,
rep11           NUMBER,
rep12           NUMBER,
rep13           NUMBER,
rep14           NUMBER,
rep15           NUMBER,
rep16           NUMBER,
rep17           NUMBER,
rep18           NUMBER,
rep19           NUMBER,
rep20           NUMBER,
rep21           NUMBER,
rep22           NUMBER,
rep23           NUMBER,
rep24           NUMBER,
rep25           NUMBER,
rep26           NUMBER,
rep27           NUMBER,
rep28           NUMBER,
rep29           NUMBER,
rep30           NUMBER,
rep31           NUMBER,
rep32           NUMBER,
rep33           NUMBER,
rep34           NUMBER,
rep35           NUMBER,
rep36           NUMBER,
rep37           NUMBER,
rep38           NUMBER,
rep39           NUMBER,
rep40           NUMBER,
rep41           NUMBER,
rep42           NUMBER,
rep43           NUMBER,
rep44           NUMBER,
rep45           NUMBER,
rep46           NUMBER,
rep47           NUMBER,
rep48           NUMBER,
rep49           NUMBER,
rep50           NUMBER,
rep51           NUMBER,
rep52           NUMBER,
rep53           NUMBER,
rep54           NUMBER,
rep55           NUMBER,
rep56           NUMBER,
rep57           NUMBER,
rep58           NUMBER,
rep59           NUMBER,
rep60           NUMBER,
rep61           NUMBER,
rep62           NUMBER,
rep63           NUMBER,
rep64           NUMBER,
rep65           NUMBER,
rep66           NUMBER,
rep67           NUMBER,
rep68           NUMBER,
rep69           NUMBER,
rep70           NUMBER,
rep71           NUMBER,
rep72           NUMBER,
rep73           NUMBER,
rep74           NUMBER,
rep75           NUMBER,
rep76           NUMBER,
rep77           NUMBER,
rep78           NUMBER,
rep79           NUMBER,
rep80           NUMBER,
rep81           NUMBER,
rep82           NUMBER,
rep83           NUMBER,
rep84           NUMBER,
rep85           NUMBER,
rep86           NUMBER,
rep87           NUMBER,
rep88           NUMBER,
rep89           NUMBER,
rep90           NUMBER,
rep91           NUMBER,
rep92           NUMBER,
rep93           NUMBER,
rep94           NUMBER,
rep95           NUMBER,
rep96           NUMBER,
rep97           NUMBER,
rep98           NUMBER,
rep99           NUMBER,
rep100          NUMBER,
thk1            NUMBER,
thk2            NUMBER,
thk3            NUMBER,
thk4            NUMBER,
thk5            NUMBER,
thk6            NUMBER,
thk7            NUMBER,
thk8            NUMBER,
thk9            NUMBER,
thk10           NUMBER,
thk11           NUMBER,
thk12           NUMBER,
thk13           NUMBER,
thk14           NUMBER,
```


Copyright © 2004 NEC Corporation

```
            thk15           NUMBER,
            thk16           NUMBER,
            thk17           NUMBER,
            thk18           NUMBER,
            thk19           NUMBER,
            thk20           NUMBER,
            thk21           NUMBER,
            thk22           NUMBER,
            thk23           NUMBER,
            thk24           NUMBER,
            thk25           NUMBER,
            key1            NUMBER,
            key2            NUMBER,
            key3            NUMBER,
            key4            NUMBER,
            key5            NUMBER,
            key6            NUMBER,
            key7            NUMBER,
            key8            NUMBER,
            key9            NUMBER,
            key10           NUMBER
        );

rem
rem  Results for payment transactions.
rem  These results are from the measurement interval only.
rem
        CREATE TABLE bench_pay_res
        (
            run_name        VARCHAR2(20),
            audit_str       VARCHAR2(10),
            proc_no         NUMBER,
            rep1            NUMBER,
            rep2            NUMBER,
            rep3            NUMBER,
            rep4            NUMBER,
            rep5            NUMBER,
            rep6            NUMBER,
            rep7            NUMBER,
            rep8            NUMBER,
            rep9            NUMBER,
            rep10           NUMBER,
            rep11           NUMBER,
            rep12           NUMBER,
            rep13           NUMBER,
            rep14           NUMBER,
            rep15           NUMBER,
            rep16           NUMBER,
            rep17           NUMBER,
            rep18           NUMBER,
            rep19           NUMBER,
            rep20           NUMBER,
            rep21           NUMBER,
            rep22           NUMBER,
            rep23           NUMBER,
            rep24           NUMBER,
            rep25           NUMBER,
            rep26           NUMBER,
            rep27           NUMBER,
            rep28           NUMBER,
            rep29           NUMBER,
            rep30           NUMBER,
            rep31           NUMBER,
            rep32           NUMBER,
            rep33           NUMBER,
            rep34           NUMBER,
            rep35           NUMBER,
            rep36           NUMBER,
            rep37           NUMBER,
            rep38           NUMBER,
            rep39           NUMBER,
            rep40           NUMBER,
            rep41           NUMBER,
            rep42           NUMBER,
            rep43           NUMBER,
            rep44           NUMBER,
            rep45           NUMBER,
            rep46           NUMBER,
            rep47           NUMBER,
            rep48           NUMBER,
            rep49           NUMBER,
            rep50           NUMBER,
            rep51           NUMBER,
            rep52           NUMBER,
            rep53           NUMBER,
            rep54           NUMBER,
            rep55           NUMBER,
            rep56           NUMBER,
            rep57           NUMBER,
            rep58           NUMBER,
            rep59           NUMBER,
            rep60           NUMBER,
            rep61           NUMBER,
            rep62           NUMBER,
            rep63           NUMBER,
            rep64           NUMBER,
            rep65           NUMBER,
            rep66           NUMBER,
            rep67           NUMBER,
            rep68           NUMBER,
            rep69           NUMBER,
            rep70           NUMBER,
            rep71           NUMBER,
            rep72           NUMBER,
            rep73           NUMBER,
            rep74           NUMBER,
            rep75           NUMBER,
            rep76           NUMBER,
            rep77           NUMBER,
            rep78           NUMBER,
            rep79           NUMBER,
            rep80           NUMBER,
```


Copyright © 2004 NEC Corporation

```
    rep81           NUMBER,
    rep82           NUMBER,
    rep83           NUMBER,
    rep84           NUMBER,
    rep85           NUMBER,
    rep86           NUMBER,
    rep87           NUMBER,
    rep88           NUMBER,
    rep89           NUMBER,
    rep90           NUMBER,
    rep91           NUMBER,
    rep92           NUMBER,
    rep93           NUMBER,
    rep94           NUMBER,
    rep95           NUMBER,
    rep96           NUMBER,
    rep97           NUMBER,
    rep98           NUMBER,
    rep99           NUMBER,
    rep100          NUMBER,
    thk1            NUMBER,
    thk2            NUMBER,
    thk3            NUMBER,
    thk4            NUMBER,
    thk5            NUMBER,
    thk6            NUMBER,
    thk7            NUMBER,
    thk8            NUMBER,
    thk9            NUMBER,
    thk10           NUMBER,
    thk11           NUMBER,
    thk12           NUMBER,
    thk13           NUMBER,
    thk14           NUMBER,
    thk15           NUMBER,
    thk16           NUMBER,
    thk17           NUMBER,
    thk18           NUMBER,
    thk19           NUMBER,
    thk20           NUMBER,
    thk21           NUMBER,
    thk22           NUMBER,
    thk23           NUMBER,
    thk24           NUMBER,
    thk25           NUMBER,
    key1            NUMBER,
    key2            NUMBER,
    key3            NUMBER,
    key4            NUMBER,
    key5            NUMBER,
    key6            NUMBER,
    key7            NUMBER,
    key8            NUMBER,
    key9            NUMBER,
    key10           NUMBER
);
```

```
rem
rem  Aggregate results for order status transactions.
rem  These results are from the measurement interval only.
rem
       CREATE TABLE tpcc_ord_res
       (
           run_name        VARCHAR2(20),
           rep1            NUMBER,
           rep2            NUMBER,
           rep3            NUMBER,
           rep4            NUMBER,
           rep5            NUMBER,
           rep6            NUMBER,
           rep7            NUMBER,
           rep8            NUMBER,
           rep9            NUMBER,
           rep10           NUMBER,
           rep11           NUMBER,
           rep12           NUMBER,
           rep13           NUMBER,
           rep14           NUMBER,
           rep15           NUMBER,
           rep16           NUMBER,
           rep17           NUMBER,
           rep18           NUMBER,
           rep19           NUMBER,
           rep20           NUMBER,
           rep21           NUMBER,
           rep22           NUMBER,
           rep23           NUMBER,
           rep24           NUMBER,
           rep25           NUMBER,
           rep26           NUMBER,
           rep27           NUMBER,
           rep28           NUMBER,
           rep29           NUMBER,
           rep30           NUMBER,
           rep31           NUMBER,
           rep32           NUMBER,
           rep33           NUMBER,
           rep34           NUMBER,
           rep35           NUMBER,
           rep36           NUMBER,
           rep37           NUMBER,
           rep38           NUMBER,
           rep39           NUMBER,
           rep40           NUMBER,
           rep41           NUMBER,
           rep42           NUMBER,
           rep43           NUMBER,
           rep44           NUMBER,
           rep45           NUMBER,
           rep46           NUMBER,
           rep47           NUMBER,
           rep48           NUMBER,
```


Copyright © 2004 NEC Corporation

```
rep49           NUMBER,
rep50           NUMBER,
rep51           NUMBER,
rep52           NUMBER,
rep53           NUMBER,
rep54           NUMBER,
rep55           NUMBER,
rep56           NUMBER,
rep57           NUMBER,
rep58           NUMBER,
rep59           NUMBER,
rep60           NUMBER,
rep61           NUMBER,
rep62           NUMBER,
rep63           NUMBER,
rep64           NUMBER,
rep65           NUMBER,
rep66           NUMBER,
rep67           NUMBER,
rep68           NUMBER,
rep69           NUMBER,
rep70           NUMBER,
rep71           NUMBER,
rep72           NUMBER,
rep73           NUMBER,
rep74           NUMBER,
rep75           NUMBER,
rep76           NUMBER,
rep77           NUMBER,
rep78           NUMBER,
rep79           NUMBER,
rep80           NUMBER,
rep81           NUMBER,
rep82           NUMBER,
rep83           NUMBER,
rep84           NUMBER,
rep85           NUMBER,
rep86           NUMBER,
rep87           NUMBER,
rep88           NUMBER,
rep89           NUMBER,
rep90           NUMBER,
rep91           NUMBER,
rep92           NUMBER,
rep93           NUMBER,
rep94           NUMBER,
rep95           NUMBER,
rep96           NUMBER,
rep97           NUMBER,
rep98           NUMBER,
rep99           NUMBER,
rep100          NUMBER,
thk1            NUMBER,
thk2            NUMBER,
thk3            NUMBER,
thk4            NUMBER,
thk5            NUMBER,
thk6            NUMBER,
thk7            NUMBER,
thk8            NUMBER,
thk9            NUMBER,
thk10           NUMBER,
thk11           NUMBER,
thk12           NUMBER,
thk13           NUMBER,
thk14           NUMBER,
thk15           NUMBER,
thk16           NUMBER,
thk17           NUMBER,
thk18           NUMBER,
thk19           NUMBER,
thk20           NUMBER,
thk21           NUMBER,
thk22           NUMBER,
thk23           NUMBER,
thk24           NUMBER,
thk25           NUMBER,
key1            NUMBER,
key2            NUMBER,
key3            NUMBER,
key4            NUMBER,
key5            NUMBER,
key6            NUMBER,
key7            NUMBER,
key8            NUMBER,
key9            NUMBER,
key10           NUMBER
        );

rem
rem  Results for order status transactions.
rem  These results are from the measurement interval only.
rem
        CREATE TABLE bench_ord_res
        (
            run_name        VARCHAR2(20),
            audit_str       VARCHAR2(10),
            proc_no         NUMBER,
            rep1            NUMBER,
            rep2            NUMBER,
            rep3            NUMBER,
            rep4            NUMBER,
            rep5            NUMBER,
            rep6            NUMBER,
            rep7            NUMBER,
            rep8            NUMBER,
            rep9            NUMBER,
            rep10           NUMBER,
            rep11           NUMBER,
            rep12           NUMBER,
            rep13           NUMBER,
            rep14           NUMBER,
```


Copyright © 2004 NEC Corporation

```
rep15           NUMBER,
rep16           NUMBER,
rep17           NUMBER,
rep18           NUMBER,
rep19           NUMBER,
rep20           NUMBER,
rep21           NUMBER,
rep22           NUMBER,
rep23           NUMBER,
rep24           NUMBER,
rep25           NUMBER,
rep26           NUMBER,
rep27           NUMBER,
rep28           NUMBER,
rep29           NUMBER,
rep30           NUMBER,
rep31           NUMBER,
rep32           NUMBER,
rep33           NUMBER,
rep34           NUMBER,
rep35           NUMBER,
rep36           NUMBER,
rep37           NUMBER,
rep38           NUMBER,
rep39           NUMBER,
rep40           NUMBER,
rep41           NUMBER,
rep42           NUMBER,
rep43           NUMBER,
rep44           NUMBER,
rep45           NUMBER,
rep46           NUMBER,
rep47           NUMBER,
rep48           NUMBER,
rep49           NUMBER,
rep50           NUMBER,
rep51           NUMBER,
rep52           NUMBER,
rep53           NUMBER,
rep54           NUMBER,
rep55           NUMBER,
rep56           NUMBER,
rep57           NUMBER,
rep58           NUMBER,
rep59           NUMBER,
rep60           NUMBER,
rep61           NUMBER,
rep62           NUMBER,
rep63           NUMBER,
rep64           NUMBER,
rep65           NUMBER,
rep66           NUMBER,
rep67           NUMBER,
rep68           NUMBER,
rep69           NUMBER,
rep70           NUMBER,
rep71           NUMBER,
rep72           NUMBER,
rep73           NUMBER,
rep74           NUMBER,
rep75           NUMBER,
rep76           NUMBER,
rep77           NUMBER,
rep78           NUMBER,
rep79           NUMBER,
rep80           NUMBER,
rep81           NUMBER,
rep82           NUMBER,
rep83           NUMBER,
rep84           NUMBER,
rep85           NUMBER,
rep86           NUMBER,
rep87           NUMBER,
rep88           NUMBER,
rep89           NUMBER,
rep90           NUMBER,
rep91           NUMBER,
rep92           NUMBER,
rep93           NUMBER,
rep94           NUMBER,
rep95           NUMBER,
rep96           NUMBER,
rep97           NUMBER,
rep98           NUMBER,
rep99           NUMBER,
rep100          NUMBER,
thk1            NUMBER,
thk2            NUMBER,
thk3            NUMBER,
thk4            NUMBER,
thk5            NUMBER,
thk6            NUMBER,
thk7            NUMBER,
thk8            NUMBER,
thk9            NUMBER,
thk10           NUMBER,
thk11           NUMBER,
thk12           NUMBER,
thk13           NUMBER,
thk14           NUMBER,
thk15           NUMBER,
thk16           NUMBER,
thk17           NUMBER,
thk18           NUMBER,
thk19           NUMBER,
thk20           NUMBER,
thk21           NUMBER,
thk22           NUMBER,
thk23           NUMBER,
thk24           NUMBER,
thk25           NUMBER,
key1            NUMBER,
```


Copyright © 2004 NEC Corporation

```
            key2            NUMBER,
            key3            NUMBER,
            key4            NUMBER,
            key5            NUMBER,
            key6            NUMBER,
            key7            NUMBER,
            key8            NUMBER,
            key9            NUMBER,
            key10           NUMBER
        );

rem
rem  Aggregate results for delivery transactions.
rem  These results are from the measurement interval only.
rem
        CREATE TABLE tpcc_del_res
        (
            run_name        VARCHAR2(20),
            rep1            NUMBER,
            rep2            NUMBER,
            rep3            NUMBER,
            rep4            NUMBER,
            rep5            NUMBER,
            rep6            NUMBER,
            rep7            NUMBER,
            rep8            NUMBER,
            rep9            NUMBER,
            rep10           NUMBER,
            rep11           NUMBER,
            rep12           NUMBER,
            rep13           NUMBER,
            rep14           NUMBER,
            rep15           NUMBER,
            rep16           NUMBER,
            rep17           NUMBER,
            rep18           NUMBER,
            rep19           NUMBER,
            rep20           NUMBER,
            rep21           NUMBER,
            rep22           NUMBER,
            rep23           NUMBER,
            rep24           NUMBER,
            rep25           NUMBER,
            rep26           NUMBER,
            rep27           NUMBER,
            rep28           NUMBER,
            rep29           NUMBER,
            rep30           NUMBER,
            rep31           NUMBER,
            rep32           NUMBER,
            rep33           NUMBER,
            rep34           NUMBER,
            rep35           NUMBER,
            rep36           NUMBER,
            rep37           NUMBER,
            rep38           NUMBER,
            rep39           NUMBER,
            rep40           NUMBER,
            rep41           NUMBER,
            rep42           NUMBER,
            rep43           NUMBER,
            rep44           NUMBER,
            rep45           NUMBER,
            rep46           NUMBER,
            rep47           NUMBER,
            rep48           NUMBER,
            rep49           NUMBER,
            rep50           NUMBER,
            rep51           NUMBER,
            rep52           NUMBER,
            rep53           NUMBER,
            rep54           NUMBER,
            rep55           NUMBER,
            rep56           NUMBER,
            rep57           NUMBER,
            rep58           NUMBER,
            rep59           NUMBER,
            rep60           NUMBER,
            rep61           NUMBER,
            rep62           NUMBER,
            rep63           NUMBER,
            rep64           NUMBER,
            rep65           NUMBER,
            rep66           NUMBER,
            rep67           NUMBER,
            rep68           NUMBER,
            rep69           NUMBER,
            rep70           NUMBER,
            rep71           NUMBER,
            rep72           NUMBER,
            rep73           NUMBER,
            rep74           NUMBER,
            rep75           NUMBER,
            rep76           NUMBER,
            rep77           NUMBER,
            rep78           NUMBER,
            rep79           NUMBER,
            rep80           NUMBER,
            rep81           NUMBER,
            rep82           NUMBER,
            rep83           NUMBER,
            rep84           NUMBER,
            rep85           NUMBER,
            rep86           NUMBER,
            rep87           NUMBER,
            rep88           NUMBER,
            rep89           NUMBER,
            rep90           NUMBER,
            rep91           NUMBER,
            rep92           NUMBER,
            rep93           NUMBER,
            rep94           NUMBER,
```


Copyright © 2004 NEC Corporation

```
            rep95           NUMBER,
            rep96           NUMBER,
            rep97           NUMBER,
            rep98           NUMBER,
            rep99           NUMBER,
            rep100          NUMBER,
            thk1            NUMBER,
            thk2            NUMBER,
            thk3            NUMBER,
            thk4            NUMBER,
            thk5            NUMBER,
            thk6            NUMBER,
            thk7            NUMBER,
            thk8            NUMBER,
            thk9            NUMBER,
            thk10           NUMBER,
            thk11           NUMBER,
            thk12           NUMBER,
            thk13           NUMBER,
            thk14           NUMBER,
            thk15           NUMBER,
            thk16           NUMBER,
            thk17           NUMBER,
            thk18           NUMBER,
            thk19           NUMBER,
            thk20           NUMBER,
            thk21           NUMBER,
            thk22           NUMBER,
            thk23           NUMBER,
            thk24           NUMBER,
            thk25           NUMBER,
            key1            NUMBER,
            key2            NUMBER,
            key3            NUMBER,
            key4            NUMBER,
            key5            NUMBER,
            key6            NUMBER,
            key7            NUMBER,
            key8            NUMBER,
            key9            NUMBER,
            key10           NUMBER
        );

rem
rem  Results for delivery transactions.
rem  These results are from the measurement interval only.
rem
        CREATE TABLE bench_del_res
        (
            run_name        VARCHAR2(20),
            audit_str       VARCHAR2(10),
            proc_no         NUMBER,
            rep1            NUMBER,
            rep2            NUMBER,
            rep3            NUMBER,
            rep4            NUMBER,
            rep5            NUMBER,
            rep6            NUMBER,
            rep7            NUMBER,
            rep8            NUMBER,
            rep9            NUMBER,
            rep10           NUMBER,
            rep11           NUMBER,
            rep12           NUMBER,
            rep13           NUMBER,
            rep14           NUMBER,
            rep15           NUMBER,
            rep16           NUMBER,
            rep17           NUMBER,
            rep18           NUMBER,
            rep19           NUMBER,
            rep20           NUMBER,
            rep21           NUMBER,
            rep22           NUMBER,
            rep23           NUMBER,
            rep24           NUMBER,
            rep25           NUMBER,
            rep26           NUMBER,
            rep27           NUMBER,
            rep28           NUMBER,
            rep29           NUMBER,
            rep30           NUMBER,
            rep31           NUMBER,
            rep32           NUMBER,
            rep33           NUMBER,
            rep34           NUMBER,
            rep35           NUMBER,
            rep36           NUMBER,
            rep37           NUMBER,
            rep38           NUMBER,
            rep39           NUMBER,
            rep40           NUMBER,
            rep41           NUMBER,
            rep42           NUMBER,
            rep43           NUMBER,
            rep44           NUMBER,
            rep45           NUMBER,
            rep46           NUMBER,
            rep47           NUMBER,
            rep48           NUMBER,
            rep49           NUMBER,
            rep50           NUMBER,
            rep51           NUMBER,
            rep52           NUMBER,
            rep53           NUMBER,
            rep54           NUMBER,
            rep55           NUMBER,
            rep56           NUMBER,
            rep57           NUMBER,
            rep58           NUMBER,
            rep59           NUMBER,
            rep60           NUMBER,
```


Copyright © 2004 NEC Corporation

```
rep61           NUMBER,
rep62           NUMBER,
rep63           NUMBER,
rep64           NUMBER,
rep65           NUMBER,
rep66           NUMBER,
rep67           NUMBER,
rep68           NUMBER,
rep69           NUMBER,
rep70           NUMBER,
rep71           NUMBER,
rep72           NUMBER,
rep73           NUMBER,
rep74           NUMBER,
rep75           NUMBER,
rep76           NUMBER,
rep77           NUMBER,
rep78           NUMBER,
rep79           NUMBER,
rep80           NUMBER,
rep81           NUMBER,
rep82           NUMBER,
rep83           NUMBER,
rep84           NUMBER,
rep85           NUMBER,
rep86           NUMBER,
rep87           NUMBER,
rep88           NUMBER,
rep89           NUMBER,
rep90           NUMBER,
rep91           NUMBER,
rep92           NUMBER,
rep93           NUMBER,
rep94           NUMBER,
rep95           NUMBER,
rep96           NUMBER,
rep97           NUMBER,
rep98           NUMBER,
rep99           NUMBER,
rep100          NUMBER,
thk1            NUMBER,
thk2            NUMBER,
thk3            NUMBER,
thk4            NUMBER,
thk5            NUMBER,
thk6            NUMBER,
thk7            NUMBER,
thk8            NUMBER,
thk9            NUMBER,
thk10           NUMBER,
thk11           NUMBER,
thk12           NUMBER,
thk13           NUMBER,
thk14           NUMBER,
thk15           NUMBER,
thk16           NUMBER,
thk17           NUMBER,
thk18           NUMBER,
thk19           NUMBER,
thk20           NUMBER,
thk21           NUMBER,
thk22           NUMBER,
thk23           NUMBER,
thk24           NUMBER,
thk25           NUMBER,
key1            NUMBER,
key2            NUMBER,
key3            NUMBER,
key4            NUMBER,
key5            NUMBER,
key6            NUMBER,
key7            NUMBER,
key8            NUMBER,
key9            NUMBER,
key10           NUMBER
        );

rem
rem  Aggregate results for stock level transactions.
rem  These results are from the measurement interval only.
rem
        CREATE TABLE tpcc_sto_res
        (
            run_name        VARCHAR2(20),
            rep1            NUMBER,
            rep2            NUMBER,
            rep3            NUMBER,
            rep4            NUMBER,
            rep5            NUMBER,
            rep6            NUMBER,
            rep7            NUMBER,
            rep8            NUMBER,
            rep9            NUMBER,
            rep10           NUMBER,
            rep11           NUMBER,
            rep12           NUMBER,
            rep13           NUMBER,
            rep14           NUMBER,
            rep15           NUMBER,
            rep16           NUMBER,
            rep17           NUMBER,
            rep18           NUMBER,
            rep19           NUMBER,
            rep20           NUMBER,
            rep21           NUMBER,
            rep22           NUMBER,
            rep23           NUMBER,
            rep24           NUMBER,
            rep25           NUMBER,
            rep26           NUMBER,
            rep27           NUMBER,
            rep28           NUMBER,
```


Copyright © 2004 NEC Corporation

```
rep29           NUMBER,
rep30           NUMBER,
rep31           NUMBER,
rep32           NUMBER,
rep33           NUMBER,
rep34           NUMBER,
rep35           NUMBER,
rep36           NUMBER,
rep37           NUMBER,
rep38           NUMBER,
rep39           NUMBER,
rep40           NUMBER,
rep41           NUMBER,
rep42           NUMBER,
rep43           NUMBER,
rep44           NUMBER,
rep45           NUMBER,
rep46           NUMBER,
rep47           NUMBER,
rep48           NUMBER,
rep49           NUMBER,
rep50           NUMBER,
rep51           NUMBER,
rep52           NUMBER,
rep53           NUMBER,
rep54           NUMBER,
rep55           NUMBER,
rep56           NUMBER,
rep57           NUMBER,
rep58           NUMBER,
rep59           NUMBER,
rep60           NUMBER,
rep61           NUMBER,
rep62           NUMBER,
rep63           NUMBER,
rep64           NUMBER,
rep65           NUMBER,
rep66           NUMBER,
rep67           NUMBER,
rep68           NUMBER,
rep69           NUMBER,
rep70           NUMBER,
rep71           NUMBER,
rep72           NUMBER,
rep73           NUMBER,
rep74           NUMBER,
rep75           NUMBER,
rep76           NUMBER,
rep77           NUMBER,
rep78           NUMBER,
rep79           NUMBER,
rep80           NUMBER,
rep81           NUMBER,
rep82           NUMBER,
rep83           NUMBER,
rep84           NUMBER,
rep85           NUMBER,
rep86           NUMBER,
rep87           NUMBER,
rep88           NUMBER,
rep89           NUMBER,
rep90           NUMBER,
rep91           NUMBER,
rep92           NUMBER,
rep93           NUMBER,
rep94           NUMBER,
rep95           NUMBER,
rep96           NUMBER,
rep97           NUMBER,
rep98           NUMBER,
rep99           NUMBER,
rep100          NUMBER,
thk1            NUMBER,
thk2            NUMBER,
thk3            NUMBER,
thk4            NUMBER,
thk5            NUMBER,
thk6            NUMBER,
thk7            NUMBER,
thk8            NUMBER,
thk9            NUMBER,
thk10           NUMBER,
thk11           NUMBER,
thk12           NUMBER,
thk13           NUMBER,
thk14           NUMBER,
thk15           NUMBER,
thk16           NUMBER,
thk17           NUMBER,
thk18           NUMBER,
thk19           NUMBER,
thk20           NUMBER,
thk21           NUMBER,
thk22           NUMBER,
thk23           NUMBER,
thk24           NUMBER,
thk25           NUMBER,
key1            NUMBER,
key2            NUMBER,
key3            NUMBER,
key4            NUMBER,
key5            NUMBER,
key6            NUMBER,
key7            NUMBER,
key8            NUMBER,
key9            NUMBER,
key10           NUMBER
        );

rem
rem  Results for stock level transactions.
rem  These results are from the measurement interval only.
```


Copyright © 2004 NEC Corporation

```
rem
        CREATE TABLE bench_sto_res
        (
            run_name        VARCHAR2(20),
            audit_str       VARCHAR2(10),
            proc_no         NUMBER,
            rep1            NUMBER,
            rep2            NUMBER,
            rep3            NUMBER,
            rep4            NUMBER,
            rep5            NUMBER,
            rep6            NUMBER,
            rep7            NUMBER,
            rep8            NUMBER,
            rep9            NUMBER,
            rep10           NUMBER,
            rep11           NUMBER,
            rep12           NUMBER,
            rep13           NUMBER,
            rep14           NUMBER,
            rep15           NUMBER,
            rep16           NUMBER,
            rep17           NUMBER,
            rep18           NUMBER,
            rep19           NUMBER,
            rep20           NUMBER,
            rep21           NUMBER,
            rep22           NUMBER,
            rep23           NUMBER,
            rep24           NUMBER,
            rep25           NUMBER,
            rep26           NUMBER,
            rep27           NUMBER,
            rep28           NUMBER,
            rep29           NUMBER,
            rep30           NUMBER,
            rep31           NUMBER,
            rep32           NUMBER,
            rep33           NUMBER,
            rep34           NUMBER,
            rep35           NUMBER,
            rep36           NUMBER,
            rep37           NUMBER,
            rep38           NUMBER,
            rep39           NUMBER,
            rep40           NUMBER,
            rep41           NUMBER,
            rep42           NUMBER,
            rep43           NUMBER,
            rep44           NUMBER,
            rep45           NUMBER,
            rep46           NUMBER,
            rep47           NUMBER,
            rep48           NUMBER,
            rep49           NUMBER,
            rep50           NUMBER,
            rep51           NUMBER,
            rep52           NUMBER,
            rep53           NUMBER,
            rep54           NUMBER,
            rep55           NUMBER,
            rep56           NUMBER,
            rep57           NUMBER,
            rep58           NUMBER,
            rep59           NUMBER,
            rep60           NUMBER,
            rep61           NUMBER,
            rep62           NUMBER,
            rep63           NUMBER,
            rep64           NUMBER,
            rep65           NUMBER,
            rep66           NUMBER,
            rep67           NUMBER,
            rep68           NUMBER,
            rep69           NUMBER,
            rep70           NUMBER,
            rep71           NUMBER,
            rep72           NUMBER,
            rep73           NUMBER,
            rep74           NUMBER,
            rep75           NUMBER,
            rep76           NUMBER,
            rep77           NUMBER,
            rep78           NUMBER,
            rep79           NUMBER,
            rep80           NUMBER,
            rep81           NUMBER,
            rep82           NUMBER,
            rep83           NUMBER,
            rep84           NUMBER,
            rep85           NUMBER,
            rep86           NUMBER,
            rep87           NUMBER,
            rep88           NUMBER,
            rep89           NUMBER,
            rep90           NUMBER,
            rep91           NUMBER,
            rep92           NUMBER,
            rep93           NUMBER,
            rep94           NUMBER,
            rep95           NUMBER,
            rep96           NUMBER,
            rep97           NUMBER,
            rep98           NUMBER,
            rep99           NUMBER,
            rep100          NUMBER,
            thk1            NUMBER,
            thk2            NUMBER,
            thk3            NUMBER,
            thk4            NUMBER,
            thk5            NUMBER,
            thk6            NUMBER,
```


Copyright © 2004 NEC Corporation

```
            thk7            NUMBER,
            thk8            NUMBER,
            thk9            NUMBER,
            thk10           NUMBER,
            thk11           NUMBER,
            thk12           NUMBER,
            thk13           NUMBER,
            thk14           NUMBER,
            thk15           NUMBER,
            thk16           NUMBER,
            thk17           NUMBER,
            thk18           NUMBER,
            thk19           NUMBER,
            thk20           NUMBER,
            thk21           NUMBER,
            thk22           NUMBER,
            thk23           NUMBER,
            thk24           NUMBER,
            thk25           NUMBER,
            key1            NUMBER,
            key2            NUMBER,
            key3            NUMBER,
            key4            NUMBER,
            key5            NUMBER,
            key6            NUMBER,
            key7            NUMBER,
            key8            NUMBER,
            key9            NUMBER,
            key10           NUMBER
        );
commit;
set echo off;
rem spool off;
rem exit;
```

require_vars.sh

```
#!/bin/sh
# make sure each environment variable argument is defined.

for var in $* ; do
  if test -z "`eval echo \\$$var`"; then
    echo Missing variable: $var - exiting.
    exit 0
  fi
done
exit 1
```

lcm.sh

```
#!/bin/sh
# echo the lcm of two numbers
if expr $2 \> $1 > /dev/null; then
  set $2 $1
# now $1 is guaranteed to be bigger
fi

lcm=$1
while expr \( \( $lcm % $1 \) + \( $lcm % $2 \) \) \> 0 > /dev/null; do
  lcm=`expr $lcm + $1`
done

echo $lcm
```

tokilobytes.sh

```
#!/bin/sh
# convert k, m, g, t into kilobytes, echo result
# no units means kilobytes is assumed.

amount=`echo $1 | sed -e's/^\(.*\)\(.\)/\1/'`
unit=`echo $1 | sed -e's/^.*\(.\)/\1/'`
if expr $unit = k \| $unit = K> /dev/null; then
  result=$amount
elif expr $unit = m \| $unit = M> /dev/null; then
  result=`$tpcc_bcexpr $amount \* $tpcc_kilo_bytes`
elif expr $unit = g \| $unit = G> /dev/null; then
  result=`$tpcc_bcexpr $amount \* $tpcc_kilo_bytes \* $tpcc_kilo_bytes`
elif expr $unit = t \| $unit = T> /dev/null; then
  result=`$tpcc_bcexpr $amount \* $tpcc_kilo_bytes \* $tpcc_kilo_bytes
\* $tpcc_kilo_bytes`
else
  #assume it's already correct.
  result=$result
fi

if test -n "$result"; then
  echo $result
else
#check if it's still valid
  echo $1
  expr $1 + 0 > /dev/null
  exit $?
fi
```

fromkilobytes.sh

```
#!/bin/sh
# round up to k, m, g, t from number of kilobytes.

amount=$1
if $tpcc_isneg `$tpcc_bcexpr $amount - $tpcc_kilo_bytes`; then
  echo ${amount}K
```


Copyright © 2004 NEC Corporation

```
  exit 0;
fi;
amount=`$tpcc_bcexpr \( $amount + $tpcc_kilo_bytes - 1 \) /
$tpcc_kilo_bytes`
if $tpcc_isneg `$tpcc_bcexpr $amount - $tpcc_kilo_bytes`; then
  echo ${amount}M
  exit 0;
fi;
amount=`$tpcc_bcexpr \( $amount + $tpcc_kilo_bytes - 1 \) /
$tpcc_kilo_bytes`
echo ${amount}G
```

estsize.sh

```
#!/bin/sh
# round down closest k or m from number of kilobytes.

# fairly small, doesn't really matter
amount=$1
if $tpcc_isneg `$tpcc_bcexpr $amount - 10000`; then
  echo ${amount}K
  exit 0;
fi;

# convert to megs, then trunc to nearest 100
amount=`$tpcc_bcexpr \( $amount + $tpcc_kilo_bytes - 1 \) /
$tpcc_kilo_bytes`
amount=`$tpcc_bcexpr \( $amount + 9 \) / 10`
amount=`$tpcc_bcexpr $amount  \* 10`

echo ${amount}M
exit 0;
```

notneg.sh

```
#!/bin/sh
# echo the first and second params provided $2 is greater than -1.

if expr $2 \> -1 > /dev/null; then
  echo $1 $2
else
  echo
fi
```

isneg.sh

```
#!/bin/sh
# exit true if negative, else false

if test `$tpcc_bcexpr "$*" | cut -b1` = -; then
  exit 0
else
  exit 1
fi
```

bcexpr.sh

```
#!/bin/sh
# send command line to bc
echo "$*" | bc
```

tabledata.sh

```
# tabledata- (not a script) data and functions for
# getting columns and info for creating tables and indices.

# forces any env variables we set to be exported
set -a

# note- clustcols will be an option- move to stepenv file!

#tp- get table param.  (that is, $tpcc_tablename_tableparam)
tp(){
  eval echo \""\$tpcc_$1_$2"\"
}

#nullauto- check if $1 null or auto, if not echo it, otherwise echo $2.
nullauto(){
  if expr "x$1" = x \| "x$1" = xauto \| "x$1" = xno > /dev/null; then
    echo "$2"
  else
    echo "$1"
  fi
}

tpcc_item_warecol=none
tpcc_item_clustcols=t
tpcc_item_hkey="`nullauto "$tpcc_item_hkey" 100000`"

if test $tpcc_np -gt 1 ; then
  tpcc_item_hash_def="(i_id + 1)"
else
  tpcc_item_hash_def="(i_id)"
fi

tpcc_item_hash="`nullauto "$tpcc_item_hash" "$tpcc_item_hash_def"`"
tpcc_item_rsize="`nullauto "$tpcc_item_rsize" 120`"
#The actual size of item needs to be a little bigger just because
#it's around boundary size for hash clusters.
tpcc_item_crsize=130

tpcc_item_cols=`cat <<!
```


Copyright © 2004 NEC Corporation

```
        i_id-number(6,0)
        i_name-varchar2(24)
        i_price-number
        i_data-varchar2(50)
        i_im_id-number
!
`

tpcc_iitem_cols=$tpcc_item_cols
tpcc_iitem_warecol=none
tpcc_iitem_indexon=item

tpcc_stok_row_factor=100000
tpcc_stok_rac_storage=t
tpcc_stok_2_RowPerBlock=5
if expr x$tpcc_compress = xt > /dev/null; then
  tpcc_stok_2_RowPerBlock=7
fi


tpcc_stok_4_RowPerBlock=11
if expr x$tpcc_overflow = xt > /dev/null; then
  if expr x$tpcc_compress = xt > /dev/null; then
    tpcc_stok_4_RowPerBlock=15
  fi
fi

tpcc_stok_8_RowPerBlock=23
if expr x$tpcc_overflow = xt > /dev/null; then
  if expr x$tpcc_compress = xt > /dev/null; then
    tpcc_stok_8_RowPerBlock=32
  fi
fi


if expr x$tpcc_compress = xt > /dev/null; then
  tpcc_stok_compress=fffffffttttttttttt
  tpcc_stok_rsize="`nullauto "$tpcc_stok_rsize" 256`"
  if expr "x$tpcc_ieee_number" = "xt" >/dev/null; then
    tpcc_stok_rsize=290
  fi
tpcc_stok_cols=`cat <<!
        s_i_id-number
        s_w_id-number
        s_quantity-number
        s_ytd-number
        s_order_cnt-number
        s_remote_cnt-number
        s_data-varchar2(50)
        s_dist_01-varchar2(24)
        s_dist_02-varchar2(24)
        s_dist_03-varchar2(24)
        s_dist_04-varchar2(24)
        s_dist_05-varchar2(24)
        s_dist_06-varchar2(24)
        s_dist_07-varchar2(24)
        s_dist_08-varchar2(24)
        s_dist_09-varchar2(24)
        s_dist_10-varchar2(24)
!
`
else
tpcc_stok_rsize="`nullauto "$tpcc_stok_rsize" 350`"
tpcc_stok_cols=`cat <<!
        s_i_id-number
        s_w_id-number
        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)
!
`
fi
tpcc_stok_clustcols=tt
tpcc_stok_hkey_def=`$tpcc_bcexpr 100000 \\* $tpcc_scale`
tpcc_stok_hkey="`nullauto "$tpcc_stok_hkey" "$tpcc_stok_hkey_def"`"
if expr $tpcc_stok_imp = parttable \| $tpcc_stok_imp = partiot \|
$tpcc_np \> 1 > /dev/null; then
  tmpdiv=`expr $tpcc_scale / $tpcc_np`
  tpcc_stok_hash_def="(abs(s_i_id - 1) * $tmpdiv + mod((s_w_id - 1),
$tmpdiv) + trunc ((s_w_id - 1) / $tmpdiv) * $tmpdiv * 100000)"
else
  tpcc_stok_hash_def="(s_i_id * $tpcc_scale + s_w_id)"
fi
tpcc_stok_hash="`nullauto "$tpcc_stok_hash" "$tpcc_stok_hash_def"`"

tpcc_stok_warecol=s_w_id

tpcc_istok_cols=$tpcc_stok_cols
tpcc_istok_warecol=$tpcc_stok_warecol
tpcc_istok_indexon=stok

tpcc_ware_row_factor=1
tpcc_ware_2_RowPerBlock=1
tpcc_ware_4_RowPerBlock=1
tpcc_ware_8_RowPerBlock=1
tpcc_ware_rac_storage=t
tpcc_ware_clustcols=t
tpcc_ware_hkey="`nullauto "$tpcc_ware_hkey" "$tpcc_scale"`"

if test $tpcc_np -gt 1 ; then
```


Copyright © 2004 NEC Corporation

```
  tpcc_ware_hash_def="(w_id - 1)"
else
  tpcc_ware_hash_def="(w_id - 1)"
fi

tpcc_ware_hash="`nullauto "$tpcc_ware_hash" "$tpcc_ware_hash_def"`"

#now we want default for ware to be auto (one block always)
#tpcc_ware_rsize="`nullauto "$tpcc_ware_rsize" 1536`"
tpcc_ware_rsize="`nullauto "$tpcc_ware_rsize" auto`"

tpcc_ware_warecol=none
tpcc_ware_cols=`cat <<!
        w_id-number
        w_ytd-number
        w_tax-number
        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)
!
`

tpcc_iware_cols=$tpcc_ware_cols
tpcc_iware_warecol=none
tpcc_iware_indexon=ware

tpcc_dist_row_factor=10
tpcc_dist_2_RowPerBlock=1
tpcc_dist_4_RowPerBlock=1
tpcc_dist_8_RowPerBlock=1
tpcc_dist_rac_storage=t
tpcc_dist_clustcols=tt
tpcc_dist_hkey_def=`$tpcc_bcexpr 10 \* $tpcc_scale`
tpcc_dist_hkey="`nullauto "$tpcc_dist_hkey" $tpcc_dist_hkey_def`"

if test $tpcc_np -gt 1 ; then
  tpcc_dist_hash_def="(((d_w_id - 1 ) * 10) + d_id)"
else
  tpcc_dist_hash_def="((d_w_id * 10) + d_id)"
fi

tpcc_dist_hash="`nullauto "$tpcc_dist_hash" "$tpcc_dist_hash_def"`"
#tpcc_dist_rsize="`nullauto "$tpcc_dist_rsize" 1536`"
tpcc_dist_rsize="`nullauto "$tpcc_dist_rsize" auto`"
tpcc_dist_warecol=d_w_id
tpcc_dist_cols=`cat <<!
        d_id-number
        d_w_id-number
        d_ytd-number
        d_next_o_id-number
        d_tax-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)
!
`

tpcc_idist_cols=$tpcc_dist_cols
tpcc_idist_warecol=d_w_id
tpcc_idist_indexon=dist

#NOTE- needs to change SQL code
tpcc_ordr_clustcols=tttfffffft
tpcc_ordr_queuesort=2t-3t-1tt-9tt-
tpcc_ordr_queuenames=2-3-1-
#NOTE- check if this is right for hash keys
tpcc_ordr_hkey_def=`$tpcc_bcexpr 10 \* $tpcc_scale`
tpcc_ordr_hkey="`nullauto "$tpcc_ordr_hkey" "$tpcc_ordr_hkey_def"`"
tpcc_ordr_hash="`nullauto "$tpcc_ordr_hash" '(o_w_id - 1) * 10 + o_d_id
- 1'`"
tpcc_ordr_rsize="`nullauto "$tpcc_ordr_rsize" 1490`"
tpcc_ordr_warecol=o_w_id

if expr $tpcc_ordr_imp = queue > /dev/null; then
# buildcreatetable sets cols to this after tablecols
# are fetched from normal ordr_cols, so we can get the
# o_number field out for queuesort. (bit of a hack, but
# it would be difficult to redo it another way)
tpcc_ordr_qcols=`cat <<!
        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
        o_number-number
!
`

tpcc_ordr_cols=`cat <<!
        o_id-number-sort
        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
!
`
else
tpcc_ordr_cols=`cat <<!
        o_id-number
        o_w_id-number
```


Copyright © 2004 NEC Corporation

```
        o_d_id-number
        o_c_id-number
        o_carrier_id-number
        o_ol_cnt-number
        o_all_local-number
        o_entry_d-date
!
 `
fi

tpcc_iordr1_warecol=$tpcc_ordr_warecol
tpcc_iordr1_cols=$tpcc_ordr_cols
tpcc_iordr1_indexon=ordr

tpcc_iordr2_warecol=$tpcc_ordr_warecol
tpcc_iordr2_cols=$tpcc_ordr_cols
tpcc_iordr2_indexon=ordr

#ordl uses ordr's cluster, so we represent this
#with another variable
tpcc_ordl_usecluster=', constraint ordl_uk primary key (ol_w_id, ol_d_id,
ol_o_id, ol_number  )) CLUSTER ordrcluster_queue(ol_w_id, ol_d_id,
ol_o_id, ol_number)'
tpcc_ordl_othercluster=ordr

tpcc_ordl_warecol=ol_w_id
if test $tpcc_ordl_imp = queue; then
tpcc_ordl_cols=`cat <<!
        ol_w_id-number
        ol_d_id-number
        ol_o_id-number-sort
        ol_number-number-sort
        ol_i_id-number
        ol_delivery_d-date
        ol_amount-number
        ol_supply_w_id-number
        ol_quantity-number
        ol_dist_info-char(24)
!
 `
else
tpcc_ordl_cols=`cat <<!
        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)
!
 `
fi

tpcc_iordl_cols=$tpcc_ordl_cols
tpcc_iordl_warecol=$tpcc_ordl_warecol
tpcc_iordl_indexon=ordl

tpcc_nord_row_factor=9000
tpcc_nord_2_RowPerBlock=90
tpcc_nord_4_RowPerBlock=210
tpcc_nord_8_RowPerBlock=440
tpcc_nord_clustcols=ttt
tpcc_nord_queuesort=1t-2t-3tt-
tpcc_nord_queuenames=1-2-3-
#NOTE- check if this is right for hash keys
tpcc_nord_hkey_def=`$tpcc_bcexpr 10 \* $tpcc_scale`
tpcc_nord_hkey="`nullauto "$tpcc_nord_hkey" "$tpcc_nord_hkey_def"`"
tpcc_nord_hash="`nullauto "$tpcc_nord_hash" '(no_w_id - 1) * 10 +
no_d_id - 1'`"
tpcc_nord_rsize="`nullauto "$tpcc_nord_rsize" 190`"
tpcc_nord_warecol=no_w_id
tpcc_nord_qcols=`cat <<!
        no_w_id-number
        no_d_id-number
        no_o_id-number
!
 `

tpcc_nord_cols=`cat <<!
        no_w_id-number
        no_d_id-number
        no_o_id-number-sort
!
 `

tpcc_inord_cols=$tpcc_nord_cols
tpcc_inord_warecol=$tpcc_nord_warecol
tpcc_inord_indexon=nord

#we regardless of overflow or compress we always want the
#size calculation for cust to be the same.

tpcc_cust_row_factor=30000
tpcc_cust_2_RowPerBlock=2
tpcc_cust_overflow_size=0
if expr x$tpcc_overflow = xt > /dev/null; then
  tpcc_cust_overflow_size=410
  if expr x$tpcc_compress = xt > /dev/null; then
    tpcc_cust_2_RowPerBlock=11
  else
    tpcc_cust_2_RowPerBlock=5
  fi
fi

tpcc_cust_4_RowPerBlock=4
if expr x$tpcc_overflow = xt > /dev/null; then
  if expr x$tpcc_compress = xt > /dev/null; then
    tpcc_cust_4_RowPerBlock=23
```


Copyright © 2004 NEC Corporation

```
  else
    tpcc_cust_4_RowPerBlock=11
  fi
fi

tpcc_cust_8_RowPerBlock=9
if expr x$tpcc_overflow = xt > /dev/null; then
  if expr x$tpcc_compress = xt > /dev/null; then
    tpcc_cust_8_RowPerBlock=47
  else
    tpcc_cust_8_RowPerBlock=23
  fi
fi

tpcc_cust_rac_storage=t
tpcc_cust_crsize=850

if expr x$tpcc_overflow = xt > /dev/null; then
  if expr x$tpcc_compress = xt > /dev/null; then
    tpcc_cust_compress=fffffffffffffttffffff
    tpcc_cust_rsize="`nullauto "$tpcc_cust_rsize" 180`"
  else
    if expr x$tpcc_ieee_number = xt > /dev/null; then
      tpcc_cust_rsize="`nullauto "$tpcc_cust_rsize" 205`"
    else
      tpcc_cust_rsize="`nullauto "$tpcc_cust_rsize" 350`"
    fi
  fi

tpcc_cust_cols=`cat <<!
        c_id-number
        c_d_id-number
        c_w_id-number
        c_discount-number
        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)
!
`
else
  tpcc_cust_rsize="`nullauto "$tpcc_cust_rsize" 850`"
tpcc_cust_cols=`cat <<!
        c_id-number
        c_d_id-number
        c_w_id-number
        c_discount-number
        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)
!
`
fi

tpcc_cust_clustcols=ttt
tpcc_cust_hkey_def=`$tpcc_bcexpr 30000 \* $tpcc_scale`
tpcc_cust_hkey="`nullauto "$tpcc_cust_hkey" "$tpcc_cust_hkey_def"`"
if expr $tpcc_np \> 1 > /dev/null; then
  tpcc_cust_hash_def='(c_w_id * 30000 + c_id * 10 + c_d_id - 30011)'
else
  tpcc_cust_hash_def="(c_id * ( $tpcc_scale * 10 ) + c_w_id * 10 +
c_d_id)"
fi
tpcc_cust_hash="`nullauto "$tpcc_cust_hash" "$tpcc_cust_hash_def"`"
tpcc_cust_warecol=c_w_id

tpcc_icust1_cols=$tpcc_cust_cols
tpcc_icust1_warecol=$tpcc_cust_warecol
tpcc_icust1_indexon=cust

tpcc_icust2_cols=$tpcc_cust_cols
tpcc_icust2_warecol=$tpcc_cust_warecol
tpcc_icust2_indexon=cust

tpcc_hist_warecol=h_w_id
tpcc_hist_cols=`cat <<!
        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)
!
```


Copyright © 2004 NEC Corporation

```
 `

set +a



#indexnames- with colums for table $1, pull out the desired indices
#and echo each.  Also allowed in format, two characters after the
#index specifying if type and SORT should be listed as well-
#example 1t-4t-2tt-3tt gives the type on each, but only SORT on the
#two entries.
#echos blank if cannot index.
indexnames(){
  tabcols=`tp $1 cols`
  if expr x$2 = xno \| x$2 = x > /dev/null; then
    echo ""
  else
    colsleft=$2
    addcomma=
    while test -n "$colsleft"; do
      desiredcol=`echo $colsleft | cut -f1 -d-`
      yestype=x`echo $desiredcol | cut -b2`
      yessort=x`echo $desiredcol | cut -b3`
      desiredcol=`echo $desiredcol | cut -b1`

      desiredcol=`echo $tabcols | cut -d' ' -f$desiredcol`
      if expr $yestype = xt > /dev/null; then
        firstcol=`echo $desiredcol | cut -f1 -d-`
        desiredcol="$firstcol `echo $desiredcol | cut -f2 -d-`"
      else
        desiredcol=`echo $desiredcol | cut -f1 -d-`
      fi
      if expr $yessort = xt > /dev/null; then
        desiredcol="$desiredcol SORT"
      fi

      echo $addcomma `echo $desiredcol | cut -f1 -d-`
      addcomma=,
# on some systems, we have "lastcol" instead of "lastcol-" like we want
due to cut's behavior.
# we can make it work on these systems by making sure the - is in.
      colsleft=`echo $colsleft | grep -`
      colsleft=`echo $colsleft | cut -f2- -d-`
    done
  fi
}

#colize- get columns desired for table $1, print with commas between.
#second argument is an string like ttfftft, representing
#which colums are to be displayed.
# $3 true if only name should be printed, otherwise name and value are
printed
# $4 has all columns that need to have SORT after them (for queue)
# $5 has all the columns that need to have compress after them (for hash
tables)
colize(){
  if expr $2 = all > /dev/null; then
    allcols=t
  else
    enabledcols=$2
  fi
  sortcols=$4
  compresscols=$5

  tabcols=`tp $1 cols`
  addcomma=
  for curcol in $tabcols ; do
    if expr x`echo $sortcols | cut -b1` = xt > /dev/null; then
      sortsql=SORT
    else
      sortsql=
    fi

    if expr x`echo $compresscols | cut -b1` = xt > /dev/null; then
      compresssql=compress
    else
      compresssql=
    fi

    if expr x`echo $enabledcols | cut -b1` = xt \| x$allcols = xt >
/dev/null; then
      if expr x$3 = xt > /dev/null; then
        echo $addcomma `echo $curcol | cut -f1 -d-`
      else
        name=`echo $curcol | cut -f1 -d-`
        type=`echo $curcol | cut -f2 -d-`
        if expr "x$tpcc_ieee_number" = "xt" >/dev/null; then
          if expr "$name" = "c_balance" \| "$name" = "d_ytd" \| "$name"
= "s_ytd" \| "$name" = "w_ytd" > /dev/null ; then
            type="binary_double"
          fi
          if expr "$name" = "c_ytd_payment" \| "$name" = "c_payment_cnt"
\| "$name" = "c_delivery_cnt" \| "$name" = "h_amount" \| "$name" =
"i_price" \| "$name" = "ol_amount" \| "$name" = "ol_quantity" \| "$name"
= "s_quantity" \| "$name" = "s_order_cnt" \| "$name" = "s_remote_cnt" >
/dev/null ; then
            type="binary_float"
          fi
        fi
        if expr x$allcols = xt > /dev/null; then
#          echo $addcomma $name $type `echo $curcol | cut -f3 -d- | tr .
' '` $sortsql $compresssql
          echo $addcomma $name $type `echo $curcol | cut -f3 -d- | tr .
' '` $sortsql
        else
#          echo $addcomma $name $type $sortsql $compresssql
          echo $addcomma $name $type $sortsql
        fi
      fi
    fi
    addcomma=,
    enabledcols=`echo $enabledcols | cut -b2-`
```


Copyright © 2004 NEC Corporation

```
    sortcols=`echo $sortcols | cut -b2-`
    compresscols=`echo $compresscols | cut -b2-`
  done
}

#createpart- create the sql to generate partitions based on warehouse.
createpart(){
  warecol=`tp $1 warecol`

  #check if unavailable for partitioning
  if test -z "$warecol"; then
    exit 1
  fi
  if expr $warecol = none > /dev/null; then
    exit 1
  fi

  echo "partition by range( $warecol ) ("

  partcount=0
  waredivide=0
  wareinc=`expr $tpcc_scale / $tpcc_np`
  wareextra=`expr $tpcc_scale \% $tpcc_np`
  addcomma=
  while expr $partcount \< $tpcc_np > /dev/null; do
    waredivide=`expr $wareinc + $waredivide + \( $wareextra / $tpcc_np
\)`
    partvalue=`expr 1 + $waredivide`
    if test $partcount = `expr $tpcc_np - 1` ; then
      partvalue="maxvalue"
    fi
    echo $addcomma partition ${1}_$partcount values less than
\( $partvalue \) tablespace ${1}_$partcount
    addcomma=,

    wareextra=`expr $wareextra + 1`
    partcount=`expr $partcount + 1`
  done
  echo ')'
}

#createindexpart- create the sql to generate partitions based on
warehouse,
# this time for indices (so no calculation is required.)
createindexpart(){
  echo 'local ('
  partcount=0
  addcomma=
  while expr $partcount \< $tpcc_np > /dev/null; do
    echo $addcomma partition ${1}_$partcount tablespace ${1}_$partcount
    addcomma=,
    partcount=`expr $partcount + 1`
  done
  echo ')'
}

createindexpart_rac_iordr2() {
  echo 'global partition by range (o_w_id) ('
  partcount=0
  warepernode=`$tpcc_bcexpr $tpcc_scale / $tpcc_np`
  addcomma=
  warenum=1
  while expr $partcount \< $tpcc_np > /dev/null; do
    warenum=`expr $warenum + $warepernode`
    echo $addcomma partition iordr2_${partcount} values less than '('
$warenum ')' tablespace iordr2_${partcount}
    addcomma=,
    partcount=`expr $partcount + 1`
  done
  echo ')'
}

tpcc_ware_tsfileinc=$tpcc_np
tpcc_dist_tsfileinc=$tpcc_np
tpcc_nord_tsfileinc=$tpcc_np
tpcc_iware_tsfileinc=$tpcc_np
tpcc_idist_tsfileinc=$tpcc_np
tpcc_inord_tsfileinc=$tpcc_np
tpcc_item_tsfileinc=1
tpcc_iitem_tsfileinc=1

if test $tpcc_np -eq 1 ; then
  tpcc_iordr2_indices=4-3-2-1-
fi

tpcc_icust1_2_RowPerBlock=90
tpcc_icust1_4_RowPerBlock=191
tpcc_icust1_8_RowPerBlock=395

tpcc_icust2_2_RowPerBlock=39
tpcc_icust2_4_RowPerBlock=82
tpcc_icust2_8_RowPerBlock=170

tpcc_istok_2_RowPerBlock=100
tpcc_istok_4_RowPerBlock=213
tpcc_istok_8_RowPerBlock=438
```

evenload.sh

```
#!/bin/sh
#evenly load using tpcc load, following parameters:
#$1 name of the table to load- this is used to choose where to log.
#$2 the number of things to load
#$3 the starting flag (usually b or j)
#$4 the ending flag (usually e or k)
#$5 the flag to load (h for history , c for cust, S for stock, etc.
#$6 if true, add dummy (only used for -o so far.)
#$7 the command to be used, if not $tpcc_load

command=$7
```


Copyright © 2004 NEC Corporation

```
if test -z "$command"; then
command='$tpcc_load'
fi

tablename=$1
# write out to file to load later
if expr "x$tpcc_rac_load" = "xt" > /dev/null ; then
  loadout=$tpcc_genscripts_dir/load${tablename}_node${tpcc_rac_node}.sh
else
  loadout=$tpcc_genscripts_dir/load${tablename}.sh
fi
rm -f ${loadout}
echo \#created automatically by $0 `date` > $loadout
echo "rm -f load${tablename}*.log" >> $loadout
echo 'cd $tpcc_bench' >> $loadout

numloaders=`$tpcc_bcexpr 2 \* $tpcc_cpu`

if expr "x$tpcc_rac_load" = "xt" > /dev/null ; then
  numloaders=`$tpcc_bcexpr $tpcc_np \* $tpcc_cpu \* 2`
fi

if expr $numloaders \> $2 > /dev/null; then
  numloaders=$2
fi

  numloaders=`$tpcc_bcexpr $tpcc_np \* $tpcc_cpu \* 2`

echo "allprocs=" >> $loadout
curstuff=1
stuffextra=`expr $2 \% $numloaders`
stuffinc=`expr $2 / $numloaders`
curloader=0

if expr "x$tpcc_rac_load" = "xt" > /dev/null ; then
  warepernode=`$tpcc_bcexpr $2 / $tpcc_np`
  procpernode=`$tpcc_bcexpr $tpcc_cpu \* 2`
  curstuff=`$tpcc_bcexpr $warepernode \* \( $tpcc_rac_node - 1 \) + 1`
  stuffinc=`expr $warepernode / $procpernode`
  stuffextra=`expr $warepernode \% $procpernode`
  curloader=`$tpcc_bcexpr $procpernode \* \( $tpcc_rac_node - 1 \) + 1`
  endloader=`$tpcc_bcexpr $procpernode \* $tpcc_rac_node + 1`

  while expr $curloader \< $endloader > /dev/null ; do

    newstuff=`expr $curstuff + $stuffinc + \( $stuffextra / $procpernode
\)`
    if expr x$6 = xt > /dev/null; then
      if expr $tpcc_os = unix > /dev/null; then
        adddummy=\${tpcc_disks_location}dummy${curloader}.dat
      else
        # is this what we actually want to do?  check nt stuff
        adddummy=\\\\\\.\\\\dummy${curloader}.dat
      fi
    else
      adddummy=
    fi
    echo "$command -M $tpcc_scale -$5 $adddummy -$3 $curstuff -$4 `expr
$newstuff - 1` >> load${tablename}${curloader}.log 2>&1 &" >> $loadout
    echo 'allprocs="$allprocs ${!}"' >> $loadout

    curstuff=$newstuff
    stuffextra=`expr $stuffextra + 1`
    curloader=`expr 1 + $curloader`
  done



else
  while expr $curloader \< $numloaders > /dev/null; do
    newstuff=`expr $curstuff + $stuffinc + \( $stuffextra / $numloaders
\)`
    if expr x$6 = xt > /dev/null; then
      if expr $tpcc_os = unix > /dev/null; then
        adddummy=\${tpcc_disks_location}dummy${curloader}.dat
      else
        # is this what we actually want to do?  check nt stuff
        adddummy=\\\\\\.\\\\dummy${curloader}.dat
      fi
    else
      adddummy=
    fi
    echo "$command -M $tpcc_scale -$5 $adddummy -$3 $curstuff -$4 `expr
$newstuff - 1` >> load${tablename}${curloader}.log 2>&1 &" >> $loadout
    echo 'allprocs="$allprocs ${!}"' >> $loadout
    curstuff=$newstuff

    stuffextra=`expr $stuffextra + 1`
    curloader=`expr 1 + $curloader`
  done
fi

cat >> $loadout <<!
error=0
for curproc in \$allprocs; do
  wait \$curproc
  error=\`expr \$? + \$error\`
done
exit \`expr \$error != 0\`
!

exit 0
```

listfiles.sh

```
#!/bin/sh
# Write out tablespace files to files.dat so the person in
# charge of the test can relink them.

#automatically export
```


Copyright © 2004 NEC Corporation

```
set -a
tpcc_listfiles=t

rm -f $tpcc_bench/files.dat
cd $tpcc_bench
echo "#generated automatically by files.dat `date`" >>
$tpcc_bench/files.dat
$tpcc_scripts/runscript.sh createts
cat >> $tpcc_bench/files.dat <<!
${tpcc_disks_location}aux.df $tpcc_sysaux_size
${tpcc_disks_location}control_001 25M
${tpcc_disks_location}control_002 25M
${tpcc_disks_location}sp_0 $tpcc_statspack_size
!

i=1
while test $i -le $tpcc_np; do
  cat >> $tpcc_bench/files.dat <<!
${tpcc_disks_location}log_${i}_1 $tpcc_logfile_size
${tpcc_disks_location}log_${i}_2 $tpcc_logfile_size
!
  i=`expr $i + 1`;
done


i=1
while test $i -le $tpcc_np; do
  cat >> $tpcc_bench/files.dat <<!
${tpcc_disks_location}system_${i} $tpcc_system_size
!
  i=`expr $i + 1`;
done

i=1
while test $i -le $tpcc_np; do
  cat >> $tpcc_bench/files.dat <<!
${tpcc_disks_location}roll${i} $tpcc_undo_size
!
  i=`expr $i + 1`;
done

tpcc_listfiles=f
#automatic export off
set +a

echo List of files written to $tpcc_bench/files.dat.  Stopping driver.

exit 99
```

runscript.sh

```
#!/bin/sh
# run a script from the script directory.
# go to log directory so we don't leave a mess in the bench dir.
cd $tpcc_bench/log

if test $tpcc_np -gt 1 ; then
  if test "$1" = "createts" ; then
    ptr=$tpcc_rac_id
    count=0
    total=$tpcc_rac_createts_count
    if test "x$tpcc_rac_createts_count" = "x" ; then
      total=$tpcc_np
    fi
    if test $total -gt $tpcc_np ; then
      total=$tpcc_np
    fi
    if expr x$tpcc_listfiles = xt > /dev/null; then
      ptr=1
      total=$tpcc_np
    fi
    while test $count -lt $total ; do
      /bin/sh $tpcc_genscripts_dir/createts_node${ptr}.sh
      ptr=`expr $ptr + 1`
      if test $ptr -gt $tpcc_np ; then
        ptr=1
      fi
      count=`expr $count + 1`
    done
  else
    if test "$1" = "loadordrordl" ; then
      /bin/sh $tpcc_genscripts_dir/loadordrordl_node${tpcc_rac_id}.sh
    else
      if test "$1" = "loadnord" ; then
        /bin/sh $tpcc_genscripts_dir/loadnord_node${tpcc_rac_id}.sh
      else
        /bin/sh $tpcc_genscripts_dir/${1}.sh
      fi
    fi
  fi
else
  /bin/sh $tpcc_genscripts_dir/${1}.sh
fi
exit $?
```

updateordrordl.sh

```
#!/bin/sh
$tpcc_sqlplus $tpcc_user_pass <<!
set serveroutput on size 80000

declare
  wid  number;
  did  number;
  oid  number;
begin
```


Copyright © 2004 NEC Corporation

```
  for wid in ${5}..${7} loop
    for did in 1..10 loop
       for oid in 1..2100 loop
          update ordr set o_entry_d=sysdate
                      where o_w_id = wid and o_d_id = did and o_id =
oid;
          update ordl set ol_delivery_d = sysdate
                      where ol_w_id = wid and ol_d_id = did and ol_o_id
= oid;
          commit;
       end loop;
    end loop;
    dbms_output.put_line(wid || 'warehouses finished');
  end loop;
end;
/

exit 0
!
```

chg_redo_1.sh

```
#!/bin/sh

sqlplus "/ as sysdba" << EOF
  set echo on
  set serveroutput on
  spool chg_redo0321.log

  alter database add logfile ('/home/oracle/temp/templog1.dbf') size
70M;
  alter database add logfile ('/home/oracle/temp/templog2.dbf') size
70M;

  alter system switch logfile;
  alter system switch logfile;
  alter system checkpoint;

  alter database drop logfile group 1;
  alter database drop logfile group 2;
  spool off
  exit;

EOF
```

chg_redo_2.sh

```
#!/bin/sh

sqlplus "/ as sysdba" << EOF
  set echo on
  set serveroutput on

  alter database add logfile ('/oradata/tpcc50000/log_1_3') size 84992M;
  alter database add logfile ('/oradata/tpcc50000/log_1_4') size 84992M;

  alter system switch logfile;
  alter system switch logfile;
  alter system checkpoint;

  alter database drop logfile group 3;
  alter database drop logfile group 4;

  exit;

EOF
```

B2. Loader Source Code

dpbcpu.c

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

/*
NAME    DPBTIME.C

DESCRIPTION
   Get time in seconds.

NOTES
   Desktop Performance Group

MODIFIED       (MM/DD/YY)
   bmoriart     05/10/95 -  V4.7 Convert from double to clock_t
   MBHULLAR     02/06/95 -  V4.5
*/

#ifdef ORA_NT
# include <windows.h>
# include < time.h>


clock_t dpbcpu(void)
{
  clock_t begin_cpu;

  begin_cpu = clock();
  return(begin_cpu);
}
#endif /* ORA_NT */
```

dpbetime.c

```
/* Copyright (c) Oracle Corporation 1995.  All Rights Reserved. */
```


Copyright © 2004 NEC Corporation

```
/*
NAME    DPBETIME.C

DESCRIPTION
   Get elapsed time in 10ths of millaseconds as a clock_t.

NOTES
   Desktop Performance Group

MODIFIED       (MM/DD/YY)
   B Moriarty   06/02/95 -  V4.8 Initial Version

*/

#ifdef ORA_OS2
#endif /* ORA_OS2 */

#ifdef ORA_NT
#include <windows.h>
#include <sys\types.h>
#include <time.h>
#include <stdio.h>

BOOL First = TRUE;
LARGE_INTEGER ICount;           /* Initial Time */
LARGE_INTEGER Tptms;          /* Ticks per tenth of millisecond */
#endif /* ORA_NT */

# ifdef __STDC__
clock_t dpbetime(void)
# else
clock_t dpbetime()
# endif /* __STDC__ */
{

#ifdef ORA_NT

   LARGE_INTEGER PFreq;              /* Ticks per Second */
   LARGE_INTEGER PCount;      /* Ticks Since 1970 */
   clock_t etime;             /* Elapsed time in tenths of millaseconds
*/

   if (First)  {
      if (!QueryPerformanceFrequency(&PFreq))
         return((clock_t)-1);
      if (!QueryPerformanceCounter(&ICount))
         return((clock_t)-1);
      Tptms.QuadPart = PFreq.QuadPart / 10000;
      First = FALSE;
      return((clock_t)0);
   }
   if (!QueryPerformanceCounter(&PCount))
      return((clock_t)-1);
   etime = (clock_t) ((PCount.QuadPart - ICount.QuadPart) /
Tptms.QuadPart);
   return(etime);

#endif /* ORA_NT */

}
```

dpbfsync.c

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

/*
NAME    DPBFSYNC.C

DESCRIPTION

   Flush o/s buffers to disk for a file.

   Calling fclose() or fflush() is not enough.  These calls will only
flush
   the buffer in the FILE struture by making a write() call to the o/s,
and
   the o/s will probably place these data in its own disk buffers.
   dpbfsync() will cause the o/s disk buffers for a file to be written
to disk.

   This function should normally be called *after* an fflush() is done,
or you
   will miss the data that is buffered in the FILE structure.

NOTES
   Desktop Performance Group

MODIFIED       (MM/DD/YY)
   C Kelly      02/24/94 -  V4.4 initial version
*/


#include <stdio.h>



#ifdef ORA_OS2
int dpbfsync(FILE *fp)
{
  return 0;
}
#endif /* ORA_OS2 */



#ifdef ORA_NT
# include <windows.h>

int dpbfsync(FILE *fp)
{
```


Copyright © 2004 NEC Corporation

```
  if (FlushFileBuffers((HANDLE)(fp->_file)) == FALSE)
  {
    return 1;
  };

  return 0;
}
#endif /* ORA_NT */



#ifdef ORA_AUX

int dpbfsync(fp)
FILE *fp;
{
  if (fsync(fp->_file) == -1)
  {
    return 1;
  };

  return 0;
}
#endif /* ORA_AUX */



#ifdef ORA_NW
int dpbfsync(FILE *fp)
{
  return 0;
}
#endif /* ORA_NW */



#ifdef ORA_DOS
int dpbfsync(FILE *fp)
{
  return 0;
}
#endif /* ORA_DOS */



#ifdef ORA_MAC
#endif /* ORA_MAC */
```

dpbinpgm.c

```
/* Copyright (c) Oracle Corporation 1994.  All Rights Reserved. */

/*
NAME    DPBINPGM.C

DESCRIPTION
   Routine that performs any o/s specific program initialization.

NOTES
   Desktop Performance Group

MODIFIED       (MM/DD/YY)
   C Kelly      04/21/94 -  V4.4 created to support Netware NLMs

*/



#ifdef ORA_NW
#include <process.h>
#include <library.h>

extern int samtid;
extern int samtgid;

#else /* ORA_NW */
#endif /* ORA_NW */



#ifdef __STDC__
void dpbinpgm(void)
#else
void dpbinpgm()
#endif /* __STDC__ */
{
# ifdef ORA_NW

  samtid  = GetThreadID();      /* get this program's thread id
*/
  samtgid = GetThreadGroupID(); /* get this program's thread group id
*/

# else /* ORA_NW */

  return;               /* do nothing for everything else */

# endif /* ORA_NW */
}
```

dpboradt.c

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

/*
NAME    DPBORADT.C

DESCRIPTION
```


Copyright © 2004 NEC Corporation

```
   Get System Date and Time and
    Return in Oracle External SQLT_DAT (Date) Format
    Returns 1-JAN-2000 00:00:00
    when not implemented or when conversion fails

NOTES
   Desktop Performance Group

MODIFIED       (MM/DD/YY)
   bmoriart     05/26/95 -  V4.8 Created
*/

#ifdef ORA_NT
# include <windows.h>
#endif /* ORA_NT */

#ifdef __STDC__
void dpboradt(char *oradt)
#else
void dpboradt(oradt)
unsigned char *oradt;
#endif /* __STDC__ */
{
   char cnvrtOK=TRUE;

# ifdef ORA_NT
   SYSTEMTIME lpst;

   GetLocalTime(&lpst);
   *oradt = (unsigned char)(lpst.wYear / 100) + 100;
   if (*oradt < 119 || *oradt > 120) cnvrtOK=FALSE;
   *(++oradt) = (unsigned char)(lpst.wYear % 100) + 100;
   if (*oradt < 100 || *oradt > 199) cnvrtOK=FALSE;
   *(++oradt) = (unsigned char)(lpst.wMonth);
   if (*oradt < 1 || *oradt > 12) cnvrtOK=FALSE;
   *(++oradt) = (unsigned char)(lpst.wDay);
   if (*oradt < 1 || *oradt > 31) cnvrtOK=FALSE;
   *(++oradt) = (unsigned char)(lpst.wHour) + 1;
   if (*oradt < 1 || *oradt > 24) cnvrtOK=FALSE;
   *(++oradt) = (unsigned char)(lpst.wMinute) + 1;
   if (*oradt < 1 || *oradt > 60) cnvrtOK=FALSE;
   *(++oradt) = (unsigned char)(lpst.wSecond) + 1;
   if (*oradt < 1 || *oradt > 60) cnvrtOK=FALSE;
#else /* ORA_NT */
   cnvrtOK = FALSE;
#endif /* ORA_NT */

   if(!cnvrtOK) { /* Use 1-JAN-2000 00:00:00 */
     *oradt++ = 120;
     *oradt++ = 100;
     *oradt++ = 1;
     *oradt++ = 1;
     *oradt++ = 1;
     *oradt++ = 1;
     *oradt++ = 1;
   }
   return;               /* do nothing for everything else */
}
```

dpbpchk.c

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

/*
NAME    DPBPCHK.C

DESCRIPTION
   Check New Process

NOTES
   Desktop Performance Group

MODIFIED       (MM/DD/YY)
   W Brumiller  02/08/93 -  Correct error handling for NT
   R Keller     01/08/92 -  Initial version

*/



#include "dpbpcntl.h"



#ifdef ORA_OS2                                  /* IBM OS/2 2.0
*/
# define INCL_DOSPROCESS                        /*
*/
# include <os2.h>                               /*
*/

unsigned long dpbpchk(pcntl *info)
{
  ULONG pid;
  APIRET rc;

  rc = DosWaitChild(DCWA_PROCESS,
                        DCWW_WAIT,
                        &info->rcodes,
                        &pid,
                        0);

  return(info->rcodes.codeResult);
};
#endif /* ORA_OS2 */
```


Copyright © 2004 NEC Corporation

```
#ifdef ORA_NT
# include <windows.h>

int dpbpchk(pcntl *info)
{
  DWORD rc;

  if (WaitForSingleObject(info->proc_info.hProcess, INFINITE) ==
0XFFFFFFFF)
  {
    return -1;
  };

  if (GetExitCodeProcess(info->proc_info.hProcess, &rc) == FALSE)
  {
    return -1;
  };

  (void)CloseHandle(info->proc_info.hProcess);
  (void)CloseHandle(info->proc_info.hThread);

  return((int)rc);
}
#endif /* ORA_NT */
```

```
#ifdef ORA_AUX
# include <errno.h>

int dpbpchk(info)
pcntl *info;
{
  extern int errno;
  int byte_mask;
  int status;
  int high_byte;
  int child;
  int i;

  byte_mask = 255;       /*  low order 8 bits are 1, bits 8..31 are 0
*/

  do
  {
    child = wait(&status);
    if (errno != ECHILD)
    {
      high_byte = ((status & (byte_mask << 8)) >> 8);
    };
  } while (errno != ECHILD);

  return high_byte;
}
#endif /* ORA_AUX */
```

dpbproc.c

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

/*
NAME    DPBPROC.C

DESCRIPTION
 Create New Process

NOTES
   Desktop Performance Group

MODIFIED       (MM/DD/YY)
   W Brumiller  02/08/93 -  Add flags for minimized window under NT
   R Keller     01/08/92 -  Initial version

*/



#include "dpbpcntl.h"



#ifdef ORA_OS2                                  /* IBM OS/2 2.0
*/
# define INCL_DOSPROCESS
# include <os2.h>                               /*
*/
# include <stdlib.h>                            /*
*/
# include <string.h>                            /*
*/

unsigned long dpbproc(char *i_argv[], pcntl *info)
{
  char *args;
  char *args2;
  char load_error[100];
  char pgm[44];
  APIRET rc;
  int i;

  args2 = args = (char *)malloc(128);

  strcpy(args, i_argv[0]);
  strcpy(pgm, i_argv[0]);
  strcat(pgm, ".exe");
```


Copyright © 2004 NEC Corporation

```
  args2 += strlen(args) + 1;

  if (i_argv[1] != NULL)
  {
    strcpy(args2, i_argv[1]);
    for (i = 2; i_argv[i] != NULL; i++)
    {
      strcat(args2, " ");
      strcat(args2, i_argv[i]);
    };
  }
  else
  {
    *args2 = '\0';
  };

  rc = DosExecPgm(load_error,                        /* spawn process
*/
                  sizeof(load_error),
                  EXEC_ASYNCRESULT,
                  args,
                  0,
                  &info->rcodes,
                  pgm);

  free(args);

  return rc;
}
#endif /* ORA_OS2 */
```

```
#ifdef ORA_NT                                /* Microsoft Windows NT
*/
#include <windows.h>
#include <stdlib.h>                          /*
*/
#include <string.h>                          /*
*/

int dpbproc(char *i_argv[], pcntl *info)
{
  BOOL rc;
  int i;
  char *args;
  STARTUPINFO start_info;

  args = (char *)malloc(128);

  memset(&start_info, 0x0, sizeof(STARTUPINFO));
  start_info.cb = sizeof(STARTUPINFO);
  start_info.lpTitle = i_argv[0];
  start_info.dwFlags = STARTF_USESHOWWINDOW;
  start_info.wShowWindow = SW_SHOWMINNOACTIVE;

  strcpy(args, i_argv[0]);                           /* get first str
*/

  for (i = 1; i_argv[i] != NULL; i++)
  {
    strcat(args," ");
    strcat(args,i_argv[i]);
  };

  if ((rc = CreateProcess(NULL,                  // image name
                          args,                  // command line
                          NULL,                  // process security attr
                          NULL,                  // thread security attr
                          TRUE,                  // inherit handles
                          CREATE_NEW_CONSOLE,    // creation flags
                          NULL,                  // environment blocks
                          NULL,                  // current directory
                          &start_info,
                          &info->proc_info)) == FALSE)
  {
    return rc;
  };

  return 0;
};

#endif /* ORA_NT */
```

```
#ifdef ORA_AUX
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>

int dpbproc(arg_list, info)
char *arg_list[];
pcntl *info;
{
  char *path = (char *)malloc(strlen(arg_list[0]) + 3);
  pid_t child;

  sprintf(path, "./%s", arg_list[0]);

  if ((child = fork()) == (pid_t)-1)
  {
    free(path);
    return -1;
  }
  else if (child == (pid_t)0)
  {
    return execv(path, arg_list);
```


Copyright © 2004 NEC Corporation

```
  }
  else
  {
    free(path);
    return 0;
  };
}
#endif /* ORA_AUX */
```

dpbprty.c

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

/*
NAME    DPBPRTY.C

DESCRIPTION
 Set O/S Priority.

NOTES
   Desktop Performance Group

MODIFIED       (MM/DD/YY)
   MBHULLAR     03/25/94 -  Change prty_str[1) to case statement
   B Moriarty   11/11/93 -  Add Get Priority
   R Keller     10/18/93 -  Redesign
   R Keller     10/16/93 -  Iniital version

*/


#ifdef ORA_OS2
#include <string.h>
#include <sys\types.h>
#endif /* ORA_OS2 */


#ifdef ORA_NW
#endif /* ORA_NW */


#ifdef ORA_NT
# include <windows.h>
# include <string.h>
# define REALCLASS 'R'
# define HIGHCLASS 'H'
# define NORMALCLASS 'N'
# define IDLECLASS 'I'
#endif /* ORA_NT */


#ifdef ORA_AUX
#endif /* ORA_AUX */


#ifdef __STDC__
int dpbprty(char *prty_str)
#else
int dpbprty(prty_str)
char *prty_str;
#endif
{
#ifdef ORA_OS2
  return 0;
#endif /* ORA_OS2 */

#ifdef ORA_AUX
  return 0;
#endif /* ORA_AUX */

#ifdef ORA_NW
  return 0;
#endif /* ORA_NW */

#ifdef ORA_NT

  HANDLE this_process, this_thread;

  DWORD class;

 int prios;

  if ( (strlen(prty_str) > 2) || prty_str[0] == '0')
  {
    return(0);                      /* return if invalid length or 0 */
  };

  this_process = GetCurrentProcess();

  switch (prty_str[0])
  {
  case IDLECLASS:
  case 'i':
    class = IDLE_PRIORITY_CLASS;
    break;

  case NORMALCLASS:
  case 'n':
    class = NORMAL_PRIORITY_CLASS;
    break;

  case HIGHCLASS:
  case 'h':
    class = HIGH_PRIORITY_CLASS;
    break;

  case REALCLASS:
  case 'r':
    class = REALTIME_PRIORITY_CLASS;
    break;
```


Copyright © 2004 NEC Corporation

```
  };

  if (!SetPriorityClass(this_process, class))
  {
    return(1);
  };

this_thread = GetCurrentThread();
switch(prty_str[1])
 {
  case '1':
  prios = THREAD_PRIORITY_IDLE;
  break;

  case '2':
  prios = THREAD_PRIORITY_LOWEST;
  break;

  case '3':
  prios = THREAD_PRIORITY_BELOW_NORMAL;
  break;

  case '4':
  prios = THREAD_PRIORITY_NORMAL;
  break;

  case '5':
  prios = THREAD_PRIORITY_ABOVE_NORMAL;
  break;

  case '6':
  prios = THREAD_PRIORITY_HIGHEST;
  break;

  case '7':
  prios = THREAD_PRIORITY_TIME_CRITICAL;
  break;

  default:
  break;
} /* End of switch statement */

  if (!SetThreadPriority(this_thread, prios))
  {
    return(2);
  }

  return 0;

# endif /* ORA_NT */

}
```

```
#ifdef __STDC__
int dpbgetprty(char *os_pri, char *prty_str, int os_pri_len)
#else
int dpbgetprty(os_pri, prty_str, os_pri_len)
char *os_pri;
char *prty_str;
int os_pri_len;
#endif /* __STDC__ */
{
#ifdef ORA_OS2
  strncpy(os_pri,prty_str,(size_t)os_pri_len);
  return 0;
#endif /* ORA_OS2 */

#ifdef ORA_AUX
  strncpy(os_pri,prty_str,os_pri_len);
  return 0;
#endif /* ORA_AUX */

#ifdef ORA_NW
  strncpy(os_pri, prty_str, os_pri_len);
  return 0;
#endif /* ORA_NW */

#ifdef ORA_NT

  HANDLE this_process, this_thread;
  DWORD pclass;
  int tpri;

  this_process = GetCurrentProcess();
  pclass = GetPriorityClass(this_process);

  switch (pclass)
  {
  case IDLE_PRIORITY_CLASS:
    strcpy(os_pri,"I");
    break;

  case NORMAL_PRIORITY_CLASS:
    strcpy(os_pri,"N");
    break;

  case HIGH_PRIORITY_CLASS:
    strcpy(os_pri,"H");
    break;

  case REALTIME_PRIORITY_CLASS:
    strcpy(os_pri,"R");
    break;

  default:
    strcpy(os_pri,"?");
    break;
  };
```


Copyright © 2004 NEC Corporation

```
  this_thread=GetCurrentThread();
  tpri=GetThreadPriority(this_thread);
  switch (tpri)
  {
  case THREAD_PRIORITY_IDLE:
    strcat(os_pri,"1");
    break;

  case THREAD_PRIORITY_LOWEST:
    strcat(os_pri,"2");
    break;

  case THREAD_PRIORITY_BELOW_NORMAL:
    strcat(os_pri,"3");
    break;

  case THREAD_PRIORITY_NORMAL:
    strcat(os_pri,"4");
    break;

  case THREAD_PRIORITY_ABOVE_NORMAL:
    strcat(os_pri,"5");
    break;

  case THREAD_PRIORITY_HIGHEST:
    strcat(os_pri,"6");
    break;

  case THREAD_PRIORITY_TIME_CRITICAL:
    strcat(os_pri,"7");
    break;

  default:
    strcat(os_pri,"?");
    break;
  };

  return 0;

#endif /* ORA_NT */
}
```

dpbtimef.c

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

/*
NAME    DPBTIMEF.C

DESCRIPTION
   Get time in seconds as a clock_t.

NOTES
   Desktop Performance Group

MODIFIED       (MM/DD/YY)
   B Moriarty   02/14/95 -  V4.6 fix NT & OS/2
   C Kelly      01/20/94 -  V4.4 added Netware support
   C Kelly      02/05/93 -  V3.1 added A/UX support
   R Keller     03/02/92 -  V3.0

*/

#ifdef ORA_OS2
# define ORA_PC
#endif /* ORA_OS2 */

#ifdef ORA_NT
# define ORA_PC
#endif /* ORA_NT */



#ifdef ORA_PC
# include <sys/types.h>
# include <sys/timeb.h>
# include <stdio.h>
# include <time.h>

# ifdef __STDC__
clock_t dpbtimef(void)
# else
clock_t dpbtimef()
# endif /* __STDC__ */
{
  struct timeb buf;

  ftime(&buf);
  return((clock_t) (buf.time));
}
#endif /* ORA_PC */



#ifdef ORA_AUX
# include <sys/time.h>
double dpbtimef()
{
  struct timeval t;
  int rc;

  do
  {
    rc = gettimeofday(&t, (struct timezone *)0);
  } while (rc != 0);

  return (((double)t.tv_sec) + (((double)t.tv_usec)/1000000));
}
#endif
```


Copyright © 2004 NEC Corporation

```
#ifdef ORA_NW
# include <time.h>
double dpbtimef()
{
  return (double)time(NULL);    /* there is no function with greater
precision */
}
#endif /* ORA_NW */


#ifdef ORA_MAC
# include <types.h>
# include <OSUtils.h>

double dpbtimef()
{
  unsigned long secs;
  GetDateTime(&secs);
  return((double) secs);
}
#endif /* ORA_MAC */
```

dpbwait.c

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

/*
NAME    DPBWAIT.C

DESCRIPTION
   Wait for n milliseconds.

NOTES
   Desktop Performance Group

MODIFIED       (MM/DD/YY)
   R Keller     03/02/92 -  V3.0
*/



#ifdef ORA_OS2
# define INCL_DOS
# include <os2.h>
# include <time.h>

void dpbwait(clock_t i)
{
  DosSleep(i);
}
#endif /* ORA_OS2 */


#ifdef ORA_NW
# include <process.h>
void dpbwait(long i)
{
  delay((unsigned)i);
};
#endif /* ORA_NW */


#ifdef ORA_AUX
void dpbwait(wait_time)
long wait_time;
{
  unsigned secs = (unsigned)(wait_time / 1000);

  while (secs)
  {
    secs = sleep(secs);
  };
}
#endif /* ORA_AUX */


#ifdef ORA_NT
# include <windows.h>

void dpbwait(long i)
{
  Sleep(i);
}
#endif /* ORA_NT */


#ifdef ORA_DOS
# include <time.h>

void dpbwait(long i)
{
  long current_time;
  long target_time;

  current_time = time(NULL);
  target_time = current_time + i/1000;

  while (current_time < target_time)
  {
    current_time = time(NULL);
  };
}
```


Copyright © 2004 NEC Corporation

```
#endif /* ORA_DOS */
```

dpbxtpgm.c

```
/* Copyright (c) Oracle Corporation 1994.  All Rights Reserved. */

/*
NAME    DPBXTPGM.C

DESCRIPTION
   Routine that performs any o/s specific program exit operations.

NOTES
   Desktop Performance Group

MODIFIED       (MM/DD/YY)
   C Kelly      04/21/94 -  V4.4 created to support Netware NLMs

*/



#ifdef ORA_NW
#include <process.h>
#include <library.h>

extern int samtid;
extern int samtgid;

#else /* ORA_NW */
#endif /* ORA_NW */



#ifdef __STDC__
void dpbxtpgm(void)
#else
void dpbxtpgm()
#endif /* __STDC__ */
{
#ifdef ORA_NW

/*
** Cleanup code for NetWare.
**   This routine will cleanup any Oracle connection should the module
**   be unexpectedly unloaded.
*/

  int oldtgid;

  oldtgid = SetThreadGroupID(samtgid);  /* switch to this NLM's thread
group */
  OraClientExit(samtid);                /* cleanup Oracle connection
*/
  SetThreadGroupID(oldtgid);            /* reset the thread group
*/

#else /* ORA_NW */

  return;               /* do nothing for everything else */

#endif /* ORA_NW */
}
```

gettime.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: gettime.c 7030100.1 96/05/21 15:31:36 plai Generic<base>
$ Copyr (c) 1993 Oracle";
#endif /* RCSID */

/*=====================================================================+
 |        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.
 +=====================================================================*/
/*
**  Options:
**     TIME_W_TIMES:      implement gettime() with times().
**     TIME_W_GETTIME:    implement gettime() with gettimeofday().
**     CPU_W_TIMES:       implement getcpu()  with times().
**     CPU_W_GETRU:       implement getcpu()  with getrusage().
**     GETRU_STATS:       collect getrusage statistics
**     GET_P_STATS:       collect get_process_stats statistics
*/

#if defined(sequent) || defined(SEQ_PSX)
# define GET_P_STATS
#endif /* sequent */

#if defined(aix) || defined(AIXRIOS)
# define TIME_W_GETTIME
# define CPU_W_TIMES
# define GETRU_STATS
```


Copyright © 2004 NEC Corporation

```
#endif /* AIXRIOS */

#if defined(a_osf) || defined(A_OSF)
# define TIME_W_GETTIME
# define CPU_W_GETRU
# define GETRU_STATS
#endif /* AIXRIOS */

#if !defined(TIME_W_GETTIME) && !defined(TIME_W_TIMES)
# define TIME_W_TIMES
#endif

#if !defined(CPU_W_GETRU) && !defined(CPU_W_TIMES)
# define CPU_W_TIMES
#endif

#ifdef GET_P_STATS
# ifdef GETRU_STATS
#  undef GETRU_STATS
# endif
#endif

#if defined(TIME_W_GETTIME) || defined(CPU_W_GETRU) ||
defined(GETRU_STATS)
# include <sys/time.h>
#endif /* TIME_W_GETTIME || CPU_W_GETRU || GETRU_STATS */

#if defined(CPU_W_GETRU) || defined(GETRU_STATS)
# include <sys/resource.h>
#endif /* CPU_W_GETRU || GETRU_STATS */

#if defined(TIME_W_TIMES) || defined (CPU_W_TIMES)
# include <sys/types.h>
# include <sys/times.h>
# include <sys/param.h>  /* most systems define HZ here */
#endif /* TIME_W_TIMES or CPU_W_TIMES */

#ifdef GET_P_STATS
# include <sys/types.h>
# include <sys/procstats.h>
#endif /* GET_P_STATS */

# include <stdio.h>

#ifdef GETRU_STATS
struct rusage selfru;
struct rusage kidsru;
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
struct process_stats selfru;
struct process_stats kidsru;
#endif /* GET_P_STATS */
```

```
void getwait(clock_t secs)
{
  printf("sleep = %lu\n", (secs/1000) / HZ);
  printf("hz = %lu\n", HZ);
  sleep((secs/1000) / HZ);
}

clock_t getetime()
{
   struct tms buf;

   return ((times (&buf) / HZ)*10000);
}

double gettime ()

{

#ifdef TIME_W_GETTIME
   struct timeval tv;

   (void) gettimeofday (&tv, (struct timezone *) 0);
   return ((double) tv.tv_sec + (1.0e-6 * (double) tv.tv_usec));
#endif /* TIME_W_GETTIME */

#ifdef TIME_W_TIMES
   struct tms buf;

   return ((double) times (&buf) / HZ);
#endif /* TIME_W_TIMES */

}



double getcpu ()

{

#ifdef CPU_W_TIMES
   struct tms buf;

   (void) times (&buf);
   return (((double) buf.tms_utime + (double) buf.tms_stime) / HZ);
#endif /* CPU_W_TIMES */

#ifdef CPU_W_GETRU
   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);
#endif /* CPU_W_GETRU */
```


Copyright © 2004 NEC Corporation

```
}



getru (fp, kids, config, runname, proc_no)

FILE *fp;
int kids;
char *config;
char *runname;
int proc_no;

{

#ifdef GETRU_STATS
   struct rusage ru;

   fprintf (fp, "%-10.10s %-10.10s %10d %10d ", config,runname, proc_no,
kids);
   getrusage (kids ? RUSAGE_CHILDREN : RUSAGE_SELF, &ru);
   print_ru (fp, &ru);
   fprintf (fp, "\n");
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
   timeval_t tv;
   struct process_stats ru;

   fprintf (fp, "%-10.10s %-10.10s %10d %10d ", config,runname, proc_no,
kids);
   if (kids)
      get_process_stats (&tv, PS_SELF, (struct process_stats *) 0, &ru);
   else
      get_process_stats (&tv, PS_SELF, &ru, (struct process_stats *) 0);
   print_ru (fp, &ru);
   fprintf (fp, "\n");
#endif /* GET_P_STATS */

}



getru1 (kids)

int kids;

{

#ifdef GETRU_STATS
   if (kids) {
      memset (&kidsru, 0, sizeof (kidsru));
      getrusage (RUSAGE_CHILDREN, &kidsru);
   }
   else {
      memset (&selfru, 0, sizeof (selfru));
      getrusage (RUSAGE_SELF, &selfru);
   }
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
   timeval_t tv;

   if (kids) {
      memset (&kidsru, 0, sizeof (kidsru));
      get_process_stats (&tv, PS_SELF, (struct process_stats *) 0,
&kidsru);
   }
   else {
      memset (&selfru, 0, sizeof (selfru));
      get_process_stats (&tv, PS_SELF, &selfru, (struct process_stats *)
0);
   }
#endif /* GET_P_STATS */

}



getru2 (fp, kids, config, runname, proc_no)

FILE *fp;
int kids;
char *config;
char *runname;
int proc_no;

{

#ifdef GETRU_STATS
   struct rusage ru;

   fprintf (fp, "%-10.10s %-10.10s %10d %10d ", config, runname, proc_no,
kids);
   getrusage (kids ? RUSAGE_CHILDREN : RUSAGE_SELF, &ru);
   if (kids)
      diffru (&ru, &kidsru);
   else
      diffru (&ru, &selfru);
   print_ru (fp, &ru);
   fprintf (fp, "\n");
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
   timeval_t tv;
   struct process_stats ru;

   fprintf (fp, "%-10.10s %-10.10s %10d %10d ", config, runname, proc_no,
kids);
   if (kids)
      get_process_stats (&tv, PS_SELF, (struct process_stats *) 0, &ru);
   else
```


Copyright © 2004 NEC Corporation

```
      get_process_stats (&tv, PS_SELF, &ru, (struct process_stats *) 0);
   if (kids)
      diffru (&ru, &kidsru);
   else
      diffru (&ru, &selfru);
   print_ru (fp, &ru);
   fprintf (fp, "\n");
#endif /* GET_P_STATS */

}



#ifdef GETRU_STATS

print_ru (fp, ru)

FILE *fp;
struct rusage *ru;

{

   fprintf (fp, "%10ld ", ru->ru_utime.tv_sec * 1000 +
                          (ru->ru_utime.tv_usec/1000));
   fprintf (fp, "%10ld ", ru->ru_stime.tv_sec * 1000 +
                          (ru->ru_stime.tv_usec/1000));
   fprintf (fp, "%10ld ", ru->ru_maxrss);
   fprintf (fp, "%10ld ", ru->ru_majflt);
   fprintf (fp, "%10ld ", ru->ru_minflt);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", ru->ru_nswap);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", ru->ru_nvcsw);
   fprintf (fp, "%10ld ", ru->ru_nivcsw);
   fprintf (fp, "%10ld ", ru->ru_nsignals);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", ru->ru_inblock);
   fprintf (fp, "%10ld ", ru->ru_oublock);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld",  0);

}



diffru (ru2, ru)

struct rusage *ru2;
struct rusage *ru;

{

   ru2->ru_utime.tv_sec -= ru->ru_utime.tv_sec;
   ru2->ru_utime.tv_usec -= ru->ru_utime.tv_usec;
   ru2->ru_stime.tv_sec -= ru->ru_stime.tv_sec;
   ru2->ru_stime.tv_usec -= ru->ru_stime.tv_usec;
   ru2->ru_maxrss -= ru->ru_maxrss;
   ru2->ru_ixrss -= ru->ru_ixrss;
   ru2->ru_idrss -= ru->ru_idrss;
   ru2->ru_minflt -= ru->ru_minflt;
   ru2->ru_majflt -= ru->ru_majflt;
   ru2->ru_nswap -= ru->ru_nswap;
   ru2->ru_inblock -= ru->ru_inblock;
   ru2->ru_oublock -= ru->ru_oublock;
   ru2->ru_msgsnd -= ru->ru_msgsnd;
   ru2->ru_msgrcv -= ru->ru_msgrcv;
   ru2->ru_nsignals -= ru->ru_nsignals;
   ru2->ru_nvcsw -= ru->ru_nvcsw;
   ru2->ru_nivcsw -= ru->ru_nivcsw;

}

#endif /* GETRU_STATS */



#ifdef GET_P_STATS

print_ru (fp, ps)

FILE *fp;
struct process_stats *ps;

{

   fprintf (fp, "%lu ", ps->ps_utime.tv_sec * 1000 +
                        (ps->ps_utime.tv_usec/1000));
   fprintf (fp, "%lu ", ps->ps_stime.tv_sec * 1000 +
                         (ps->ps_stime.tv_usec/1000));
   fprintf (fp, "%lu ", ps->ps_maxrss);
   fprintf (fp, "%lu ", ps->ps_pagein);
   fprintf (fp, "%lu ", ps->ps_reclaim);
   fprintf (fp, "%lu ", ps->ps_zerofill);
   fprintf (fp, "%lu ", ps->ps_pffincr);
   fprintf (fp, "%lu ", ps->ps_pffdecr);
   fprintf (fp, "%lu ", ps->ps_swap);
   fprintf (fp, "%lu ", ps->ps_syscall);
   fprintf (fp, "%lu ", ps->ps_volcsw);
   fprintf (fp, "%lu ", ps->ps_involcsw);
   fprintf (fp, "%lu ", ps->ps_signal);
   fprintf (fp, "%lu ", ps->ps_lread);
   fprintf (fp, "%lu ", ps->ps_lwrite);
   fprintf (fp, "%lu ", ps->ps_bread);
   fprintf (fp, "%lu ", ps->ps_bwrite);
   fprintf (fp, "%lu ", ps->ps_phread);
   fprintf (fp, "%lu",  ps->ps_phwrite);

}
```


Copyright © 2004 NEC Corporation

```
diffru (ru2, ru)

struct process_stats *ru2;
struct process_stats *ru;

{

   ru2->ps_utime.tv_sec -= ru->ps_utime.tv_sec;
   ru2->ps_utime.tv_usec -= ru->ps_utime.tv_usec;
   ru2->ps_stime.tv_sec -= ru->ps_stime.tv_sec;
   ru2->ps_stime.tv_usec -= ru->ps_stime.tv_usec;
   ru2->ps_maxrss -= ru->ps_maxrss;
   ru2->ps_pagein -= ru->ps_pagein;
   ru2->ps_reclaim -= ru->ps_reclaim;
   ru2->ps_zerofill -= ru->ps_zerofill;
   ru2->ps_pffincr -= ru->ps_pffincr;
   ru2->ps_pffdecr -= ru->ps_pffdecr;
   ru2->ps_swap -= ru->ps_swap;
   ru2->ps_syscall -= ru->ps_syscall;
   ru2->ps_volcsw -= ru->ps_volcsw;
   ru2->ps_involcsw -= ru->ps_involcsw;
   ru2->ps_signal -= ru->ps_signal;
   ru2->ps_lread -= ru->ps_lread;
   ru2->ps_lwrite -= ru->ps_lwrite;
   ru2->ps_bread -= ru->ps_bread;
   ru2->ps_bwrite -= ru->ps_bwrite;
   ru2->ps_phread -= ru->ps_phread;
   ru2->ps_phwrite -= ru->ps_phwrite;

}

#endif /* GET_P_STATS */
```

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 wares> [options]
 |              options: -A  load all tables
 |                       -w  load ware table
 |                       -d  load dist table
 |                       -c  load cust table
 |                       -i  load item table
 |                       -s  load stok table (cluster around s_w_id)
 |                       -S  load stok table (cluster around s_i_id)
 |                       -h  load hist table
 |                       -n  load new-order table
 |                       -o <oline file> load order and order-line table
 |                       -b <ware#>  beginning ware number
 |                       -e <ware#>  ending ware 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>
#include "dpbcore.h"
# define gettime dpbtimef
# define getcpu  dpbcpu
#define lrand48() ((long)rand() <<15 | rand())
#ifdef __STDC__
#  define PROTO(args)    args
#else
#  define PROTO(args)    ()
#endif
#endif

#define DISTARR 10            /* dist insert array size     */
#define CUSTARR 100           /* cust insert array size     */
#define STOCARR 100           /* stok insert array size     */
#define ITEMARR 100           /* item insert array size     */
#define HISTARR 100             /* hist insert array size    */
#define ORDEARR 100             /* order insert array size      */
#define NEWOARR 100             /* new order insert array size  */

#define DISTFAC 10            /* max. dist id       */
#define CUSTFAC 3000          /* max. cust id       */
#define STOCFAC 100000        /* max. stok 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. */
```


Copyright © 2004 NEC Corporation

```
#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 NOT_SERIALIZABLE  8177  /* ORA-08177: transaction not
serializable */
#define SNAPSHOT_TOO_OLD  1555  /* ORA-01555: snapshot too old */
#define RECOVERR -10
#define IRRECERR -20

#define SQLTXTW "INSERT INTO ware (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 dist (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 SQLTXTCQUERY "select /*+ HASH ( cust ) */ count(*) from cust
where c_w_id = :s_c_w_id and c_d_id = :s_c_d_id and c_id = :s_c_id"

#define SQLTXTC "INSERT INTO cust (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 hist (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 SQLTXTSQUERY "select /*+ HASH ( stok ) */ count(*) from stok
where s_w_id = :s_s_w_id and s_i_id = :s_s_i_id "

#define SQLTXTS "INSERT INTO stok (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_0
6, \
   :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 ordr (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 ordr (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 ordl (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 ordl (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 nord (no_o_id, no_d_id, no_w_id) VALUES
(:no_o_id, :no_d_id, :no_w_id)"

#define SQLTXTENHA "alter session set \"_enable_hash_overflow\"=true"
#define SQLTXTDIHA "alter session set \"_enable_hash_overflow\"=false"

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();
```


Copyright © 2004 NEC Corporation

```
void randdatastr();
void randnum();
void randlastname (char*, int);
int NURand();
void sysdate();

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

OCIStmt *curw;
OCIStmt *curd;
OCIStmt *curc;
OCIStmt *curcs;
OCIStmt *curh;
OCIStmt *curs;
OCIStmt *curss;
OCIStmt *curi;
OCIStmt *curo1;
OCIStmt *curo2;
OCIStmt *curol1;
OCIStmt *curol2;
OCIStmt *curno;

OCIBind *w_id_bp = (OCIBind *) 0;
OCIBind *w_name_bp = (OCIBind *) 0;
OCIBind *w_street1_bp = (OCIBind *) 0;
OCIBind *w_street2_bp = (OCIBind *) 0;
OCIBind *w_city_bp = (OCIBind *) 0;
OCIBind *w_state_bp = (OCIBind *) 0;
OCIBind *w_zip_bp = (OCIBind *) 0;
OCIBind *w_tax_bp = (OCIBind *) 0;

OCIBind *d_id_bp = (OCIBind *) 0;
OCIBind *d_w_id_bp = (OCIBind *) 0;
OCIBind *d_name_bp = (OCIBind *) 0;
OCIBind *d_street1_bp = (OCIBind *) 0;
OCIBind *d_street2_bp = (OCIBind *) 0;
OCIBind *d_city_bp = (OCIBind *) 0;
OCIBind *d_state_bp = (OCIBind *) 0;
OCIBind *d_zip_bp = (OCIBind *) 0;
OCIBind *d_tax_bp = (OCIBind *) 0;

OCIDefine *s_c_ret_bp = (OCIDefine *) 0;
OCIBind *s_c_id_bp = (OCIBind *) 0;
OCIBind *s_c_d_id_bp = (OCIBind *) 0;
OCIBind *s_c_w_id_bp = (OCIBind *) 0;

OCIBind *c_id_bp = (OCIBind *) 0;
OCIBind *c_d_id_bp = (OCIBind *) 0;
OCIBind *c_w_id_bp = (OCIBind *) 0;
OCIBind *c_first_bp = (OCIBind *) 0;
OCIBind *c_last_bp = (OCIBind *) 0;
OCIBind *c_street1_bp = (OCIBind *) 0;
OCIBind *c_street2_bp = (OCIBind *) 0;
OCIBind *c_city_bp = (OCIBind *) 0;
OCIBind *c_state_bp = (OCIBind *) 0;
OCIBind *c_zip_bp = (OCIBind *) 0;
OCIBind *c_phone_bp = (OCIBind *) 0;
OCIBind *c_discount_bp = (OCIBind *) 0;
OCIBind *c_credit_bp = (OCIBind *) 0;
OCIBind *c_data_bp = (OCIBind *) 0;

OCIBind *i_id_bp = (OCIBind *) 0;
OCIBind *i_im_id_bp = (OCIBind *) 0;
OCIBind *i_name_bp = (OCIBind *) 0;
OCIBind *i_price_bp = (OCIBind *) 0;
OCIBind *i_data_bp = (OCIBind *) 0;

OCIDefine *s_s_ret_bp = (OCIDefine *) 0;
OCIBind *s_s_i_id_bp = (OCIBind *) 0;
OCIBind *s_s_w_id_bp = (OCIBind *) 0;

OCIBind *s_i_id_bp = (OCIBind *) 0;
OCIBind *s_w_id_bp = (OCIBind *) 0;
OCIBind *s_quantity_bp = (OCIBind *) 0;
OCIBind *s_dist_01_bp = (OCIBind *) 0;
OCIBind *s_dist_02_bp = (OCIBind *) 0;
OCIBind *s_dist_03_bp = (OCIBind *) 0;
OCIBind *s_dist_04_bp = (OCIBind *) 0;
OCIBind *s_dist_05_bp = (OCIBind *) 0;
OCIBind *s_dist_06_bp = (OCIBind *) 0;
OCIBind *s_dist_07_bp = (OCIBind *) 0;
OCIBind *s_dist_08_bp = (OCIBind *) 0;
OCIBind *s_dist_09_bp = (OCIBind *) 0;
OCIBind *s_dist_10_bp = (OCIBind *) 0;
OCIBind *s_data_bp = (OCIBind *) 0;

OCIBind *h_c_id_bp = (OCIBind *) 0;
OCIBind *h_c_d_id_bp = (OCIBind *) 0;
OCIBind *h_c_w_id_bp = (OCIBind *) 0;
OCIBind *h_d_id_bp = (OCIBind *) 0;
OCIBind *h_w_id_bp = (OCIBind *) 0;
OCIBind *h_data_bp = (OCIBind *) 0;

OCIBind *ol_o_id_bp = (OCIBind *) 0;
OCIBind *ol_d_id_bp = (OCIBind *) 0;
OCIBind *ol_w_id_bp = (OCIBind *) 0;
OCIBind *ol_i_id_bp = (OCIBind *) 0;
OCIBind *ol_number_bp = (OCIBind *) 0;
OCIBind *ol_supply_w_id_bp = (OCIBind *) 0;
OCIBind *ol_dist_info_bp = (OCIBind *) 0;
OCIBind *ol_amount_bp = (OCIBind *) 0;

OCIBind *o_id_bp = (OCIBind *) 0;
OCIBind *o_d_id_bp = (OCIBind *) 0;
OCIBind *o_w_id_bp = (OCIBind *) 0;
OCIBind *o_c_id_bp = (OCIBind *) 0;
OCIBind *o_carrier_id_bp = (OCIBind *) 0;
OCIBind *o_ol_cnt_bp  = (OCIBind *) 0;
```


Copyright © 2004 NEC Corporation

```
OCIBind *o_ocnt_bp = (OCIBind *) 0;
OCIBind *o_olcnt_bp  = (OCIBind *) 0;


OCIBind *no_o_id_bp  = (OCIBind *) 0;
OCIBind *no_d_id_bp  = (OCIBind *) 0;
OCIBind *no_w_id_bp  = (OCIBind *) 0;

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 ware table\n");
   fprintf (stderr, "\t-d :\tload dist table\n");
   fprintf (stderr, "\t-c :\tload cust table\n");
   fprintf (stderr, "\t-i :\tload item table\n");
   fprintf (stderr, "\t-s :\tload stok table (cluster around
s_w_id)\n");
   fprintf (stderr, "\t-S :\tload stok table (cluster around
s_i_id)\n");
   fprintf (stderr, "\t-h :\tload hist 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 ware number\n");
   fprintf (stderr, "\t-e <ware#> :\tending ware 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,"\t   $tpcc_bench must be set to the location of the
kit\n");
   fprintf (stderr, "\n");
   exit(1);
}

int sqlfile(fnam,linebuf)
char   *fnam;
text   *linebuf;
{
    FILE *fd;
    int nulpt = 0;
    char realfile[512];

    sprintf(realfile,"%s",fnam);
    fd = fopen(realfile,"r");
    if (!fd)
    {
      return (0);
    }
    while (fgets((char *)linebuf+nulpt, SQL_BUF_SIZE, fd))
    {
        nulpt = strlen((char *)linebuf);
    }
    return(nulpt);
}

void quit()
{
   OCIERROR(errhp,OCISessionEnd ( tpcsvc,errhp, tpcusr, OCI_DEFAULT));
   OCIERROR(errhp,OCIServerDetach ( tpcsrv, errhp, OCI_DEFAULT));
   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);
}

void main (argc, argv)
int argc;
char *argv[];
{
   char *uid="tpcc";
   char *pwd="tpcc";
   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];
   float w_tax;

   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];
   float d_tax[10];

   int s_c_id;
   int s_c_d_id;
   int s_c_w_id;
```


Copyright © 2004 NEC Corporation

```
int s_c_count;

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_s_count;
int s_s_i_id;
int s_s_w_id;

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[1500];
int ol_d_id[1500];
int ol_w_id[1500];
int ol_number[1500];
int ol_i_id[1500];
int ol_supply_w_id[1500];
int ol_amount[1500];
char ol_dist_info[1500][24];
int o_cnt;
int ol_cnt;

ub2 ol_o_id_len[1500];
ub2 ol_d_id_len[1500];
ub2 ol_w_id_len[1500];
ub2 ol_number_len[1500];
ub2 ol_i_id_len[1500];
ub2 ol_supply_w_id_len[1500];
ub2 ol_dist_info_len[1500];
ub2 ol_amount_len[1500];

ub4 ol_o_id_clen;
ub4 ol_d_id_clen;
ub4 ol_w_id_clen;
ub4 ol_number_clen;
ub4 ol_i_id_clen;
ub4 ol_supply_w_id_clen;
ub4 ol_dist_info_clen;
ub4 ol_amount_clen;

ub2 o_id_len[100];
ub2 o_d_id_len[100];
ub2 o_w_id_len[100];
ub2 o_c_id_len[100];
ub2 o_carrier_id_len[100];
ub2 o_ol_cnt_len[100];

ub4 o_id_clen;
ub4 o_d_id_clen;
ub4 o_w_id_clen;
ub4 o_c_id_clen;
ub4 o_carrier_id_clen;
ub4 o_ol_cnt_clen;

text stmbuf[16*1024];

int no_o_id[100];
int no_d_id[100];
int no_w_id[100];

char sdate[30];

#ifdef ORA_NT
   clock_t begin_time, end_time;
   clock_t begin_cpu, end_cpu;

   char *arg_ptr,  **end_args;
#else
   double begin_time, end_time;
```


Copyright © 2004 NEC Corporation

```
   double begin_cpu, end_cpu;
   double gettime(), getcpu();

   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];
   char* basename;
   int status;
#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;
                   strcpy (fname, *argv++);
                   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;
```


Copyright © 2004 NEC Corporation

```
         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 (do_o) {
      if ((basename = getenv ("tpcc_bench")) == NULL)
      {
         fprintf (stderr, "$tpcc_bench is not set");
         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 ();
      }
```


Copyright © 2004 NEC Corporation

```
   }

/*----------------------------------------------------------------+
 | Prepare to insert into database.                               |
 +----------------------------------------------------------------*/

   sysdate (sdate);
   if (!gen) {

      /* log on to Oracle */

      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);
      OCIServerAttach(tpcsrv, errhp, (text *)0,0,OCI_DEFAULT);
      OCIAttrSet((dvoid *)tpcsvc, OCI_HTYPE_SVCCTX, (dvoid *)tpcsrv,
                 (ub4)0,OCI_ATTR_SERVER, errhp);
      OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&tpcusr,
OCI_HTYPE_SESSION, 0 , (dvoid **)0);
      OCIAttrSet((dvoid *)tpcusr, OCI_HTYPE_SESSION, (dvoid *)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);

      fprintf (stderr, "\nConnected to Oracle userid '%s/%s'.\n", uid,
pwd);

      /* open cursors and parse statement */
      if (do_A || do_w)  {
         OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&curw),
OCI_HTYPE_STMT, 0, (dvoid**)0));
         OCIERROR(errhp,OCIStmtPrepare(curw, errhp, (text *)SQLTXTW,
                     strlen((char *)SQLTXTW), (ub4) OCI_NTV_SYNTAX,
(ub4) OCI_DEFAULT));
      }

      if (do_A || do_d) {
         OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&curd),
OCI_HTYPE_STMT, 0, (dvoid**)0));
         OCIERROR(errhp,OCIStmtPrepare(curd, errhp, (text *)SQLTXTD,
                     strlen((char *)SQLTXTD), (ub4) OCI_NTV_SYNTAX,
(ub4) OCI_DEFAULT));
      }

      if (do_A || do_c) {
         OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&curc),
OCI_HTYPE_STMT, 0, (dvoid**)0));
         OCIERROR(errhp,OCIStmtPrepare(curc, errhp, (text *)SQLTXTC,
                     strlen((char *)SQLTXTC), (ub4) OCI_NTV_SYNTAX,
(ub4) OCI_DEFAULT));
         OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&curcs),
OCI_HTYPE_STMT, 0, (dvoid**)0));
         OCIERROR(errhp,OCIStmtPrepare(curcs, errhp, (text
*)SQLTXTCQUERY,
                     strlen((char *)SQLTXTCQUERY), (ub4) OCI_NTV_SYNTAX,
(ub4) OCI_DEFAULT));
      }

      if (do_A || do_h) {
         OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&curh),
OCI_HTYPE_STMT, 0, (dvoid**)0));
         OCIERROR(errhp,OCIStmtPrepare(curh, errhp, (text *)SQLTXTH,
                     strlen((char *)SQLTXTH), (ub4) OCI_NTV_SYNTAX,
(ub4) OCI_DEFAULT));
      }

      if (do_A || do_s || do_S) {
         OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&curs),
OCI_HTYPE_STMT, 0, (dvoid**)0));
         OCIERROR(errhp,OCIStmtPrepare(curs, errhp, (text *)SQLTXTS,
                     strlen((char *)SQLTXTS), (ub4) OCI_NTV_SYNTAX,
(ub4) OCI_DEFAULT));
         OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&curss),
OCI_HTYPE_STMT, 0, (dvoid**)0));
         OCIERROR(errhp,OCIStmtPrepare(curss, errhp, (text
*)SQLTXTSQUERY,
                     strlen((char *)SQLTXTSQUERY), (ub4) OCI_NTV_SYNTAX,
(ub4) OCI_DEFAULT));
      }

      if (do_A || do_i) {
         OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&curi),
OCI_HTYPE_STMT, 0, (dvoid**)0));
         OCIERROR(errhp,OCIStmtPrepare(curi, errhp, (text *)SQLTXTI,
                     strlen((char *)SQLTXTI), (ub4) OCI_NTV_SYNTAX,
(ub4) OCI_DEFAULT));
      }

      if (do_A || do_o) {
         int stat;
         char fname[160];
         OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&curo1),
OCI_HTYPE_STMT, 0, (dvoid**)0));
         DISCARD strcpy(fname,basename);
         DISCARD strcat(fname, "/");
         DISCARD strcat(fname, "benchrun/blocks/load_ordordl.sql");
         stat = sqlfile(fname, stmbuf);
         if (!stat)
         {
           fprintf (stderr, "unable to open %s \n",fname);
           quit();
```


Copyright © 2004 NEC Corporation

```
          exit(1);
        }
        OCIERROR(errhp,OCIStmtPrepare(curo1, errhp, stmbuf,
                     strlen((char *)stmbuf), (ub4) OCI_NTV_SYNTAX, (ub4)
OCI_DEFAULT));
      }

      if (do_A || do_n) {
        OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&curno),
OCI_HTYPE_STMT, 0, (dvoid**)0));
        OCIERROR(errhp,OCIStmtPrepare(curno, errhp, (text *)SQLTXTNO,
                     strlen((char *)SQLTXTNO), (ub4) OCI_NTV_SYNTAX,
(ub4) OCI_DEFAULT));
      }

      /* bind variables */

      /* warehouse */

      if (do_A || do_w) {
        OCIERROR(errhp, OCIBindByName(curw, &w_id_bp, errhp, (text
*)(":w_id"), strlen(":w_id")),
                      (ub1 *)&(w_id), sizeof(w_id), SQLT_INT, (dvoid *)
0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curw, &w_name_bp, errhp,(text
*)":w_name", strlen(":w_name"),
                      (ub1 *)w_name, 11, SQLT_STR, (dvoid *) 0, (ub2 *)0,
(ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curw, &w_street1_bp, errhp, (text
*)":w_street_1",
                     strlen(":w_street_1"), (ub1 *)w_street_1, 21,
SQLT_STR,
                     (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curw, &w_street2_bp, errhp, (text
*)":w_street_2",
                     strlen(":w_street_2"), (ub1 *)w_street_2, 21,
SQLT_STR,
                     (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curw, &w_city_bp, errhp, (text
*)":w_city",
                     strlen(":w_city"), (ub1 *)w_city, 21, SQLT_STR,
                     (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curw, &w_state_bp, errhp, (text
*)":w_state",
                     strlen(":w_state"), (ub1 *)w_state, 2, SQLT_CHR,
                     (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curw, &w_zip_bp, errhp, (text
*)":w_zip",
                     strlen(":w_zip"), (ub1 *)w_zip, 9, SQLT_CHR,
                     (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curw, &w_tax_bp, errhp, (text
*)":w_tax",
                     strlen(":w_tax"), (ub1 *) & w_tax, sizeof(w_tax),
SQLT_FLT,
                     (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
      }

      /* district */

      if (do_A || do_d) {
        OCIERROR(errhp, OCIBindByName(curd, &d_id_bp, errhp, (text
*)":d_id",
                    strlen(":d_id"), (ub1 *)d_id, sizeof(int), SQLT_INT,
                     (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curd, &d_w_id_bp, errhp, (text
*)":d_w_id",
                     strlen(":d_w_id"), (ub1 *)d_w_id, sizeof(int),
SQLT_INT,
                     (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curd, &d_name_bp, errhp, (text
*)":d_name",
                     strlen(":d_name"), (ub1 *)d_name, 11, SQLT_STR,
                     (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curd, &d_street1_bp, errhp, (text
*)":d_street_1",
                     strlen(":d_street_1"), (ub1 *)d_street_1, 21,
SQLT_STR,
                     (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curd, &d_street2_bp, errhp, (text
*)":d_street_2",
                     strlen(":d_street_2"), (ub1 *)d_street_2, 21,
SQLT_STR,
                     (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curd, &d_city_bp, errhp, (text
*)":d_city",
                     strlen(":d_city"), (ub1 *)d_city, 21, SQLT_STR,
                     (dvoid *) 0, (ub2 *)0, (ub2 *)0,
```


Copyright © 2004 NEC Corporation

```
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curd, &d_state_bp, errhp, (text
*)":d_state",
                        strlen(":d_state"), (ub1 *)d_state, 2, SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curd, &d_zip_bp, errhp, (text
*)":d_zip",
                        strlen(":d_zip"), (ub1 *)d_zip, 9, SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curd, &d_tax_bp, errhp, (text
*)":d_tax",
                        strlen(":d_tax"), (ub1 *)d_tax, sizeof(float),
SQLT_FLT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
      }

      /* customer */

      if (do_A || do_c) {
         OCIERROR(errhp, OCIBindByName(curcs, &s_c_id_bp, errhp, (text
*)":s_c_id",
                        strlen(":s_c_id"), (ub1 *)&s_c_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curcs, &s_c_w_id_bp, errhp, (text
*)":s_c_w_id",
                        strlen(":s_c_w_id"), (ub1 *)&s_c_w_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curcs, &s_c_d_id_bp, errhp, (text
*)":s_c_d_id",
                        strlen(":s_c_d_id"), (ub1 *)&s_c_d_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


OCIDefineByPos(curcs,&s_c_ret_bp,errhp,1,&s_c_count,sizeof(int),SQLT_INT
,\
                        0,0,0,OCI_DEFAULT);

         OCIERROR(errhp, OCIBindByName(curc, &c_id_bp, errhp, (text
*)":c_id",
                        strlen(":c_id"), (ub1 *)c_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curc, &c_d_id_bp, errhp, (text
*)":c_d_id",
                        strlen(":c_d_id"), (ub1 *)c_d_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curc, &c_w_id_bp, errhp, (text
*)":c_w_id",
                        strlen(":c_w_id"), (ub1 *)c_w_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curc, &c_first_bp, errhp, (text
*)":c_first",
                       strlen(":c_first"), (ub1 *)c_first, 17, SQLT_STR,
                       (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curc, &c_last_bp, errhp, (text
*)":c_last",
                       strlen(":c_last"), (ub1 *)c_last, 17, SQLT_STR,
                       (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curc, &c_street1_bp, errhp, (text
*)":c_street_1",
                        strlen(":c_street_1"), (ub1 *)c_street_1, 21,
SQLT_STR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curc, &c_street2_bp, errhp, (text
*)":c_street_2",
                        strlen(":c_street_2"), (ub1 *)c_street_2, 21,
SQLT_STR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curc, &c_city_bp, errhp, (text
*)":c_city",
                        strlen(":c_city"), (ub1 *)c_city, 21, SQLT_STR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curc, &c_state_bp, errhp, (text
*)":c_state",
                        strlen(":c_state"), (ub1 *)c_state, 2, SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curc, &c_zip_bp, errhp, (text
*)":c_zip",
```


Copyright © 2004 NEC Corporation

```
                        strlen(":c_zip"), (ub1 *)c_zip, 9, SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curc, &c_phone_bp, errhp, (text
*)":c_phone",
                        strlen(":c_phone"), (ub1 *)c_phone, 16, SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curc, &c_credit_bp, errhp, (text
*)":c_credit",
                        strlen(":c_credit"), (ub1 *)c_credit, 2, SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curc, &c_discount_bp, errhp,
(text *)":c_discount",
                        strlen(":c_discount"), (ub1 *)c_discount,
sizeof(float), SQLT_FLT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curc, &c_data_bp, errhp, (text
*)":c_data",
                        strlen(":c_data"), (ub1 *)c_data, 501, SQLT_STR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
      }

      /* item */

      if (do_A || do_i) {
         OCIERROR(errhp, OCIBindByName(curi, &i_id_bp, errhp, (text
*)":i_id",
                        strlen(":i_id"), (ub1 *)i_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curi, &i_im_id_bp, errhp, (text
*)":i_im_id",
                        strlen(":i_im_id"), (ub1 *)i_im_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curi, &i_name_bp, errhp, (text
*)":i_name",
                        strlen(":i_name"), (ub1 *)i_name, 25, SQLT_STR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curi, &i_price_bp, errhp, (text
*)":i_price",
                        strlen(":i_price"), (ub1 *)i_price, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curi, &i_data_bp, errhp, (text
*)":i_data",
                        strlen(":i_data"), (ub1 *)i_data, 51, SQLT_STR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
      }

      /* stock */

      if (do_A || do_s || do_S) {
         OCIERROR(errhp, OCIBindByName(curss, &s_s_i_id_bp, errhp, (text
*)":s_s_i_id",
                        strlen(":s_s_i_id"), (ub1 *)&s_s_i_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curss, &s_s_w_id_bp, errhp, (text
*)":s_s_w_id",
                        strlen(":s_s_w_id"), (ub1 *)&s_s_w_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

OCIDefineByPos(curss,&s_s_ret_bp,errhp,1,&s_s_count,sizeof(int),SQLT_INT
,\
                        0,0,0,OCI_DEFAULT);

         OCIERROR(errhp, OCIBindByName(curs, &s_i_id_bp, errhp, (text
*)":s_i_id",
                        strlen(":s_i_id"), (ub1 *)s_i_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curs, &s_w_id_bp, errhp, (text
*)":s_w_id",
                        strlen(":s_w_id"), (ub1 *)s_w_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curs, &s_quantity_bp, errhp,
(text *)":s_quantity",
                        strlen(":s_quantity"), (ub1 *)s_quantity,
sizeof(int), SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curs, &s_dist_01_bp, errhp, (text
*)":s_dist_01",
```


Copyright © 2004 NEC Corporation

```
                        strlen(":s_dist_01"), (ub1 *)s_dist_01, 24,
SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


         OCIERROR(errhp, OCIBindByName(curs, &s_dist_02_bp, errhp, (text
*)":s_dist_02",
                        strlen(":s_dist_02"), (ub1 *)s_dist_02, 24,
SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


         OCIERROR(errhp, OCIBindByName(curs, &s_dist_03_bp, errhp, (text
*)":s_dist_03",
                        strlen(":s_dist_03"), (ub1 *)s_dist_03, 24,
SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


         OCIERROR(errhp, OCIBindByName(curs, &s_dist_04_bp, errhp, (text
*)":s_dist_04",
                        strlen(":s_dist_04"), (ub1 *)s_dist_04, 24,
SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


         OCIERROR(errhp, OCIBindByName(curs, &s_dist_05_bp, errhp, (text
*)":s_dist_05",
                        strlen(":s_dist_05"), (ub1 *)s_dist_05, 24,
SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


         OCIERROR(errhp, OCIBindByName(curs, &s_dist_06_bp, errhp, (text
*)":s_dist_06",
                        strlen(":s_dist_06"), (ub1 *)s_dist_06, 24,
SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


         OCIERROR(errhp, OCIBindByName(curs, &s_dist_07_bp, errhp, (text
*)":s_dist_07",
                        strlen(":s_dist_07"), (ub1 *)s_dist_07, 24,
SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


         OCIERROR(errhp, OCIBindByName(curs, &s_dist_08_bp, errhp, (text
*)":s_dist_08",
                        strlen(":s_dist_08"), (ub1 *)s_dist_08, 24,
SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


         OCIERROR(errhp, OCIBindByName(curs, &s_dist_09_bp, errhp, (text
*)":s_dist_09",
                        strlen(":s_dist_09"), (ub1 *)s_dist_09, 24,
SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


         OCIERROR(errhp, OCIBindByName(curs, &s_dist_10_bp, errhp, (text
*)":s_dist_10",
                        strlen(":s_dist_10"), (ub1 *)s_dist_10, 24,
SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curs, &s_data_bp, errhp, (text
*)":s_data",
                        strlen(":s_data"), (ub1 *)s_data, 51, SQLT_STR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
      }

      /* history */

      if (do_A || do_h) {
         OCIERROR(errhp, OCIBindByName(curh, &h_c_id_bp, errhp, (text
*)":h_c_id",
                        strlen(":h_c_id"), (ub1 *)h_c_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curh, &h_c_d_id_bp, errhp, (text
*)":h_c_d_id",
                        strlen(":h_c_d_id"), (ub1 *)h_d_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curh, &h_c_w_id_bp, errhp, (text
*)":h_c_w_id",
                        strlen(":h_c_w_id"), (ub1 *)h_w_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curh, &h_d_id_bp, errhp, (text
*)":h_d_id",
                        strlen(":h_d_id"), (ub1 *)h_d_id, sizeof(int),
SQLT_INT,
```


Copyright © 2004 NEC Corporation

```
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curh, &h_w_id_bp, errhp, (text
*)":h_w_id",
                        strlen(":h_w_id"), (ub1 *)h_w_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curh, &h_data_bp, errhp, (text
*)":h_data",
                        strlen(":h_data"), (ub1 *)h_data, 25, SQLT_STR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                        (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

      }

      /* order and order_line (delivered) */

      if (do_A || do_o) {

         for (i = 0; i < ORDEARR; i++ ) {
            o_id_len[i] = sizeof(int);
            o_d_id_len[i] = sizeof(int);
            o_w_id_len[i] = sizeof(int);
            o_c_id_len[i] = sizeof(int);
            o_carrier_id_len[i] = sizeof(int);
            o_ol_cnt_len[i] = sizeof(int);
         }

         OCIERROR(errhp, OCIBindByName(curo1, &ol_o_id_bp, errhp, (text
*)":ol_o_id",
                        strlen(":ol_o_id"), (ub1 *)ol_o_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)ol_o_id_len, (ub2 *)0,
                        (ub4) 15*ORDEARR, (ub4 *)&ol_o_id_clen, (ub4)
OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &ol_d_id_bp, errhp, (text
*)":ol_d_id",
                        strlen(":ol_d_id"), (ub1 *)ol_d_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)ol_d_id_len, (ub2 *)0,
                        (ub4) 15*ORDEARR, (ub4 *) &ol_d_id_clen, (ub4)
OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &ol_w_id_bp, errhp, (text
*)":ol_w_id",
                        strlen(":ol_w_id"), (ub1 *)ol_w_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)ol_w_id_len, (ub2 *)0,
                        (ub4) 15*ORDEARR, (ub4 *) &ol_w_id_clen, (ub4)
OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &ol_number_bp, errhp,
(text *)":ol_number",
                        strlen(":ol_number"), (ub1 *)ol_number,
sizeof(int), SQLT_INT,
                        (dvoid *) 0, (ub2 *)ol_number_len, (ub2 *)0,
                        (ub4) 15*ORDEARR, (ub4 *) &ol_number_clen, (ub4)
OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &ol_i_id_bp, errhp, (text
*)":ol_i_id",
                        strlen(":ol_i_id"), (ub1 *)ol_i_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)ol_i_id_len, (ub2 *)0,
                        (ub4) 15*ORDEARR, (ub4 *) &ol_i_id_clen, (ub4)
OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &ol_supply_w_id_bp, errhp,
(text *)":ol_supply_w_id",
                        strlen(":ol_supply_w_id"), (ub1 *)ol_supply_w_id,
sizeof(int), SQLT_INT,
                        (dvoid *) 0, (ub2 *)ol_supply_w_id_len, (ub2 *)0,
                        (ub4) 15*ORDEARR, (ub4 *) &ol_supply_w_id_clen,
(ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &ol_dist_info_bp, errhp,
(text *)":ol_dist_info",
                        strlen(":ol_dist_info"), (ub1 *)ol_dist_info, 24,
SQLT_CHR,
                        (dvoid *) 0, (ub2 *)ol_dist_info_len, (ub2 *)0,
                        (ub4) 15*ORDEARR, (ub4 *) &ol_dist_info_clen,
(ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &ol_amount_bp, errhp,
(text *)":ol_amount",
                        strlen(":ol_amount"), (ub1 *)ol_amount,
sizeof(int), SQLT_INT,
                        (dvoid *) 0, (ub2 *)ol_amount_len, (ub2 *)0,
                        (ub4) 15*ORDEARR, (ub4 *) &ol_amount_clen, (ub4)
OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &o_id_bp, errhp, (text
*)":o_id",
                        strlen(":o_id"), (ub1 *)o_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)o_id_len, (ub2 *)0,
                        (ub4) ORDEARR, (ub4 *) &o_id_clen, (ub4)
OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &o_d_id_bp, errhp, (text
*)":o_d_id",
                        strlen(":o_d_id"), (ub1 *)o_d_id, sizeof(int),
SQLT_INT,
                        (dvoid *) 0, (ub2 *)o_d_id_len, (ub2 *)0,
                        (ub4) ORDEARR, (ub4 *) &o_d_id_clen, (ub4)
OCI_DEFAULT));
```


Copyright © 2004 NEC Corporation

```
         OCIERROR(errhp, OCIBindByName(curo1, &o_w_id_bp, errhp, (text
*)":o_w_id",
                      strlen(":o_w_id"), (ub1 *)o_w_id, sizeof(int),
SQLT_INT,
                      (dvoid *) 0, (ub2 *)o_w_id_len, (ub2 *)0,
                      (ub4) ORDEARR, (ub4 *) &o_w_id_clen, (ub4)
OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &o_c_id_bp, errhp, (text
*)":o_c_id",
                      strlen(":o_c_id"), (ub1 *)o_c_id, sizeof(int),
SQLT_INT,
                      (dvoid *) 0, (ub2 *)o_c_id_len, (ub2 *)0,
                      (ub4) ORDEARR, (ub4 *) &o_c_id_clen, (ub4)
OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &o_carrier_id_bp, errhp,
(text *)":o_carrier_id",
                      strlen(":o_carrier_id"), (ub1 *)o_carrier_id,
sizeof(int), SQLT_INT,
                      (dvoid *) 0, (ub2 *)o_carrier_id_len, (ub2 *)0,
                      (ub4) ORDEARR, (ub4 *) &o_carrier_id_clen, (ub4)
OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &o_ol_cnt_bp, errhp, (text
*)":o_ol_cnt",
                      strlen(":o_ol_cnt"), (ub1 *)o_ol_cnt, sizeof(int),
SQLT_INT,
                      (dvoid *) 0, (ub2 *)o_ol_cnt_len, (ub2 *)0,
                      (ub4) ORDEARR, (ub4 *) &o_ol_cnt_clen, (ub4)
OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &o_ocnt_bp, errhp, (text
*)":order_rows",
                      strlen(":order_rows"), (ub1 *)&o_cnt, sizeof(int),
SQLT_INT,
                      (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curo1, &o_olcnt_bp, errhp, (text
*)":ordl_rows",
                      strlen(":ordl_rows"), (ub1 *)&ol_cnt, sizeof(int),
SQLT_INT,
                      (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
      }

      /* new order */

      if (do_A || do_n) {
         OCIERROR(errhp, OCIBindByName(curno, &no_o_id_bp, errhp, (text
*)":no_o_id",
                      strlen(":no_o_id"), (ub1 *)no_o_id, sizeof(int),
SQLT_INT,
                      (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curno, &no_d_id_bp, errhp, (text
*)":no_d_id",
                      strlen(":no_d_id"), (ub1 *)no_d_id, sizeof(int),
SQLT_INT,
                      (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

         OCIERROR(errhp, OCIBindByName(curno, &no_w_id_bp, errhp, (text
*)":no_w_id",
                      strlen(":no_w_id"), (ub1 *)no_w_id, sizeof(int),
SQLT_INT,
                      (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                      (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
      }
   }

/*-----------------------------------------------------------------+
 | 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++) {

         w_tax = (float) ((lrand48 () % 2001) * 0.0001);
         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) {
            printf ("%d 30000000 %6.4f %s %s %s %s %s %s\n", loop, w_tax,
                    w_name, w_street_1, w_street_2, w_city, str2, num9);
```


Copyright © 2004 NEC Corporation

```
            fflush (stdout);
         }
         else {
            w_id = loop;
            strncpy (w_state, str2, 2);
            strncpy (w_zip, num9, 9);

          status = OCIStmtExecute(tpcsvc, curw, errhp, (ub4) 1, (ub4) 0,
                        (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                        (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
           if (status != OCI_SUCCESS) {
              fprintf (stderr, "Error at ware %d\n", loop);
              OCIERROR(errhp, status);
              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++) {
            d_tax[i] = (float) ((lrand48 () % 2001) * 0.0001);
            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 3000000 %6.4f 3001 %s %s %s %s %s %s\n",
                       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 {
            status = OCIStmtExecute(tpcsvc, curd, errhp, (ub4) DISTARR,
(ub4) 0,
                        (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                        (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
            if (status != OCI_SUCCESS) {
                 fprintf (stderr, "Aborted at ware %d, dist 1\n", dwid);
                  OCIERROR(errhp, status);
                  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);

      if (getenv("tpcc_hash_overflow")) {
        fprintf(stderr,"Hash overflow is enabled\n");
        OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0,
(dvoid**)0);
        sprintf ((char *) stmbuf, SQLTXTENHA);
        OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf),
                       OCI_NTV_SYNTAX, OCI_DEFAULT);
```


Copyright © 2004 NEC Corporation

```
        OCIERROR(errhp,OCIStmtExecute(tpcsvc, curi,
errhp,1,0,0,0,OCI_DEFAULT));
        OCIHandleFree(curi, OCI_HTYPE_STMT);
        fprintf (stderr,"Customer loaded for horizontal
partitioning\n");
      }
      else
      {
        fprintf (stderr,"Customer not loaded for horizontal
partitioning\n");
      }
      begin_time = gettime ();
      begin_cpu = getcpu ();

      s_c_id = 1;
      s_c_d_id = 1;
      s_c_w_id = bware;

      while (s_c_w_id <= eware) {
         status = OCIStmtExecute(tpcsvc, curcs, errhp, (ub4) 1, (ub4) 0,
                  (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                  (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
            if (status != OCI_SUCCESS) {
                  OCIERROR(errhp, status);
                  quit ();
                  exit (1);
            }

            if (s_c_count == 0) {
              s_c_w_id--;
              break;
            }
            else s_c_w_id++;
      }

      if (s_c_w_id < bware ) s_c_w_id = bware;
      else {
        if (s_c_w_id > eware ) s_c_w_id = eware;
        while (s_c_d_id <= DISTFAC) {
           status = OCIStmtExecute(tpcsvc, curcs, errhp, (ub4) 1, (ub4)
0,
                    (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                    (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
              if (status != OCI_SUCCESS) {
                   fprintf (stderr, "Select failed\n");
                    OCIERROR(errhp, status);
                    quit ();
                    exit (1);
              }
              if (s_c_count == 0) {
                s_c_d_id--;
                break;
              }
              else s_c_d_id++;
        }
        if (s_c_d_id > DISTFAC) s_c_d_id = DISTFAC;
        while (s_c_id <= CUSTFAC) {
           status = OCIStmtExecute(tpcsvc, curcs, errhp, (ub4) 1, (ub4)
0,
                    (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                    (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
              if (status != OCI_SUCCESS) {
                    OCIERROR(errhp, status);
                    quit ();
                    exit (1);
              }
              if (s_c_count == 0) break;
              else s_c_id++;
        }
      }
      if (s_c_id > CUSTFAC) {
        s_c_d_id=1;
        s_c_w_id++;
        s_c_id=1;
      }

      fprintf (stderr, "start at wid: %d, did: %d, cid: %d\n
",s_c_w_id, s_c_d_id, s_c_id);
      cid = s_c_id - 1;
      cdid = s_c_d_id;
      cwid = s_c_w_id;
      nrows = (eware - s_c_w_id + 1) * DISTFAC * CUSTFAC - (s_c_d_id -
1) * CUSTFAC - s_c_id + 1;
      fprintf (stderr, "remaining rows: %d\n   ", nrows);
      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < CUSTARR && row < nrows; i++, row++) {
            cid++;
            if (cid > CUSTFAC) {        /* cycle cust id */
               cid = 1;                 /* cheap mod */
               cdid++;                  /* shift dist cycle */
               if (cdid > DISTFAC) {
                  cdid = 1;
                  cwid++;               /* shift ware 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);
```


Copyright © 2004 NEC Corporation

```
            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 {
            status = OCIStmtExecute(tpcsvc, curc, errhp, (ub4) i, (ub4)
0,
                        (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                        (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);

            if (status != OCI_SUCCESS) {
                 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]);
                 OCIERROR(errhp, status);
                 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 < 0 ? 0 : nrows, end_time - begin_time, end_cpu -
begin_cpu);

      if (getenv("tpcc_hash_overflow")) {
        fprintf(stderr,"Hash overflow is disabled\n");
        OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0,
(dvoid**)0);
        sprintf ((char *) stmbuf, SQLTXTDIHA);
        OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf),
                       OCI_NTV_SYNTAX, OCI_DEFAULT);
        OCIERROR(errhp,OCIStmtExecute(tpcsvc, curi,
errhp,1,0,0,0,OCI_DEFAULT));
        OCIHandleFree(curi, OCI_HTYPE_STMT);
      }
   }


/*-----------------------------------------------------------------+
 | 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 {
            status = OCIStmtExecute(tpcsvc, curi, errhp, (ub4) ITEMARR,
(ub4) 0,
                        (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                        (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
            if (status != OCI_SUCCESS) {
                 fprintf (stderr, "Aborted at i_id %d\n", i_id[0]);
```


Copyright © 2004 NEC Corporation

```
                OCIERROR(errhp, status);
                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 ();

     s_s_i_id = 1;
     s_s_w_id = bware;

     while (s_s_w_id <= eware) {
        status = OCIStmtExecute(tpcsvc, curss, errhp, (ub4) 1, (ub4) 0,
                 (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                 (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
           if (status != OCI_SUCCESS) {
                 OCIERROR(errhp, status);
                 quit ();
                 exit (1);
           }
           if (s_s_count == 0) {
             s_s_w_id--;
             break;
           }
           else s_s_w_id++;
     }

     if (s_s_w_id < bware ) s_s_w_id = bware;
     else {
        if (s_s_w_id > eware) s_s_w_id = eware;
        while (s_s_i_id<=STOCFAC) {
           status = OCIStmtExecute(tpcsvc, curss, errhp, (ub4) 1, (ub4)
0,
                    (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                    (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
              if (status != OCI_SUCCESS) {
                    OCIERROR(errhp, status);
                    quit ();
                    exit (1);
              }
              if (s_s_count == 0) {
                break;
              }
              else s_s_i_id++;
        }
     }
     if (s_s_i_id > STOCFAC) {
       s_s_i_id=1;
       s_s_w_id++;
     }

     fprintf(stderr,"start at s_i_id: %d, s_w_id: %d\n   ", s_s_i_id,
s_s_w_id);

     sid = s_s_i_id - 1;
     swid = s_s_w_id;
     nrows = (eware - s_s_w_id + 1) * STOCFAC - ( s_s_i_id - 1);
     fprintf (stderr, "remaining rows: %d\n   ", nrows);
     loopcount = 0;

     for (row = 0; row < nrows; ) {
         /* added row < nrows condition on next line - alex.ni */
       for (i = 0; (i < STOCARR) && (row < nrows); i++, row++) {
            if (++sid > STOCFAC) {        /* cheap mod */
               sid = 1;
               swid++;
            }
            s_quantity[i] = (lrand48 () % 91) + 10;
            randstr (s_dist_01[i], 24, 24);
            randstr (s_dist_02[i], 24, 24);
            randstr (s_dist_03[i], 24, 24);
            randstr (s_dist_04[i], 24, 24);
            randstr (s_dist_05[i], 24, 24);
            randstr (s_dist_06[i], 24, 24);
            randstr (s_dist_07[i], 24, 24);
            randstr (s_dist_08[i], 24, 24);
            randstr (s_dist_09[i], 24, 24);
            randstr (s_dist_10[i], 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], s_dist_01[i],
s_dist_02[i],
```


Copyright © 2004 NEC Corporation

```
                        s_dist_03[i], s_dist_04[i], s_dist_05[i],
s_dist_06[i],
                        s_dist_07[i], s_dist_08[i], s_dist_09[i],
s_dist_10[i],
                        s_data[i]);
            }
            else {
               s_i_id[i] = sid;
               s_w_id[i] = swid;
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
           /* Changed to STOCKARR to i - alex.ni */
            status = OCIStmtExecute(tpcsvc, curs, errhp, (ub4) i, (ub4)
0,
                         (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                         (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
            if (status != OCI_SUCCESS) {
                  fprintf (stderr, "Aborted at w_id %d, s_i_id %d\n",
s_w_id[0], s_i_id[0]);
                  OCIERROR(errhp, status);
                  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 < 0 ? 0 : 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 ();

      s_s_i_id = bitem;
      s_s_w_id = bware;

      while (s_s_i_id <= eitem) {
         status = OCIStmtExecute(tpcsvc, curss, errhp, (ub4) 1, (ub4) 0,
                  (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                  (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
            if (status != OCI_SUCCESS) {
                  OCIERROR(errhp, status);
                  quit ();
                  exit (1);
            }
            if (s_s_count == 0) {
              s_s_i_id--;
              break;
            }
            else s_s_i_id++;
      }

      if (s_s_i_id < bitem ) s_s_i_id = bitem;
      else {
        if (s_s_i_id > eitem) s_s_i_id = eitem;
        while (s_s_w_id <= eware) {
           status = OCIStmtExecute(tpcsvc, curss, errhp, (ub4) 1, (ub4)
0,
                    (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                    (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
              if (status != OCI_SUCCESS) {
                    OCIERROR(errhp, status);
                    quit ();
                    exit (1);
              }
              if (s_s_count == 0) {
                break;
              }
              else s_s_w_id++;
        }
      }
      if (s_s_w_id > eware) {
        s_s_w_id=bware;
        s_s_i_id++;
      }

      fprintf(stderr,"start at s_i_id: %d, s_w_id: %d\n   ", s_s_i_id,
s_s_w_id);

      sid = s_s_i_id;
      swid = s_s_w_id - 1;
      nrows = (eitem - s_s_i_id + 1) * (eware - bware + 1) - (s_s_w_id -
bware);
      fprintf (stderr, "remaining rows: %d\n   ", nrows);
      loopcount = 0;
```


Copyright © 2004 NEC Corporation

```
      for (row = 0; row < nrows; ) {
         for (i = 0; i < STOCARR && row < nrows; i++, row++) {
            if (++swid > eware) {          /* cheap mod */
               swid = bware;
               sid++;
            }
            s_quantity[i] = (lrand48 () % 91) + 10;
            randstr (s_dist_01[i], 24, 24);
            randstr (s_dist_02[i], 24, 24);
            randstr (s_dist_03[i], 24, 24);
            randstr (s_dist_04[i], 24, 24);
            randstr (s_dist_05[i], 24, 24);
            randstr (s_dist_06[i], 24, 24);
            randstr (s_dist_07[i], 24, 24);
            randstr (s_dist_08[i], 24, 24);
            randstr (s_dist_09[i], 24, 24);
            randstr (s_dist_10[i], 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], s_dist_01[i],
s_dist_02[i],
                       s_dist_03[i], s_dist_04[i], s_dist_05[i],
s_dist_06[i],
                       s_dist_07[i], s_dist_08[i], s_dist_09[i],
s_dist_10[i],
                       s_data[i]);
            }
            else {
               s_i_id[i] = sid;
               s_w_id[i] = swid;
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            status = OCIStmtExecute(tpcsvc, curs, errhp, (ub4) i, (ub4)
0,
                        (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                        (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
            if (status != OCI_SUCCESS) {
                  fprintf (stderr, "Aborted at w_id %d, s_i_id %d\n",
s_w_id[0], s_i_id[0]);
                  OCIERROR(errhp, status);
                  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 < 0 ? 0 : 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 {
```


Copyright © 2004 NEC Corporation

```
            status = OCIStmtExecute(tpcsvc, curh, errhp, (ub4) HISTARR,
(ub4) 0,
                        (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                        (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
            if (status != OCI_SUCCESS) {
                  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]);
                  OCIERROR(errhp, status);
                  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) {

      int batch_olcnt;

      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; ) {

         batch_olcnt = 0;

         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];
               if (cid >= 2101 ) {
                   o_carrier_id[i] = 11;
               }
            }

            for (j = 0; j < o_ol_cnt[i]; j++, batch_olcnt++ ) {
               ol_i_id[batch_olcnt] = sid = lrand48 () % 100000 + 1;
               if (cid < 2101)
                  ol_amount[batch_olcnt] = 0;
               else
                  ol_amount[batch_olcnt] = (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[batch_olcnt], cwid,
                              ol_amount[batch_olcnt], 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[batch_olcnt],
cwid,
                              ol_amount[batch_olcnt], str24[j]);
                  }
```


Copyright © 2004 NEC Corporation

```
      }
      else {
         ol_o_id[batch_olcnt] = cid;
         ol_d_id[batch_olcnt] = cdid;
         ol_w_id[batch_olcnt] = cwid;
         ol_number[batch_olcnt] = j + 1;
         ol_supply_w_id[batch_olcnt] = cwid;
         strncpy (ol_dist_info[batch_olcnt], str24[j], 24);
      }
   }
   if (gen) {
      fflush (olfp);
   }
}

o_cnt =  ORDEARR;
ol_cnt =  batch_olcnt;

for (j = 0; j < batch_olcnt; j++) {
   ol_o_id_len[j] = sizeof(int);
   ol_d_id_len[j] = sizeof(int);
   ol_w_id_len[j] = sizeof(int);
   ol_number_len[j] = sizeof(int);
   ol_i_id_len[j] = sizeof(int);
   ol_supply_w_id_len[j] = sizeof(int);
   ol_dist_info_len[j] = 24;
   ol_amount_len[j] = sizeof(int);
}
for (j = batch_olcnt; j < 15*ORDEARR; j++) {
   ol_o_id_len[j] = 0;
   ol_d_id_len[j] = 0;
   ol_w_id_len[j] = 0;
   ol_number_len[j] = 0;
   ol_i_id_len[j] = 0;
   ol_supply_w_id_len[j] = 0;
   ol_dist_info_len[j] = 0;
   ol_amount_len[j] = 0;
}

o_id_clen = ORDEARR;
o_d_id_clen = ORDEARR;
o_w_id_clen = ORDEARR;
o_c_id_clen = ORDEARR;
o_carrier_id_clen = ORDEARR;
o_ol_cnt_clen = ORDEARR;

ol_o_id_clen = batch_olcnt;
ol_d_id_clen = batch_olcnt;
ol_w_id_clen = batch_olcnt;
ol_number_clen = batch_olcnt;
ol_i_id_clen = batch_olcnt;
ol_supply_w_id_clen = batch_olcnt;
ol_dist_info_clen = batch_olcnt;
ol_amount_clen = batch_olcnt;
```

```
         OCIERROR(errhp, OCIStmtExecute(tpcsvc, curo1, errhp, (ub4) 1,
(ub4) 0,
                              (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                              (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS ));

         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;
```


Copyright © 2004 NEC Corporation

```
               no_d_id[i] = cdid;
               no_w_id[i] = cwid;
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            status = OCIStmtExecute(tpcsvc, curno, errhp, (ub4) NEWOARR,
(ub4) 0,
                                    (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                                    (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
            if (status != OCI_SUCCESS) {
              fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n",
cwid, cdid, cid + 2100);
                OCIERROR(errhp, status);
                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';
```


Copyright © 2004 NEC Corporation

```
      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);
}

int ocierror(fname, lineno, errhp, status)
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:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Error - OCI_SUCCESS_WITH_INFO\n");
    lstat = OCIErrorGet (errhp, recno++, (text *) NULL, &errcode, errbuf,
                         (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
    fprintf(stderr,"Error - %s\n", errbuf);
    break;
  case OCI_NEED_DATA:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Error - OCI_NEED_DATA\n");
    return (IRRECERR);
  case OCI_NO_DATA:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Error - OCI_NO_DATA\n");
    return (IRRECERR);
  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)
    {
      fprintf(stderr,"Module %s Line %d\n", fname, lineno);
      fprintf(stderr,"Error - %s\n", errbuf);
      lstat = OCIErrorGet (errhp, recno++, (text *) NULL, &errcode,
errbuf,
                         (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
    }
    return (errcode);
  case OCI_INVALID_HANDLE:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Error - OCI_INVALID_HANDLE\n");
    exit(-1);
  case OCI_STILL_EXECUTING:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Error - OCI_STILL_EXECUTE\n");
    return (IRRECERR);
  case OCI_CONTINUE:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Error - OCI_CONTINUE\n");
    return (IRRECERR);
  default:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
```


Copyright © 2004 NEC Corporation

```
    fprintf(stderr,"Status - %s\n", status);
    return (IRRECERR);
  }
  return (RECOVERR);
}
```

dpbcore.h

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

/*
NAME    DPBCORE.H

DESCRIPTION
   Header for CORE function

NOTES
   Desktop Performance Group

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

*/

#ifndef __dpbcore__
# define __dpbcore__


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

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


#endif /* __dpbcore__ */
```

dpbpcntl.h

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

/*
NAME    DPBPCNTL.H

DESCRIPTION
   OSD structures for process control

NOTES
```


Copyright © 2004 NEC Corporation

```
   Desktop Performance Group

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

*/

#ifndef __dpbpcntl__
# define __dpbpcntl__



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



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



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



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



#endif /* __dpbpcntl__ */
```

Makefile.linux

```
#==================================================================+
#        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA       |
#                  OPEN SYSTEMS PERFORMANCE GROUP                  |
#                       All Rights Reserved                        |
#==================================================================+
# FILENAME
#      Makefile
# DESCRIPTION
#      Makefile for lib for batch driver, load program and tx testing.
#==================================================================
#
#  Programs:
#
#      dpblibunix.o
#

all: compile dpblibunix.o

#include $(ORACLE_HOME)/bench/buildtools/prefix.mk
I_SYM=-I
#include $(ORACLE_HOME)/rdbms/lib/env_rdbms.mk
REMOVE=rm
#CC=/opt/SunProd/SUNWspro6.1/bin/../WS6U1/bin/cc
CC=/usr/bin/gcc

TARGS=compile cleanup

TPCBIN=.
INCLUDE=$(I_SYM). $(I_SYM)$(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

MEMBS=
OBJS=gettime.o dpbproc.o dpbwait.o dpbpchk.o dpbtimef.o

CFLAGS=

files:

compile: $(OBJS)
        @-$(DOTARGS)
```


Copyright © 2004 NEC Corporation

```
cleanup:
	$(REMOVE) $(OBJS) dpblibunix.o

dpbtimef.o: dpbtimef.c
	$(CC) $(CFLAGS) -DORA_PC $(INCLUDE) -c dpbtimef.c

dpbproc.o: dpbproc.c
	$(CC) $(CFLAGS) -DORA_AUX $(INCLUDE) -c dpbproc.c

dpbwait.o: dpbwait.c
	$(CC) $(CFLAGS) -DORA_AUX $(INCLUDE) -c dpbwait.c

dpbpchk.o: dpbpchk.c
	$(CC) $(CFLAGS) -DORA_AUX $(INCLUDE) -c dpbpchk.c

gettime.o: gettime.c
	$(CC) $(CFLAGS) $(INCLUDE) -c gettime.c

trigger.o: trigger.c

dpblibunix.o: $(OBJS)
	$(LD) -r -o $@ $(OBJS)

c_trans_tux.o: $(CTRANTUX_OBJS)
	$(LD) -r -o $@ $(CTRANTUX_OBJS)
```

Appendix C : Tunable Parameters

The procedure of a performance run

1. Boot up server, clients and RTEs.
2. Set IRQs on the server using irq.sh.
3. Start eight Oracle listeners on the server.
4. Set affinity for these listeners using setaffi.
5. Startup the database using run.ora.
6. Start Apache HTTP Server on the clients using httpd.conf.
7. Start Tuxedo on the clients using ubbconfig.
8. Set priority of Oracle processes using setrrpri.sh.
9. Start the RTEs.

Server Configuration

rr.c

```
#include <stdio.h>
#include <unistd.h>
#include <sched.h>
#include <sys/types.h>

main(int argc, char *argv[])
{
	struct sched_param sp;
	int i;

	if (argc < 4) {
		fprintf(stderr, "usage: %s -p <prio> pid...\n", argv[0]);
		exit(-1);
	}

	if (!strcmp("-p", argv[1])) {
		sp.sched_priority = atoi(argv[2]);
	}

/*	printf("setting priority to: %d\n", sp.sched_priority);*/
	for (i = 3; i < argc; i++) {
		pid_t pid = atoi(argv[i]);
		if (sched_setscheduler(pid, SCHED_RR, &sp) == -1) {
			perror("sched_setscheduler");
			exit(-1);
		}
	}

	exit(0);
}
```

setrrpri.sh

```
#sleep $1
/home/oracle/priority/rr -p 48 $(ps aux | grep ora_ | grep -v grep | awk
'{print $2}')
/home/oracle/priority/rr -p 48 $(ps aux | grep oracletp | grep -v grep |
awk '{print $2}')

# Run lgwr at a higher priority
/home/oracle/priority/rr -p 49 $(ps aux | grep ora_lgwr | grep -v grep |
awk '{print $2}')

sleep 2
date > priset"$1".txt
ps -elf | grep ora* >> priset"$1".txt

sh getproc.sh > getproc"$1".txt
```


Copyright © 2004 NEC Corporation

prealloc.sh

```
#!/bin/ksh

date

sqlplus tpcc/tpcc << !!
alter table hist enable table lock;
alter table ordr enable table lock;
alter table ordl enable table lock;
commit;
exit;
!!

#
# order/orderline table
#
BLK=`sqlplus -S tpcc/tpcc << !!
select sum(blocks) from dba_free_space where tablespace_name='ORDR_0';
quit;
!!`
EXT=`echo $BLK | awk '{printf "%d\n", ($3/6327)-1}'`
E=1
while [ $E -le $EXT ]
do
sqlplus -S tpcc/tpcc << !!
alter cluster ordrcluster_queue allocate extent (instance 10000);
commit;
quit;
!!
E=`expr $E + 1`
done

#
# iorder2 index
#
BLK=`sqlplus -S tpcc/tpcc << !!
select sum(blocks) from dba_free_space where tablespace_name='IORDR2_0';
quit;
!!`
EXT=`echo $BLK | awk '{printf "%d\n", ($3/1920)-2}'`
E=1
while [ $E -le $EXT ]
do
sqlplus -S tpcc/tpcc << !!
alter index iordr2 allocate extent (instance 10000);
commit;
quit;
!!
E=`expr $E + 1`
done

#
# history table
#
BLK=`sqlplus -S tpcc/tpcc << !!
select sum(blocks) from dba_free_space where tablespace_name='HIST_0';
quit;
!!`
EXT=`echo $BLK | awk '{printf "%d\n", ($3/25332)}'`
E=1
while [ $E -le $EXT ]
do
sqlplus -S tpcc/tpcc << !!
alter table hist allocate extent (instance 10000);
commit;
quit;
!!
E=`expr $E + 1`
done


sqlplus tpcc/tpcc << !!
alter table hist disable table lock;
alter table ordr disable table lock;
alter table ordl disable table lock;
commit;
quit;
!!
```

irq.sh

```
#!/bin/sh
echo "r 00000001" > /proc/irq/56/smp_affinity
echo "r 00001000" > /proc/irq/60/smp_affinity
echo "r 10000000" > /proc/irq/64/smp_affinity
echo "r 00000100" > /proc/irq/68/smp_affinity
```


Copyright © 2004 NEC Corporation

```
echo "r 00000010" > /proc/irq/76/smp_affinity
echo "r 00000100" > /proc/irq/84/smp_affinity
echo "r 00000100" > /proc/irq/88/smp_affinity
echo "r 00001000" > /proc/irq/92/smp_affinity
echo "r 00001000" > /proc/irq/96/smp_affinity
echo "r 00000100" > /proc/irq/100/smp_affinity
echo "r 00001000" > /proc/irq/108/smp_affinity
echo "r 00010000" > /proc/irq/116/smp_affinity
echo "r 00100000" > /proc/irq/120/smp_affinity
echo "r 01000000" > /proc/irq/124/smp_affinity
echo "r 10000000" > /proc/irq/128/smp_affinity
echo "r 00010000" > /proc/irq/140/smp_affinity
echo "r 00100000" > /proc/irq/148/smp_affinity
echo "r 01000000" > /proc/irq/152/smp_affinity
echo "r 10000000" > /proc/irq/156/smp_affinity
echo "r 00010000" > /proc/irq/160/smp_affinity
echo "r 00100000" > /proc/irq/164/smp_affinity
echo "r 01000000" > /proc/irq/172/smp_affinity
```

setaffi.c

```
#include <stdio.h>
#include <stdlib.h>
#include <sched.h>
#include <errno.h>

#include <linux/unistd.h>
#include <sys/types.h>

/*
 * provide the proper syscall information if our libc
 * is not yet updated.
 */

#undef __NR_sched_setaffinity
#ifndef __NR_sched_setaffinity

#define __NR_sched_setaffinity     1231
#define __NR_sched_getaffinity     1232

unsigned long
sched_setaffinity(pid_t pid, unsigned int len, unsigned long *user_mask_ptr)
{
	return (unsigned long) syscall(__NR_sched_setaffinity, pid, len, user_mask_ptr);
}

unsigned long
sched_getaffinity(pid_t pid, unsigned int len, unsigned long *user_mask_ptr)
{
	return (unsigned long) syscall(__NR_sched_getaffinity, pid, len, user_mask_ptr);
}

#endif

int main(int argc, char * argv[])
{
		unsigned long new_mask;
		unsigned long junk[10];
		unsigned long cur_mask;
		unsigned long junk2[10];
		unsigned int len = sizeof(new_mask);
		pid_t pid;

		if (argc != 3) {
			printf(" usage: %s <pid> <cpu_mask>\n", argv[0]);
			return -1;
		}

		pid = atol(argv[1]);
		sscanf(argv[2], "%x", &new_mask);

		if (sched_getaffinity(pid, len, &cur_mask) < 0) {
			printf("error: could not get pid %d's affinity. errno=%d\n",
pid,errno);
			return -1;
		}

		printf(" pid %d's old affinity: %08lx\n", pid, cur_mask);

		if (sched_setaffinity(pid, len, &new_mask)) {
			printf("error: could not set pid %d's affinity.\n", pid);
			return -1;
		}

		if (sched_getaffinity(pid, len, &cur_mask) < 0) {
			printf("error: could not get pid %d's affinity.\n", pid);
			return -1;
		}

		printf(" pid %d's new affinity: %08lx\n", pid, cur_mask);
```


Copyright © 2004 NEC Corporation

```
        return 0;
}
```

boot.local

```
#! /bin/sh
#
# Copyright (c) 2002 SuSE Linux AG Nuernberg, Germany.  All rights reserved.
#
# Author: Werner Fink <werner@suse.de>, 1996
#         Burchard Steinbild, 1996
#
# /etc/init.d/boot.local
#
# script with local commands to be executed from init on system startup
#
# Here you should add things, that should happen directly after booting
# before we're going to the first run level.
#

echo "100 100000 120 512" > /proc/sys/kernel/sem
echo 0x8000000000 > /proc/sys/kernel/shmmax
echo 0x20000000 > /proc/sys/kernel/shmall
echo 5242880 > /proc/sys/fs/aio-max-nr
echo 1798 > /proc/sys/vm/nr_hugepages
```

~oracle/.bashrc

```
# .bashrc

# User specific aliases and functions

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

alias ls='ls -aF'
export ORACLE_BASE=/home/oracle
export ORACLE_HOME=$ORACLE_BASE/031014
export ORACLE_SID=tpcc
export
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/rdbms/lib:/lib:/usr
/lib
export PATH=$ORACLE_HOME/bin:$PATH
```

Client Configuration

httpd.conf

```
# this file shows Apache settings that need to be added/modified to
support  TPC-C module

#
# KeepAliveTimeout: Number of seconds to wait for the next request from
the
# same client on the same connection.
# Apache issue: This have to be as long as the TPC-C run if you use
keep-alive user connections from RTE

Timeout 14400

KeepAlive On
KeepAliveTimeout 15000
MaxKeepAliveRequests 15000

HostnameLookups Off
UseCanonicalName Off

# worker MPM
# StartServers: initial number of server processes to start
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept spare
# MaxSpareThreads: maximum number of worker threads which are kept spare
# ThreadsPerChild: constant number of worker threads in each server
process
# MaxRequestsPerChild: maximum number of requests a server process
serves
<IfModule worker.c>
StartServers        270
ServerLimit        300
ThreadLimit         100
MaxClients         13000
MinSpareThreads     25
MaxSpareThreads    13000
ThreadsPerChild     50
MaxRequestsPerChild  0
</IfModule>

<IfModule !mpm_winnt.c>
```


Copyright © 2004 NEC Corporation

```
<IfModule !mpm_netware.c>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
#  . On SCO (ODT 3) use "User nouser" and "Group nogroup".
#  . On HPUX you may not be able to use shared memory as nobody, and the
#    suggested workaround is to create a user www and use that user.
#  NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
#  when the value of (unsigned)Group is above 60000;
#  don't use Group #-1 on these systems!
#
#User nobody
#Group #-1
#  oracle:dba is a user that have access to tuexdo server application


User oracle
Group   dba
</IfModule>
</IfModule>

Listen 8080

#
# LoadModule foo_module modules/mod_foo.so
LoadModule tpcc_module        modules/libmodtpcc.so

#
# tpcc module (ported from TPCC ISAPI extension)
#
<IfModule mod_tpcc.c>
        MaxConnections 13000
        MaximumWarehouses 50000
        Path    /tmp/
         Shm_Path /tmp/
        Log     OFF
        Debug   OFF
</IfModule>

<Location /tpcc.dll>
    SetHandler tpcc
</Location>
```

~root/.bashrc

```
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

export LANG=C
export TUXDIR=/home/bea/tuxedo8.1
export APPDIR=/home/oracle/tpccrun/tmdir
export TUXCONFIG=/home/bea/tuxedo8.1/tuxconfig
export COBCPY=$TUXDIR/cobinclude
export COBOPT="-C ANS85 -C ALIGN=8 -C NOIBMCOMP -C TRUNC=ANSI -C
OSEXT=cbl"
export WEBJAVADIR=$TUXDIR/udataobj/webgui/java
export ORACLE_BASE=/home/oracle
export ORACLE_HOME=$ORACLE_BASE/oracle031027
export
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$TUXDIR/lib:LD_LIBRARY_PATH
export ORACLE_SID=tpcc
export PATH=$ORACLE_HOME/bin:$TUXDIR/bin:$PATH
export LIBPATH=$ORACLE_HOME/lib:$TUXDIR/lib:$LIBPATH
export SHLIB_PATH=$ORACLE_HOME/lib32:TUXDIR/lib:$SHLIB_PATH
export TWO_TASK=tpcc

ulimit -u 30000
```

boot.local

```
#! /bin/sh
#
# Copyright (c) 2002 SuSE Linux AG Nuernberg, Germany.  All rights reserved.
#
# Author: Werner Fink <werner@suse.de>, 1996
#         Burchard Steinbild <feedback@suse.de>, 1996
#
# /etc/init.d/boot.local
#
# script with local commands to be executed from init on system startup
#
# Here you should add things, that should happen directly after booting
# before we're going to the first run level.
```


Copyright © 2004 NEC Corporation

```
#

echo 15000 > /proc/sys/kernel/msgmni
echo "2000 32000 64 128" > /proc/sys/kernel/sem
echo 536870912 > /proc/sys/kernel/shmmax
echo 32767 > /proc/sys/fs/file-max
echo 536870912 > /proc/sys/kernel/shmall
echo 32767 > /proc/sys/kernel/threads-max
echo 8192 > /proc/sys/kernel/shmmni
```

ubbconfig

```
#       Copyright (c) 1994 Unix System Laboratories, Inc.
#       All rights reserved
#
#       THIS IS UNPUBLISHED PROPRIETARY
#       SOURCE CODE OF Unix System Laboratories, Inc.
#       The copyright notice above does not
#       evidence any actual or intended
#       publication of such source code.
#
#ident  "@(#) apps/rpcsimp/ubbconfig     $Revision: 1.1 $"
*RESOURCES
IPCKEY 200002
MAXACCESSERS 15000
MAXGTT 2048
MAXSERVERS 1200
MAXSERVICES 1200
MODEL SHM
MASTER acl02
LDBAL Y
SCANUNIT 15
SANITYSCAN 20
BLOCKTIME 60
DBBLWAIT  4
BBLQUERY 120

*MACHINES
DEFAULT:
"acl02"
LMID=acl02
TUXDIR="/home/bea/tuxedo8.1"
APPDIR="/home/oracle/tpccrun/tmdir"
TUXCONFIG="/home/bea/tuxedo8.1/tuxconfig"
ULOGPFX="/home/bea/tuxedo8.1/ulog"
#UID= 0
#GID= 0


*GROUPS
GROUPGEN1
LMID=acl02
GRPNO=1
OPENINFO=NONE

GROUPGEN2
LMID=acl02
GRPNO=2
OPENINFO=NONE

GROUPGEN3
LMID=acl02
GRPNO=3
OPENINFO=NONE

GROUPGEN4
LMID=acl02
GRPNO=4
OPENINFO=NONE

GROUPGEN5
LMID=acl02
GRPNO=5
OPENINFO=NONE

GROUPGEN6
LMID=acl02
GRPNO=6
OPENINFO=NONE

GROUPGEN7
LMID=acl02
GRPNO=7
OPENINFO=NONE

GROUPGEN8
LMID=acl02
```


Copyright © 2004 NEC Corporation

```
GRPNO=8
OPENINFO=NONE

GROUPGEN9
LMID=acl02
GRPNO=9
OPENINFO=NONE

GROUPGEN10
LMID=acl02
GRPNO=10
OPENINFO=NONE

GROUPGEN11
LMID=acl02
GRPNO=11
OPENINFO=NONE

GROUPGEN12
LMID=acl02
GRPNO=12
OPENINFO=NONE

GROUPGEN13
LMID=acl02
GRPNO=13
OPENINFO=NONE

GROUPGEN14
LMID=acl02
GRPNO=14
OPENINFO=NONE

*SERVERS
DEFAULT:
generic SRVGRP=GROUPGEN1
SRVID=100
MIN=1 MAX=1
REPLYQ=Y
RQADDR=OPSTUXSERVER1
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

generic SRVGRP=GROUPGEN2
SRVID=200
MIN=1 MAX=1
REPLYQ=Y
RQADDR=OPSTUXSERVER2
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

generic SRVGRP=GROUPGEN3
SRVID=300
MIN=1 MAX=1
REPLYQ=Y
RQADDR=OPSTUXSERVER3
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

generic SRVGRP=GROUPGEN4
SRVID=400
MIN=1 MAX=1
REPLYQ=Y
RQADDR=OPSTUXSERVER4
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

generic SRVGRP=GROUPGEN5
SRVID=500
MIN=1 MAX=1
REPLYQ=Y
RQADDR=OPSTUXSERVER5
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

generic SRVGRP=GROUPGEN6
SRVID=600
MIN=1 MAX=1
REPLYQ=Y
RQADDR=OPSTUXSERVER6
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

generic SRVGRP=GROUPGEN7
SRVID=700
MIN=1 MAX=1
REPLYQ=Y
RQADDR=OPSTUXSERVER7
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

generic SRVGRP=GROUPGEN8
SRVID=800
MIN=1 MAX=1
REPLYQ=Y
```


Copyright © 2004 NEC Corporation

RQADDR=OPSTUXSERVER8
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

generic SRVGRP=GROUPGEN9
SRVID=900
MIN=1 MAX=1
REPLYQ=Y
RQADDR=OPSTUXSERVER9
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

generic SRVGRP=GROUPGEN10
SRVID=1000
MIN=1 MAX=1
REPLYQ=Y
RQADDR=OPSTUXSERVER10
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

generic SRVGRP=GROUPGEN11
SRVID=1100
MIN=1 MAX=1
REPLYQ=Y
RQADDR=OPSTUXSERVER11
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

generic SRVGRP=GROUPGEN12
SRVID=1200
MIN=1 MAX=1
REPLYQ=Y
RQADDR=OPSTUXSERVER12
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

generic SRVGRP=GROUPGEN13
SRVID=1300
MIN=1 MAX=1
REPLYQ=Y
RQADDR=OPSTUXSERVER13
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

generic SRVGRP=GROUPGEN14
SRVID=1400
MIN=1 MAX=1
REPLYQ=Y
RQADDR=OPSTUXSERVER14
CLOPT="-s OPSTUXSERVER:OPSTUXSERVER"

*SERVICES
#OPSTUXSERVER

RTE input parameter

The following parameters were used with NEC proprietary RTE.

RTE master driver

```
Help message:

        Parameter                                                Default
        ---------------------------------------------------------------
        General test parameters
        -u Ramp Up Time (sec)                                    600
        -s Steady State Time (sec)                               1200
        -d Ramp Down Time (sec)                                  120
        -c Number of Connections                                 10
        -w Number of Warehouses                                  1
        -n C value for NURand                                    66

        Reporting/Logging options
        -R Name of Report File (txt format)                      report.txt
        -X Name of Report File (xlt format)                      report.xlt
        -P Name of NewOrder Trouguput Curve File                 thrput.xlt
        -l Enable Statistics Logging.
        -L Name of Statistics Log File                           slog
        -o Generate 'success' File for Durability Test
        -O Name of 'success' File                                succ

        Misc
        -C Connect Rate of a Client Driver (users/min)           1000
        -S Start Rate of a Client Driver (users/min)             1000
        -v Verbose Mode (experimental)

        Tunable parameters
        -W Comma Separated List of Distribution Weights
                                                                  4480
(NewOrder)
                                                                 4308
(Payment)
                                                                404
(OrderStatus)
                                                               404
(Delivery)
                                                              404
(StockLevel)
        -T Comma Separated List of Think Times (sec)
```


Copyright © 2004 NEC Corporation

```
                                                        12.05
(NewOrder)
                                                       12.05 (Payment)
                                                         10.05
(OrderStatus)
                                                          5.05
(Delivery)
                                                          5.05
(StockLevel)
       -K Comma Separated List of Keying Times (sec)
                                                         18.01
(NewOrder)
                                                        3.01 (Payment)
                                                       2.01
(OrderStatus)
                                                         2.01
(Delivery)
                                                        2.01
(StockLevel)

       -D Web Browser Delay Time (sec)                     0.10

Command-line parameters:
       master -u10800 -s10800 -d300 -c484000 -w48400 -C1500 -S50 -n66 -
T12.07,12.07,10.07,5.07,5.07

During the run, we adjusted the measurement interval dynamically.
Followings are the master.exe console output messages.

[Thu Apr 01 20:58:21] Start test with 484000/484000 user(s)
[Thu Apr 01 20:58:21] Start ramp up (CTRL-C to start M.I. immediately.)
[Thu Apr 01 20:58:21] Ramp up period: 0/10800 secs
[Thu Apr 01 20:58:22] Ramp up period: 1/10800 secs
[Thu Apr 01 20:58:23] Ramp up period: 2/10800 secs
                        :
[Thu Apr 01 22:05:59] Ramp up period: 4058/10800 secs
[Thu Apr 01 22:05:59] Catch CTRL-C event. Go to next phase in 10 secs.
[Thu Apr 01 22:06:00] Sent duration changing message (4068, 10800, 300).
[Thu Apr 01 22:06:00] Ramp up period: 4059/4068 secs
[Thu Apr 01 22:06:01] Ramp up period: 4060/4068 secs
[Thu Apr 01 22:06:02] Ramp up period: 4061/4068 secs
[Thu Apr 01 22:06:03] Ramp up period: 4062/4068 secs
[Thu Apr 01 22:06:04] Ramp up period: 4063/4068 secs
[Thu Apr 01 22:06:05] Ramp up period: 4064/4068 secs
[Thu Apr 01 22:06:06] Ramp up period: 4065/4068 secs
[Thu Apr 01 22:06:07] Ramp up period: 4066/4068 secs
[Thu Apr 01 22:06:08] Ramp up period: 4067/4068 secs
[Thu Apr 01 22:06:09] Start steady state (CTRL-C to stop M.I.
immediately.)
[Thu Apr 01 22:06:09] Steady state period: 0/10800 secs
[Thu Apr 01 22:06:10] Steady state period: 1/10800 secs
[Thu Apr 01 22:06:11] Steady state period: 2/10800 secs
[Thu Apr 01 22:06:12] Steady state period: 3/10800 secs
[Thu Apr 01 22:06:13] Steady state period: 4/10800 secs
[Thu Apr 01 22:06:14] Steady state period: 5/10800 secs
[Thu Apr 01 22:06:15] Steady state period: 6/10800 secs
                        :
[Fri Apr 02 00:06:56] Steady state period: 7247/10800 secs
[Fri Apr 02 00:06:57] Steady state period: 7248/10800 secs
[Fri Apr 02 00:06:58] Steady state period: 7249/10800 secs
[Fri Apr 02 00:06:59] Steady state period: 7250/10800 secs
[Fri Apr 02 00:06:59] Catch CTRL-C event. Go to next phase in 10 secs.
[Fri Apr 02 00:06:59] Sent duration changing message (4068, 7260, 300).
[Fri Apr 02 00:07:00] Steady state period: 7251/7260 secs
[Fri Apr 02 00:07:01] Steady state period: 7252/7260 secs
[Fri Apr 02 00:07:02] Steady state period: 7253/7260 secs
[Fri Apr 02 00:07:03] Steady state period: 7254/7260 secs
[Fri Apr 02 00:07:04] Steady state period: 7255/7260 secs
[Fri Apr 02 00:07:05] Steady state period: 7256/7260 secs
[Fri Apr 02 00:07:06] Steady state period: 7257/7260 secs
[Fri Apr 02 00:07:07] Steady state period: 7258/7260 secs
[Fri Apr 02 00:07:08] Steady state period: 7259/7260 secs
[Fri Apr 02 00:07:09] Start ramp down (CTRL-C to stop test immediately.)
[Fri Apr 02 00:07:09] Ramp down period: 0/300 secs
[Fri Apr 02 00:07:10] Ramp down period: 1/300 secs
[Fri Apr 02 00:07:11] Ramp down period: 2/300 secs
[Fri Apr 02 00:07:12] Ramp down period: 3/300 secs
[Fri Apr 02 00:07:13] Ramp down period: 4/300 secs
[Fri Apr 02 00:07:14] Ramp down period: 5/300 secs
[Fri Apr 02 00:07:15] Ramp down period: 6/300 secs
[Fri Apr 02 00:07:16] Ramp down period: 7/300 secs
                        :
[Fri Apr 02 00:12:07] Ramp down period: 298/300 secs
[Fri Apr 02 00:12:08] Ramp down period: 299/300 secs
[Fri Apr 02 00:12:09] Waiting for 400 client driver(s) to finish.
[Fri Apr 02 00:12:12] Waiting for 363 client driver(s) to finish.
[Fri Apr 02 00:12:15] Waiting for 323 client driver(s) to finish.
[Fri Apr 02 00:12:18] Waiting for 283 client driver(s) to finish.
[Fri Apr 02 00:12:21] Waiting for 246 client driver(s) to finish.
[Fri Apr 02 00:12:24] Waiting for 210 client driver(s) to finish.
[Fri Apr 02 00:12:27] Waiting for 170 client driver(s) to finish.
[Fri Apr 02 00:12:30] Waiting for 129 client driver(s) to finish.
[Fri Apr 02 00:12:33] Waiting for 89 client driver(s) to finish.
[Fri Apr 02 00:12:36] Waiting for 47 client driver(s) to finish.
[Fri Apr 02 00:12:39] Waiting for 6 client driver(s) to finish.
[Fri Apr 02 00:12:42] Test complete. See report.txt for details.
```

RTE slave driver

```
Help message:

       Parameter                        Default
       -------------------------------------
       -M Master Driver Servername      .
       -I HTTP Servername
       -p HTTP Server Port              80
       -c Number of Connections         10
       -o Starting User ID              1
       -d Disable HTTP Keep-Alive
```


Copyright © 2004 NEC Corporation

Following is the script file used to launch all 484,000 users
(= 1210 users * 10 slave drivers * 40 RTE machines)

```
if %COMPUTERNAME%==RTE02 (
start cmd /k c:\client7 -Mbmaster -Iacl02 -o1 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl022 -o1211 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl02 -o2421 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl022 -o3631 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl02 -o4841 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl022 -o6051 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl02 -o7261 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl022 -o8471 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl02 -o9681 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl022 -o10891 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE03 (
start cmd /k c:\client7 -Mbmaster -Iacl03 -o12101 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl032 -o13311 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl03 -o14521 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl032 -o15731 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl03 -o16941 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl032 -o18151 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl03 -o19361 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl032 -o20571 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl03 -o21781 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl032 -o22991 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE04 (
start cmd /k c:\client7 -Mbmaster -Iacl04 -o24201 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl042 -o25411 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl04 -o26621 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl042 -o27831 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl04 -o29041 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl042 -o30251 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl04 -o31461 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl042 -o32671 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl04 -o33881 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl042 -o35091 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE05 (
start cmd /k c:\client7 -Mbmaster -Iacl05 -o36301 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl052 -o37511 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl05 -o38721 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl052 -o39931 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl05 -o41141 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl052 -o42351 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl05 -o43561 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl052 -o44771 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl05 -o45981 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl052 -o47191 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE06 (
start cmd /k c:\client7 -Mbmaster -Iacl06 -o48401 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl062 -o49611 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl06 -o50821 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl062 -o52031 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl06 -o53241 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl062 -o54451 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl06 -o55661 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl062 -o56871 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl06 -o58081 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl062 -o59291 -c1210 -p8080
```


Copyright © 2004 NEC Corporation

```
sleep 60
)
if %COMPUTERNAME%==RTE07 (
start cmd /k c:\client7 -Mbmaster -Iacl07 -o60501 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl072 -o61711 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl07 -o62921 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl072 -o64131 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl07 -o65341 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl072 -o66551 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl07 -o67761 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl072 -o68971 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl07 -o70181 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl072 -o71391 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE08 (
start cmd /k c:\client7 -Mbmaster -Iacl08 -o72601 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl082 -o73811 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl08 -o75021 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl082 -o76231 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl08 -o77441 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl082 -o78651 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl08 -o79861 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl082 -o81071 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl08 -o82281 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl082 -o83491 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE10 (
start cmd /k c:\client7 -Mbmaster -Iacl10 -o84701 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl102 -o85911 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl10 -o87121 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl102 -o88331 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl10 -o89541 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl102 -o90751 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl10 -o91961 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl102 -o93171 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl10 -o94381 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl102 -o95591 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE11 (
start cmd /k c:\client7 -Mbmaster -Iacl11 -o96801 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl112 -o98011 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl11 -o99221 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl112 -o100431 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl11 -o101641 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl112 -o102851 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl11 -o104061 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl112 -o105271 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl11 -o106481 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl112 -o107691 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE12 (
start cmd /k c:\client7 -Mbmaster -Iacl12 -o108901 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl122 -o110111 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl12 -o111321 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl122 -o112531 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl12 -o113741 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl122 -o114951 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl12 -o116161 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl122 -o117371 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl12 -o118581 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl122 -o119791 -c1210 -p8080
sleep 60
)
```


Copyright © 2004 NEC Corporation

```
if %COMPUTERNAME%==RTE13 (
start cmd /k c:\client7 -Mbmaster -Iacl13 -o121001 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl132 -o122211 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl13 -o123421 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl132 -o124631 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl13 -o125841 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl132 -o127051 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl13 -o128261 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl132 -o129471 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl13 -o130681 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl132 -o131891 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE14 (
start cmd /k c:\client7 -Mbmaster -Iacl14 -o133101 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl142 -o134311 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl14 -o135521 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl142 -o136731 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl14 -o137941 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl142 -o139151 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl14 -o140361 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl142 -o141571 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl14 -o142781 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl142 -o143991 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE15 (
start cmd /k c:\client7 -Mbmaster -Iacl15 -o145201 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl152 -o146411 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl15 -o147621 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl152 -o148831 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl15 -o150041 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl152 -o151251 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl15 -o152461 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl152 -o153671 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl15 -o154881 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl152 -o156091 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE16 (
start cmd /k c:\client7 -Mbmaster -Iacl16 -o157301 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl162 -o158511 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl16 -o159721 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl162 -o160931 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl16 -o162141 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl162 -o163351 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl16 -o164561 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl162 -o165771 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl16 -o166981 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl162 -o168191 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE17 (
start cmd /k c:\client7 -Mbmaster -Iacl17 -o169401 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl172 -o170611 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl17 -o171821 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl172 -o173031 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl17 -o174241 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl172 -o175451 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl17 -o176661 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl172 -o177871 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl17 -o179081 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl172 -o180291 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE18 (
start cmd /k c:\client7 -Mbmaster -Iacl18 -o181501 -c1210 -p8080
```


Copyright © 2004 NEC Corporation

```
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl182 -o182711 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl18 -o183921 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl182 -o185131 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl18 -o186341 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl182 -o187551 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl18 -o188761 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl182 -o189971 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl18 -o191181 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl182 -o192391 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE19 (
start cmd /k c:\client7 -Mbmaster -Iacl19 -o193601 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl192 -o194811 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl19 -o196021 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl192 -o197231 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl19 -o198441 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl192 -o199651 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl19 -o200861 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl192 -o202071 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl19 -o203281 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl192 -o204491 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE20 (
start cmd /k c:\client7 -Mbmaster -Iacl20 -o205701 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl202 -o206911 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl20 -o208121 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl202 -o209331 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl20 -o210541 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl202 -o211751 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl20 -o212961 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl202 -o214171 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl20 -o215381 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl202 -o216591 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE21 (
start cmd /k c:\client7 -Mbmaster -Iacl21 -o217801 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl212 -o219011 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl21 -o220221 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl212 -o221431 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl21 -o222641 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl212 -o223851 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl21 -o225061 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl212 -o226271 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl21 -o227481 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl212 -o228691 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE22 (
start cmd /k c:\client7 -Mbmaster -Iacl22 -o229901 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl222 -o231111 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl22 -o232321 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl222 -o233531 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl22 -o234741 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl222 -o235951 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl22 -o237161 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl222 -o238371 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl22 -o239581 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl222 -o240791 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE23 (
start cmd /k c:\client7 -Mbmaster -Iacl23 -o242001 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl232 -o243211 -c1210 -p8080
```


Copyright © 2004 NEC Corporation

```
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl23 -o244421 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl232 -o245631 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl23 -o246841 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl232 -o248051 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl23 -o249261 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl232 -o250471 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl23 -o251681 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl232 -o252891 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE24 (
start cmd /k c:\client7 -Mbmaster -Iacl24 -o254101 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl242 -o255311 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl24 -o256521 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl242 -o257731 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl24 -o258941 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl242 -o260151 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl24 -o261361 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl242 -o262571 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl24 -o263781 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl242 -o264991 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE25 (
start cmd /k c:\client7 -Mbmaster -Iacl25 -o266201 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl252 -o267411 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl25 -o268621 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl252 -o269831 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl25 -o271041 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl252 -o272251 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl25 -o273461 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl252 -o274671 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl25 -o275881 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl252 -o277091 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE33 (
start cmd /k c:\client7 -Mbmaster -Iacl33 -o278301 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl332 -o279511 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl33 -o280721 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl332 -o281931 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl33 -o283141 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl332 -o284351 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl33 -o285561 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl332 -o286771 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl33 -o287981 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl332 -o289191 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE34 (
start cmd /k c:\client7 -Mbmaster -Iacl34 -o290401 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl342 -o291611 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl34 -o292821 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl342 -o294031 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl34 -o295241 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl342 -o296451 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl34 -o297661 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl342 -o298871 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl34 -o300081 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl342 -o301291 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE35 (
start cmd /k c:\client7 -Mbmaster -Iacl35 -o302501 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl352 -o303711 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl35 -o304921 -c1210 -p8080
```


Copyright © 2004 NEC Corporation

```
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl352 -o306131 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl35 -o307341 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl352 -o308551 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl35 -o309761 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl352 -o310971 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl35 -o312181 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl352 -o313391 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE36 (
start cmd /k c:\client7 -Mbmaster -Iacl36 -o314601 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl362 -o315811 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl36 -o317021 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl362 -o318231 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl36 -o319441 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl362 -o320651 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl36 -o321861 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl362 -o323071 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl36 -o324281 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl362 -o325491 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE38 (
start cmd /k c:\client7 -Mbmaster -Iacl38 -o326701 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl382 -o327911 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl38 -o329121 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl382 -o330331 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl38 -o331541 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl382 -o332751 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl38 -o333961 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl382 -o335171 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl38 -o336381 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl382 -o337591 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE39 (
start cmd /k c:\client7 -Mbmaster -Iacl39 -o338801 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl392 -o340011 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl39 -o341221 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl392 -o342431 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl39 -o343641 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl392 -o344851 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl39 -o346061 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl392 -o347271 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl39 -o348481 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl392 -o349691 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE40 (
start cmd /k c:\client7 -Mbmaster -Iacl40 -o350901 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl402 -o352111 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl40 -o353321 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl402 -o354531 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl40 -o355741 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl402 -o356951 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl40 -o358161 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl402 -o359371 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl40 -o360581 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl402 -o361791 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE41 (
start cmd /k c:\client7 -Mbmaster -Iacl41 -o363001 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl412 -o364211 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl41 -o365421 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl412 -o366631 -c1210 -p8080
```


Copyright © 2004 NEC Corporation

```
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl41 -o367841 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl412 -o369051 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl41 -o370261 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl412 -o371471 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl41 -o372681 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl412 -o373891 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE42 (
start cmd /k c:\client7 -Mbmaster -Iacl42 -o375101 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl422 -o376311 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl42 -o377521 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl422 -o378731 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl42 -o379941 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl422 -o381151 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl42 -o382361 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl422 -o383571 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl42 -o384781 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl422 -o385991 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE43 (
start cmd /k c:\client7 -Mbmaster -Iacl43 -o387201 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl432 -o388411 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl43 -o389621 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl432 -o390831 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl43 -o392041 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl432 -o393251 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl43 -o394461 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl432 -o395671 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl43 -o396881 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl432 -o398091 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE44 (
start cmd /k c:\client7 -Mbmaster -Iacl44 -o399301 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl442 -o400511 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl44 -o401721 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl442 -o402931 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl44 -o404141 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl442 -o405351 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl44 -o406561 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl442 -o407771 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl44 -o408981 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl442 -o410191 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE45 (
start cmd /k c:\client7 -Mbmaster -Iacl45 -o411401 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl452 -o412611 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl45 -o413821 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl452 -o415031 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl45 -o416241 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl452 -o417451 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl45 -o418661 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl452 -o419871 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl45 -o421081 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl452 -o422291 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE46 (
start cmd /k c:\client7 -Mbmaster -Iacl46 -o423501 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl462 -o424711 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl46 -o425921 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl462 -o427131 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl46 -o428341 -c1210 -p8080
```


Copyright © 2004 NEC Corporation

```
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl462 -o429551 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl46 -o430761 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl462 -o431971 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl46 -o433181 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl462 -o434391 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE47 (
start cmd /k c:\client7 -Mbmaster -Iacl47 -o435601 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl472 -o436811 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl47 -o438021 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl472 -o439231 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl47 -o440441 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl472 -o441651 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl47 -o442861 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl472 -o444071 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl47 -o445281 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl472 -o446491 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE48 (
start cmd /k c:\client7 -Mbmaster -Iacl48 -o447701 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl482 -o448911 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl48 -o450121 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl482 -o451331 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl48 -o452541 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl482 -o453751 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl48 -o454961 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl482 -o456171 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl48 -o457381 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl482 -o458591 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE49 (
start cmd /k c:\client7 -Mbmaster -Iacl49 -o459801 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl492 -o461011 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl49 -o462221 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl492 -o463431 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl49 -o464641 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl492 -o465851 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl49 -o467061 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl492 -o468271 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl49 -o469481 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl492 -o470691 -c1210 -p8080
sleep 60
)
if %COMPUTERNAME%==RTE50 (
start cmd /k c:\client7 -Mbmaster -Iacl50 -o471901 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl502 -o473111 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl50 -o474321 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl502 -o475531 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl50 -o476741 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl502 -o477951 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl50 -o479161 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl502 -o480371 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl50 -o481581 -c1210 -p8080
sleep 60
start cmd /k c:\client7 -Mbmaster -Iacl502 -o482791 -c1210 -p8080
sleep 60
)
```


Copyright © 2004 NEC Corporation

Appendix D : Space Calculation

60 Day Space

TPM	609467.8
Warehouses	48400

SEGMENT	TYPE	BLOCK_SIZE	BLOCKS	FIVE_PCT	DAILY_GROW	**BLOCKS (MB)**	**FIVE_PCT (MB)**	**DAILY_GROW (MB)**
CUSTCLUSTER	CLUSTER	2048	667424940	33371247	0	1,303,564	65,178	0
DB_STAT	SYS	2048	1048576	0	0	2,048	0	0
DISTCLUSTER	CLUSTER	2048	526336	26317	0	1,028	51	0
HIST	TABLE	4096	22951698	0	4,476,263	89,655	0	17,485
ICUST1	INDEX	2048	18370268	918513	0	35,879	1,794	0
ICUST2	INDEX	2048	41056229	2052811	0	80,188	4,009	0
IDIST	INDEX	2048	6549	327	0	13	1	0
IITEM	INDEX	2048	945	47	0	2	0	0
IORDR2	INDEX	16384	3285120	164256	0	51,330	2,567	0
ISTOK	INDEX	2048	55366407	2768320	0	108,138	5,407	0
ITEMCLUSTER	CLUSTER	2048	14336	717	0	28	1	0
IWARE	INDEX	2048	510	26	0	1	0	0
NORDCLUSTER_QUEUE	CLUSTER	2048	5779460	288973	0	11,288	564	0
ORDRCLUSTER_QUEUE	CLUSTER	16384	81219880	0	15,840,289	1,269,061	0	247,505
ROLL_SEG	SYS	8192	3657728	0	0	28,576	0	0
STOKCLUSTER	CLUSTER	2048	1001055360	50052768	0	1,955,186	97,759	0
SYSAUX	SYS	2048	61440	0	0	120	0	0
SYSTEM	SYS	2048	102400	0	0	200	0	0
SYS_IQ0000009394$$	INDEX	2048	1733838	86692	0	3,386	169	0
SYS_IQ0000009434$$	INDEX	16384	360696	18035	0	5,636	282	0
WARECLUSTER	CLUSTER	2048	55296	2765	0	108	5	0
Total						4945435.09	177788.7715	264989.909

Dynamic space (MB)	1358715.70	Initial MB for (Hist+Ordrcluster)
Static space (MB)	3764508.17	
Free space (MB)	264989.91	
Daily growth (MB)	264989.91	
Daily spread (MB)	0.00	Oracle may be configured such that daily spread is 0
60-day space (MB)	19663902.71	
60-day space (GB)	19203.03	
Log used (KB)	534153329.88	
Log used/N_O txn. (KB)	4.91	Log used per New-Order transactions
8-hour log (GB)	1368.76	

os,file,swap	16.95

	Disk size(GB)	Priced Qty	Priced(GB)	Needed(GB)	Extra(GB)
Database, sys	33.259	1080	35919.72	19203.03	16716.69
	33.91	1	33.91		
Mirrored log	33.258	98	3259.28	2737.53	521.76

Copyright © 2004 NEC Corporation

Appendix E : Price Quotation

Date: Mon, 05 Apr 2004 10:02:56 -0700
From: MaryBeth Pierantoni <mary.beth.pierantoni@oracle.com>
Reply-To: mary.beth.pierantoni@oracle.com
Organization: Server Technologies
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.4) Gecko/20030624 Netscape/7.1 (ax)
X-Accept-Language: en-us, en
MIME-Version: 1.0
To: k-aoyagi@cp.jp.nec.com
Subject: Pricing

Product	Price	Qty	Extended Price
Oracle Database 10g Enterprise Edition for 3 years, Per Processor, Unlimited Users	$20,000	32	$640,000
Oracle Database Server Support Package for 3 years	$6,000	1	$6,000
Oracle Mandatory E-Business Discount			<$129,200>
Oracle TOTAL			$516,800

Contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081

Copyright © 2004 NEC Corporation

THE ECOMMERCE TRANSACTION PLATFORM

March 31 ,2004

Mr. Manabu Miyazaki
Computers Division, NEC Corporation
1-10, Nisshin-chou, Fuchu
Tokyo, 183-8501, Japan

Per your request I am enclosing the pricing information regarding TUXEDO 8.1 that you requested. This pricing applies to Tuxedo 6.4, 6.5, 7.1,8.0 and 8.1. Please note that Tuxedo 8.1 is our most recent version of Tuxedo. Core functionality services (CFS)-R pricing is appropriate for your activities. NEC systems are classified as either a Tier 1, 2, 3, 4 or 5 systems depending on the performance and CPU capacity of the system. The NEC Express5800/120Rf-2are Tier 1 machines – price is $1,200 per server (License), eligible for a 5% discount = $1,140 per server + $252 per server (7x24) for support – support is non discountable. This quote is valid for 60 days from the date of this letter.

Tuxedo Core Functionality Services (CFS-R) Program Product Pricing and Description

TUX-CFS-R 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-R 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 6.4, 6.5, 7.1,8.0, and 8.1. Prices range from $1,200 for Tier 1 to $100,000 for Tier 5. Under this pricing option EVERY system running TUX-CFS-R at the user site must have a TUXEDO license installed and pay the appropriate per server license fees.

Very Truly Yours,

Robert J. Gieringer

Rob Gieringer,
Worldwide Pricing Manager

1

Copyright © 2004 NEC Corporation

BEA Tux/CFS-R Unlimited User License Fees Per Server

Unlimited User License fees per server	Number of Users	Dollar Amount	Maintenance (5 x 9) 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	Unlimited	$1,200.00	$216	$252
Tier 2 - PC Servers with 3 or 4 CPUs, Midrange RISC Uni-processor servers and workstations with up to 2 CPUs	Unlimited	$4,800.00	$864	$1,008
Tier 3 - Midrange Multiprocessors, up to 8 CPUs per system capacity	Unlimited	$12,000.00	$2,160	$2,520
Tier 4 - Large (more than 8, less than 32 CPUs)	Unlimited	$40,000.00	$7,200	$8,400
Tier 5 - Massively Parallel Systems, > 32 processors	Unlimited	$100,000.00	$18,000	$21,000

Copyright © 2004 NEC Corporation

SUSE LINUX AG
Maxfeldstr. 5
D-90409 Nuernberg
Germany

suse@suse.com
www.suse.com
dirk.huebner@suse.com

T: +49 (0) 911 – 740 53 0
F: +49 (0) 911 – 740 53 482

SUSE LINUX AG | Maxfeldstr. 5 | D-90409 Nuernberg

Koichi Aoyagi
NEC Corporation
4-14-22 Shibaura, Minato-ku
Tokyo, 1088558 Japan

Your sign	Our Sign	Customer No	Offer No	Date
			NBG-NEC-20040303	1. April 2004

SUSE LINUX Enterprise Server 9 for Itanium Processors – NEC 32 Processor Server
SUSE LINUX Standard Server 8 – NEC 2x Xeon – 40 Systems

Per your request for pricing information for SUSE Linux Enterprise Server 9 for Itanium Processors and SUSE Linux Standard Server 8 to be used in your TPC-C benchmark testing, SUSE is pleased to provide the following quote. All prices are shown in USD excl. VAT.

Partnumber	Description	Units	Unit Price	Extended Price
N/A*	3 years Maintenance for SUSE LINUX Enterprise Server 9 for Itanium Processors (per CPU)	32	$246.68	$7,893.76
N/A*	3 years Premium Support for SUSE LINUX Enterprise Server 9 for Itanium Processors (per CPU)	32	$369.56	$11,825.92
2133-MFJ-3Y-2-VPO	3 years Maintenance for SUSE LINUX Standard Server 8	40	$704.15	$28,166.00
2133-S-ST-3Y-2-VPO	3 years Premium Support for SUSE LINUX Standard Server 8	40	$1,053.98	$42,159.20
			Total	$90,044.88

(22% discount applied to complete order based on the total dollar volume of the order.)

*Please note that these products have not been released yet and will be available for purchase on September 1st, 2004, at which time the part numbers will have been determined. The products above will be available for purchase at *www.suse.com* or US sales at 1-888-875-4689 or contact suse@suse.com

SUSE Linux AG

i.V. D. Nadt

page 1/1

SUSE LINUX AG | Vorstand: Richard Seibt (Vorsitzender), Amnon Harman | Vorsitzender des Aufsichtsrates: Dr. Heinrich J. Rodewig
HRB Nuernberg 15904 | Ust-IdNr.: DE 192 157 791 | Bankverbindung: HypoVereinsbank AG, Nuernberg, BLZ 760 200 70, Konto 1560 318 320

Copyright © 2004 NEC Corporation

CDW The Right Technology. Right Away.™

800 835 4239

SHOPPING CART

Your Saved Carts | Save This Cart | Edit Saved Carts | Send To An Associate

Continue to Checkout

Click ⊖ to remove an item from your cart

Quantity	Product	CDW	Usually Ships	Price	Ext. Price
8	Allied Telesyn AT-FS716	191222	Same Day	$78.99	$631.92
3	Intel PRO/100+ Adapter 20 pack	263768	4-6 Days	$549.87	$1,649.61
5	Linksys Gigabit 8-Port Workgroup Switch	481324	1-3 Days	$159.36	$796.80
6	NEC AccuSync 500 15" CRT white	573757	Same Day	$129.64	$777.84
22	QLogic SANblade 2340 2Gb 133MHz PCI-X Single Channel HBA	408081	Same Day	$1,189.48	$26,168.56
54	Tripp Lite CAT5e RJ-45M to RJ-45M molded 25' patch cable	324516	Same Day	$7.32	$395.28
88	Tripp Lite CAT5e RJ-45M to RJ45M Cross-Over 10' cable gray	324527	Same Day	$6.48	$570.24
				Sub-Total	$30,990.25

Click ⊖ to remove an item from your cart

Update | Clear Cart | Continue to Checkout

Continue Shopping | **Go to CDW.com Homepage**

Shipping Calc:

Enter a postal code to quickly estimate shipping cost.

QuickCart:

Enter a **CDW part number** to quickly add it to your cart.

* **Sample:** CDW Part #

Ususally Ships:	Same Day
CDW Part:	XXXXXX
Mfg. Part:	XXXXXX-XXXXXX
UNSPSC:	XXXXXXXX

Copyright © 2004 NEC Corporation