



TPC BenchmarkTMC

Full Disclosure Report

*Fujitsu
PRIMEQUEST 580 c/s
W/ 96 Front-Ends*

running

*Oracle Database 10g Release 2
Enterprise Edition with Partitioning*

October 30, 2007

First Edition - October 2007

The benchmark results contained in this document were submitted for compliance with version 5.9 of the TPC Benchmark C Standard Specification. The result of that action is to place these benchmark results into the sixty day "under review" status as of October 2007.

Fujitsu 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. Fujitsu 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, Fujitsu 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. Fujitsu 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 (C) 2007 Fujitsu Limited. 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, October 2007.

Fujitsu and PRIMEQUEST are trademarks or registered trademarks of Fujitsu Limited.

PRIMERGY is a registered trademark of Fujitsu-Siemens Computers GmbH.

ORACLE, SQL*DBA, SQL*Loader, SQL*net, SQL*Plus, Oracle10g, Pro*c and PL/SQL are trademarks of Oracle Corporation.

Intel, Pentium, XEON and Itanium2 are trademarks or registered trademarks of Intel Corporation.


Linux is a registered trademarks of Linus Torvalds.

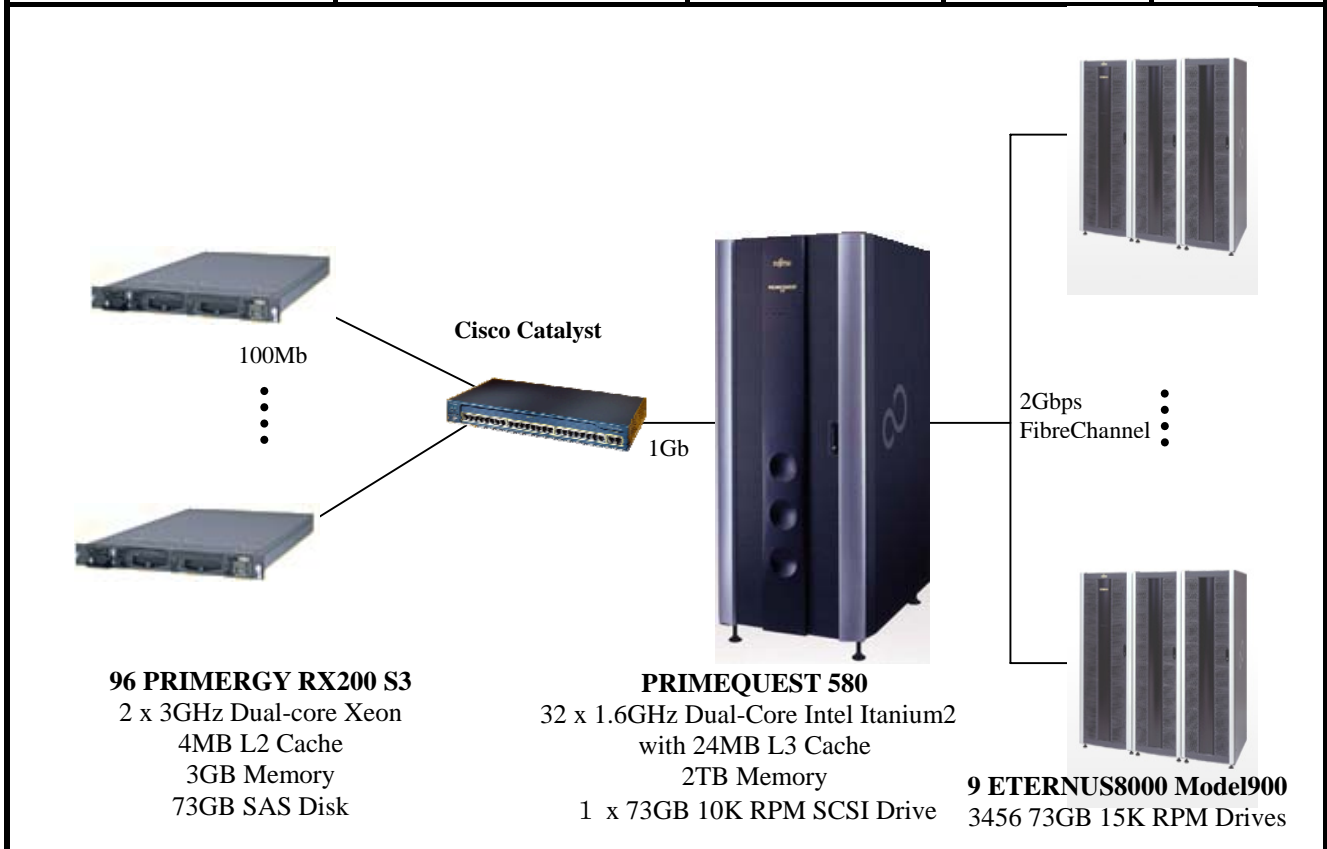
Red Hat is a registered trademarks of Red Hat, Inc.

BEA and Tuxedo are registered trademarks of BEA System, Inc.

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

All other brand or product names mentioned herein are trademarks or registered trademarks of their respective owners.

	PRIMEQUEST 580 c/s w/96 Front-Ends		TPC-C Rev 5.9	
			Report Date: October 30, 2007	
Total System Cost	TPC-C Throughput	Price/Performance	Availability Date	
\$10,306,995 USD	2,196,268 tpmC	\$4.70USD/tpmC	April 30, 2008	
Database Server Processors/Cores/Threads	Database Manager	Operating system	Other Software	Number of users
32/64/128 Intel Itanium 2 1.6GHz	Oracle Database 10g Release 2 Enterprise Edition with Partitioning	Red Hat Enterprise Linux 4 AS	BEA Tuxedo 8.1	1,747,200



System Component	Qty	Server:	Qty	Each of 96 Clients:
Processors/Cores/Threads	32/64/128	1.6GHz Dual-Core Intel Itanium2 9050	2/4/4	3GHz Intel Dual-core Xeon
Cache Memory		24MB L3 Cache		4MB L2 Cache
Memory	64	32GB (4 x 8GB DDR2-400)	3	1GB(2 x 512MB PC2-5300F) 2GB(2 x 1024MB PC2-5300F)
Disk Controllers	36	4G bps FibreChannel (used at 2Gbps)	1	Serial Attached SCSI (SAS)
Disk Drives	1 3456	73GB 10K rpm 73GB 15K rpm	1	73 GB 15K rpm
Total Storage		252,361 GB		7,008 GB



**PRIMEQUEST 580 c/s
w/96 Front-Ends**

TPC-C Rev 5.9
Report Date: October 30, 2007

Description	Part Number	Third Party Brand	Pricing	Unit Price	Qty	Extended Price	3Year Maint. Price
Server Hardware							
PRIMEQUEST 580 Base Unit	MC5B0P211U			168,000.00	1	168,000.00	79,368.00
System Board	MC-87SB11			18,000.00	8	144,000.00	
CPU Module(Dual core Itanium2 9050/1.6GHz/24MB L3/533MHz)	MC-01EA11			37,920.00	32	1,213,440.00	250,368.00
32GB Memory Module (4x8GB DDR2-400)	MC-02A611			77,440.00	64	4,956,160.00	
I/O Unit	MC-87UX11			15,000.00	7	105,000.00	
BMC Module	MC-87BM11			1,720.00	1	1,720.00	
Disk Drive Unit (3.5inch, 73GB, 10,000rpm, Ultra320)	MC-03D321			680.00	1	680.00	
Gigabit Switch Board (w/ 8 external 1000Base-T ports)	MC-87GE11			11,850.00	2	23,700.00	
Additional Power Supply	MC-57PS21U			7,600.00	1	7,600.00	
External I/O Cabinet	MC-87RK11U			20,850.00	1	20,850.00	
PCI-Box	MC-07PB21U			20,700.00	1	20,700.00	
PCI Unit	MC-07PU21			5,170.00	3	15,510.00	
PCI Unit Cable (5m)	MC-07CA11			890.00	3	2,670.00	
FibreChannel Card (4Gbps, PCI-X, dual port)	MC-08FC41			4,270.00	36	153,720.00	
FibreChannel Cable (15m, LC-LC)	CBL-MLLB15			330.00	72	23,760.00	
Flat Panel Display	MC-07FL41			4,830.00	1	4,830.00	
USB Keyboard	MC-07KB11			50.00	1	50.00	
USB Mouse	MC-07MU21			26.00	1	26.00	
Server Hardware Subtotals						6,862,416.00	329,736.00
Storage							
ETERNUS8000 Model900 Base Unit w/ 2 Controllers, 4 Drive Enclosures	E890S20A			200,000.00	9	1,800,000.00	1,033,452.00
Additional Expansion Rack	E800CR1			12,000.00	18	216,000.00	
Additional Controllers (2 sets)	E800CJ01			40,000.00	9	360,000.00	
Cache Memory (2GBx4)	E800CM41			25,793.00	9	232,137.00	
FibreChannel Host Interfaces (4Gbps, dual port, 2 sets)	E800CH14			12,800.00	36	460,800.00	
Drive Enclosures for Base Unit (4 sets)	E800CE11			39,400.00	9	354,600.00	
Drive Enclosures for Expansion Rack (4 sets)	E800CE21			31,000.00	54	1,674,000.00	
Disk Drive Unit (73GB, 15,000rpm)	E800CA3			746.00	3456	2,578,176.00	
Storage Subtotals						7,675,713.00	1,033,452.00
Server Software							
Red Hat Enterprise Linux 4 AS (for Intel Itanium)	MCT0738US			7,497.00	1	7,497.00	5,130.00
Oracle Database 10g Enterprise Edition, Unlimited Users, Per Processor, 3 years				20,000.00	32*	640,000.00	
Partitioning, Unlimited Users, Per Processor, 3 years				5,000.00	32*	160,000.00	
Oracle Database Server Support Package for 3 years				2,000.00	3	6,000.00	
Server Software Subtotals						807,497.00	11,130.00
Client Hardware							
PRIMERGY RX200 S3 (Xeon 5160, 2x1000Base-T)	S26361-K995-V261			3,032.00	96	291,072.00	52,800.00
CPU Module (Xeon 5160 3GHz)	S26361-F3323-E300			1,479.00	96	141,984.00	
1GB Base Memory (2 x 512MB PC2-5300F)	S26361-F3230-B521			5.00	96	480.00	
2GB Memory Module (2 x 1024MB PC-5300F)	S26361-F3230-E522			365.00	96	35,040.00	
Drive Bay (2 x 3.5" HDD slots)	S26361-F3317-E200			44.00	96	4,224.00	
Hard Disk Drive (3.5inch, 73GB, 10,000rpm, SAS 3Gbps)	S26361-F3204-E173			264.00	96	25,344.00	
Internal CD-RW/ DVD Unit	S26361-F3123-E1			68.40	96	6,566.40	
USB FDD	S26391-F264-L226			58.80	1	58.80	
19inch Rack (24U)	S26361-K826-V102			1,591.00	5	7,955.00	
LCD/KB/Pointing Device Unit (1U)	S26361-K1023-V200			1,821.60	5	9,108.00	
KVM Switch (sports, 1U)	S26361-F2293-E801			708.00	15	10,620.00	
KVM Cable (1.8m)	S26361-F2293-L20			10.80	106	1,144.80	
Client Hardware Subtotals						533,597.00	52,800.00



**PRIMEQUEST 580 c/s
w/96 Front-Ends**

TPC-C Rev 5.9
Report Date: October 30, 2007

Description	Part Number	Third Party Brand	Pricing	Unit Price	Qty	Extended Price	3Year Maint. Price
Client Software							
Red Hat Enterprise Linux 4 ES (for x86)	S26361-F2346-E212	1		1,368.00	96	131,328.00	
BEA TUXEDO 8.1 CFS-R (for RHEL4 x86)		3		1,080.00	96	103,680.00	72,576.00
Client Software Subtotals						235,008.00	72,576.00
User Connectivity							
Cisco Catalyst 2950T-24 Switch		4		799.00	5	3,995.00	
Cisco SMARTnet 24x7x4 Maintenance		4			5		1,635.00
User Connectivity Subtotals						3,995.00	1,635.00
Total						16,118,226.00	1,501,329.00
Oracle Mandatory E-Business Discount		2				(161,200.00)	
Large Configuration Discount and Support Prepayment*		1				(6,742,404.90)	(408,956.40)
Total						9,214,622.00	1,092,373.00
Three-Year Cost of Ownership							\$10,306,995

Pricing Sources: 1 = Fujitsu , 2 = Oracle , 3 = BEA , 4 = Computer Online
 Audited by: Francois Raab, InfoSizing, Inc. (www.sizing.com)
 Oracle Corp. pricing contact: MariBeth Pierantoni, mary.beth.pierantoni@oracle.com,
 916-315-5081
 * 32=0.50 x 64. Explanation: For the purposes of counting the number of processors which require licensing, an Intel multicore chip with "n" cores shall be determined by multiplying "n" cores by a factor of 0.50.

Three-Year Cost of Ownership USD	\$10,306,995
tpmC	2,196,268
\$ USD / tpmC	\$4.70

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

**Numerical Quantities Summary for
PRIMEQUEST 580 c/s w/ 96 Front-Ends**

Oracle Database 10g Release 2 Enterprise Edition with Partitioning

MQTH, Computed Maximum Qualified Throughput

2,196,268 tpmC

Response Times (in seconds)	Average	90th %	Maximum
New-Order	0.352	0.344	38.972
Payment	0.337	0.333	38.966
Order-Status	0.344	0.341	38.946
Delivery (interactive portion)	0.103	0.104	0.325
Delivery (deferred portion)	0.243	0.242	38.849
Stock-Level	0.330	0.325	38.922
Menu	0.103	0.104	0.363

Transaction Mix, in percent of total transaction

New-Order	44.94%
Payment	43.02%
Order-Status	4.02%
Delivery(interactive)	4.01%
Stock-Level	4.01%

Emulation Delay (in seconds)

	Response Time	Menu
New-Order	0.1	0.1
Payment	0.1	0.1
Order-Status	0.1	0.1
Delivery (interactive)	0.1	0.1
Stock-Level	0.1	0.1

Keying/Think Times (in seconds)

	Keying Time			Think Time		
	Min	Avg	Max	Min	Avg	Max
New-Order	18.002	18.012	18.035	0.000	12.016	120.201
Payment	3.002	3.012	3.032	0.000	12.016	120.200
Order-Status	2.003	2.012	2.031	0.000	10.019	100.198
Delivery (interactive)	2.002	2.012	2.031	0.000	5.019	50.199
Stock-Level	2.003	2.012	2.029	0.000	5.019	50.198

Test Duration

Ramp-up time	129 Min. 0 Sec.
Measurement interval	120 minutes
Transactions during measurement interval(all types)	586,739,828

Checkpointing

Number of checkpoints	4
Checkpoint interval (seconds)	1,754

Table Of Contents

PREFACE

GENERAL ITEMS	4
0.1 APPLICATION CODE AND DEFINITION STATEMENTS	4
0.2 TEST SPONSOR.....	4
0.3 PARAMETER SETTINGS.....	4
0.4 CONFIGURATION DIAGRAMS.....	5
CLAUSE 1: LOGICAL DATA BASE DESIGN RELATED ITEMS	7
1.1 TABLE DEFINITIONS	7
1.2 PHYSICAL ORGANIZATION OF DATABASE	7
1.3 INSERT AND DELETE OPERATIONS	7
1.4 PARTITIONING.....	7
1.5 REPLICATION, DUPLICATION OR ADDITIONS	8
CLAUSE 2: TRANSACTION AND TERMINAL PROFILES RELATED ITEMS	9
2.1 RANDOM NUMBER GENERATION	9
2.2 INPUT/OUTPUT SCREEN LAYOUT.....	9
2.3 PRICED TERMINAL FEATURE VERIFICATION	9
2.4 PRESENTATION MANAGER OR INTELLIGENT TERMINAL	9
2.5 TRANSACTION PROFILES	10
2.6 QUEUING MECHANISM.....	10
CLAUSE 3: TRANSACTION AND SYSTEM PROPERTIES RELATED ITEMS	11
3.1 TRANSACTION SYSTEM PROPERTIES (ACID).....	11
3.2 ATOMICITY.....	11
3.2.1 Completed Transactions.....	11
3.2.2 Aborted Transactions	12
3.3 CONSISTENCY	12
3.4 ISOLATION	12
3.5 DURABILITY.....	13
3.5.1 Loss of Log Disk	13
3.5.2 Loss of Data Disk	13
3.5.3 Instantaneous Interruption and Loss of Memory.....	14
CLAUSE 4: SCALING AND DATA BASE POPULATION RELATED ITEMS	15
4.1 INITIAL CARDINALITY OF TABLES.....	15
4.2 DATABASE LAYOUT	16
4.3 TYPE OF DATABASE.....	43
4.4 DATABASE MAPPING	43
4.5 60 DAY SPACE.....	43
CLAUSE 5: PERFORMANCE METRICS AND RESPONSE TIME RELATED ITEMS	44
5.1 THROUGHPUT.....	44
5.2 RESPONSE TIMES.....	44

5.3	KEYING AND THINK TIMES	45
5.4	RESPONSE TIME FREQUENCY DISTRIBUTION CURVES AND OTHER GRAPHS	45
5.5	STEADY STATE DETERMINATION	50
5.6	WORK PERFORMED DURING STEADY STATE	50
5.7	REPRODUCIBILITY	51
5.8	MEASUREMENT PERIOD DURATION	51
5.9	REGULATION OF TRANSACTION MIX	51
5.10	TRANSACTION STATISTICS	52
5.11	CHECKPOINT COUNT AND LOCATION.....	52
CLAUSE 6: SUT, DRIVER, AND COMMUNICATION DEFINITION RELATED ITEMS		53
6.1	RTE DESCRIPTIONS.....	53
6.2	LOSS OF TERMINAL CONNECTIONS.....	53
6.3	EMULATED COMPONENTS.....	53
6.4	FUNCTIONAL DIAGRAMS.....	53
6.5	NETWORKS.....	54
6.6	OPERATOR INTERVENTION	54
CLAUSE 7: PRICING RELATED ITEMS		55
7.1	HARDWARE AND SOFTWARE COMPONENTS.....	55
7.2	AVAILABILITY	55
7.3	THROUGHPUT AND PRICE PERFORMANCE.....	55
7.4	COUNTRY SPECIFIC PRICING	56
7.5	USAGE PRICING	56
7.6	SYSTEM PRICING	56
CLAUSE 9: AUDIT RELATED ITEMS		57
9.1	AUDITOR'S REPORT	57
9.2	AVAILABILITY OF THE FULL DISCLOSURE REPORT.....	57
APPENDIX A: CLIENT SOURCE CODE		58
APPENDIX B: SERVER SOURCE CODE.....		100
APPENDIX C: RTE SCRIPTS.....		145
APPENDIX D: SYSTEM TUNABLES.....		174
APPENDIX E: DATABASE CREATION CODE		266
APPENDIX F: 60 DAY SPACE CALCULATION.....		304
APPENDIX G: NUMERICAL QUANTITIES SUMMARY PER CLIENT		305
APPENDIX H: PRICE QUOTES		310
APPENDIX I: AUDITOR'S ATTESTATION LETTER.....		316

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C test conducted by Fujitsu Ltd. on the Fujitsu PRIMEQUEST 580 w/96 Front-Ends. The operating system and the DBMS used on the server were Red Hat Enterprise Linux 4 AS for Itanium Processor Family and Oracle Database 10g Release 2 Enterprise Edition with Partitioning.

The operating system on the clients was Red Hat Enterprise Linux 4 ES for x86.

Those clients ran Apache HTTP Server and BEA Tuxedo 8.1 CFS-R.

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), and the availability date are reported as:

2,196,268 tpmC
\$4.70 USD/tpmC
April 30, 2008

Standard and Executive Summary Statements

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

Auditor

The benchmark configuration, environment and methodology, along with the pricing model used to calculate the cost per tpmC, were audited by Francois Raab of InfoSizing to verify compliance with the relevant TPC specifications.

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 5.9.

TPC Benchmark C Overview

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

TPC Benchmark C is an On Line Transaction Processing (OLTP) workload. It is a mixture of read-only and update intensive transactions that simulate the activities found in complex OLTP application environments. It does so by exercising a breadth of system components associated with 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 of data access and update

The performance metric reported by TPC-C is a “business throughput” measuring the number of orders processed per minute. Multiple transactions are used to simulate the business activity of processing an order, and each transaction is subject to a

response time constraint. The performance metric for this benchmark is expressed in transactions-per-minute-C (tpmC). To be compliant with the TPC-C standard, all references to tpmC results must include the tpmC rate, the associated price-per-tpmC, and the availability date of the priced configuration.

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

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

General Items

0.1 Application Code and Definition Statements

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

Appendix A and B contain all source codes implemented in this benchmark.

0.2 Test Sponsor

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

Fujitsu and Oracle Corp. were joint sponsors of this TPC Benchmark C.

0.3 Parameter Settings

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

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

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

Appendix D contains the parameters for the database, the operating system, and the

configuration for the transaction monitor.

0.4 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/cores/threads.*
- *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 unit, including their protocol type.*
- *Number and LAN (e.g., Ethernet) connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure (see Clause 8.1.8).*
- *Type and the run-time execution location of software components (e.g., DBMS, client processes, transaction monitors, software drivers, etc.).*

The System Under Test (SUT), a PRIMEQUEST 580 c/s w/96 Front-Ends, is depicted in the following diagrams.

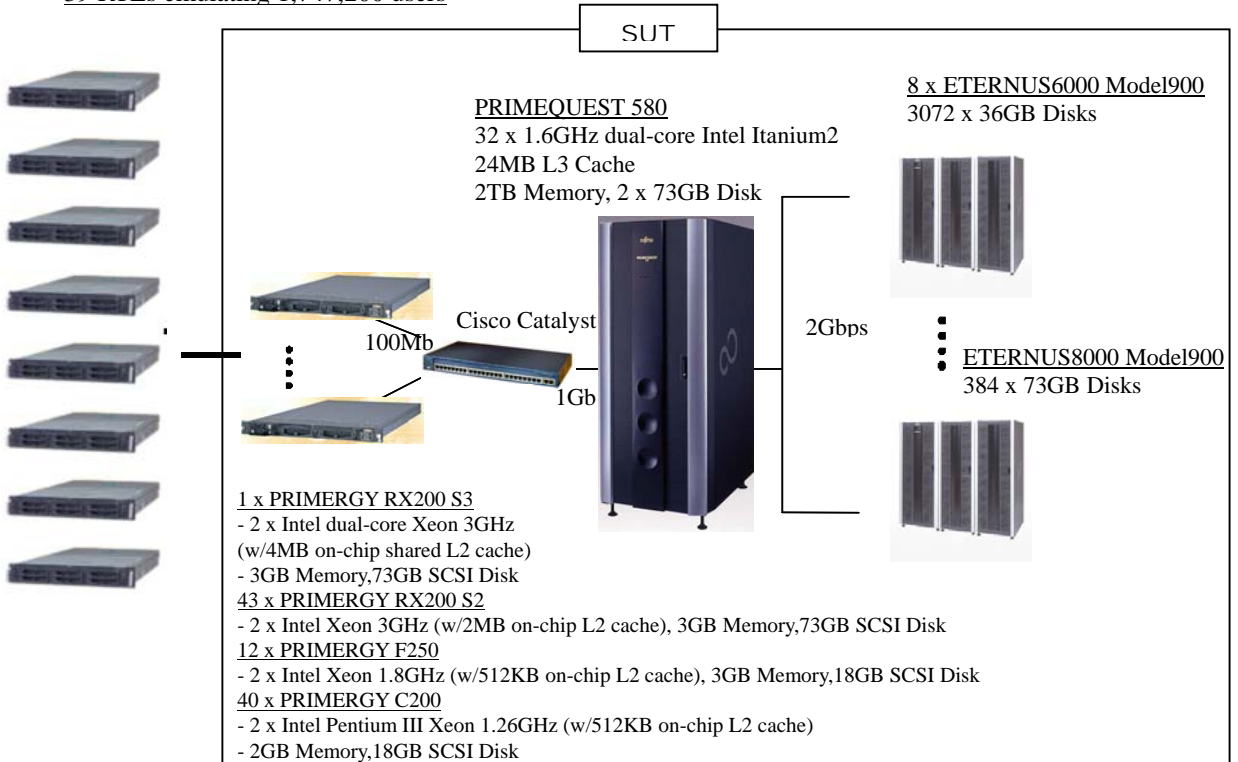
The configuration diagrams for both the tested and priced systems are included on the following pages.

There were differences between the priced and measured configurations. The differences are:

- A RTE was used in the tested configuration.
- The clients other than the PRIMERGY RX200 S3 were replaced by the PRIMERGY RX200 S3 in the priced configuration.
- The ETERNUS6000 Storage subsystems were replaced by the ETERNUS8000 Storage subsystems in the priced configuration.

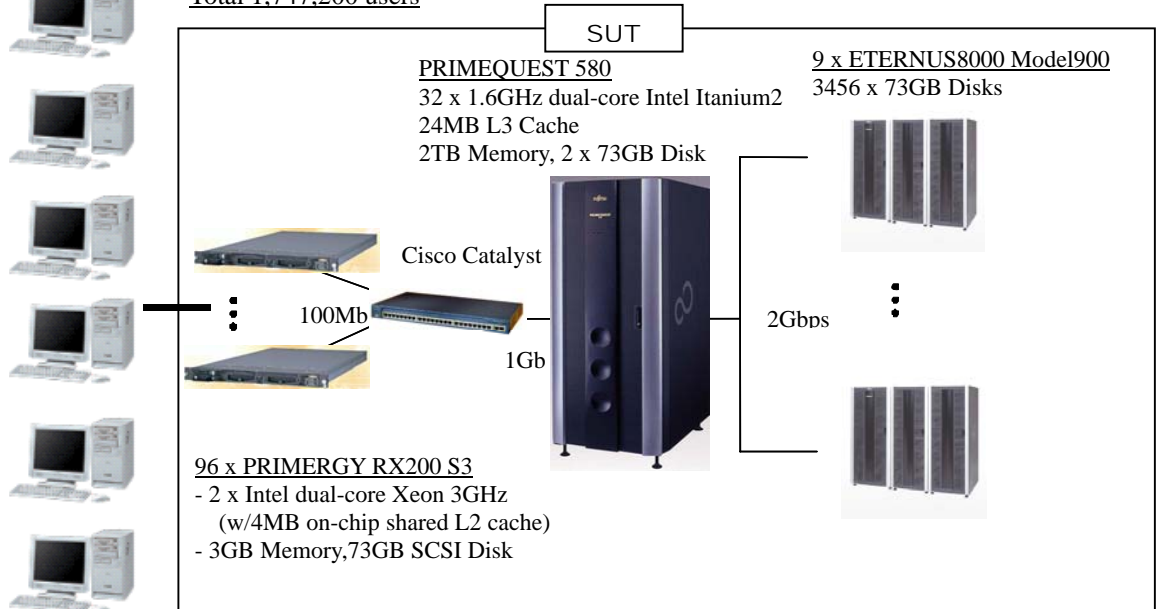
PRIMEQUEST 580 Tested Configuration

39 RTEs emulating 1,747,200 users



PRIMEQUEST 580 Priced Configuration

Total 1,747,200 users



Clause 1 Related Items

1.1 Table Definitions

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

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

1.2 Physical Organization of Database

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

Physical space was allocated to Oracle Database 10g Release2 Enterprise Edition with Partitioning on the server disks according to the details provided in section 4.2. The size of the space segments on each disk was calculated to provide even distribution of data across the disk drives.

1.3 Insert and Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restrictions 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 verified and fully operational during the entire benchmark.

1.4 Partitioning

While there are a few restrictions placed upon horizontal or vertical partitioning of

tables and rows in the TPC-C benchmark, any such partitioning must be disclosed. Replication of tables, if used, must be disclosed. Additional and/or duplicated attributes in any table must be disclosed along with a statement on the impact on performance.

Horizontal partitioning was used for one (hist) of the tables and one (iordr2) of the indices. The detail of this partitioning can be understood by examining the table and index definition statements in Appendix E. Vertical partitioning and additional or duplicated attributes were not used in this implementation.

1.5 Replication, Duplication or Additions

Replication of tables, if used, must be disclosed(see Clause 1.4.6). 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 replications, duplications or additional attributes were used in this benchmark.

Clause 2 Related Items

2.1 Random Number Generation

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

The seeds for each user were generated using the terminal id and the unix time of measurement start, which was given by the RTE master process. The terminal id is unique number across all RTE emulated users. Since the seeds were incremented by the same start value, they were also unique across all users.

2.2 Input/Output Screen Layout

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

All screen layouts followed the specification exactly.

2.3 Priced Terminal Feature Verification

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

The terminal attributes were verified by the auditor manually exercising each specification during the onsite audit portion of this benchmark.

2.4 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 forms applications is listed in Appendix A.

2.5 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 backs as a results of an unused item number must be disclosed.

The number of items per orders entered by New-Order transactions must be disclosed. The percentage of home and remote Payment transactions must be disclosed. The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the database must be disclosed.

The percentage of Delivery transactions that were skipped as a result of an insufficient number of rows in the NEW-ORDER table must be disclosed. The mix (i.e., percentages) of transaction types seen by the SUT must be disclosed.

Table 2.1 lists the numerical quantities that Clauses 8.1.3.5 to 8.1.3.11 require.

Table 2.1 Transaction Statistics

Statistic		Value
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse	85.00%
	Remote warehouse	15.00%
	Accessed by last name	60.00%
Order Status	Accessed by last name	59.99%
Delivery	Skipped transactions	None
Transaction Mix	New Order	44.94%
	Payment	43.02%
	Order status	4.02%
	Delivery	4.01%
	Stock level	4.01%

2.6 Queuing Mechanism

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

Delivery transactions were submitted to servers using the same mechanism that other transactions used, Tuxedo API. The only difference was that `tpacall()` was used instead of `tpcall()` to call the server process asynchronously, i.e., control would return to the client thread immediately and the deferred delivery part would complete asynchronously in the server process.

Clause 3 Related Items

3.1 Transaction System Properties (ACID)

The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met. This includes disclosing which case was followed for the execution of Isolation Test 7.

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

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

3.2 Atomicity

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

3.2.1 Completed Transactions

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

A row was randomly selected from the warehouse, district and customer tables, and the balances noted. A payment transaction was started with the same warehouse, district and customer identifiers and a known amount. The payment transaction was

committed and the rows were verified to contain correctly updated balances.

3.2.2 Aborted Transactions

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

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

3.3 Consistency

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

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

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

These consistency conditions were tested using a shell script to issue queries to the database. The results of the queries verified that the database was consistent for all four tests.

A performance run was completed including a full 120 minutes of steady state and checkpoints.

The shell script was executed again. The result of the same queries verified that the database remained consistent after the run.

3.4 Isolation

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

The benchmark specification defines nine required tests to be performed to demonstrate that the required levels of transaction isolation are met. These tests, described in Clauses 3.4.2.1 - 3.4.2.9, were all performed and verified as required.

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 by the auditor to demonstrate the required isolation had been met.

For Isolation test seven, case D was followed.

3.5 Durability

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

3.5.1 Loss of Log Disk

To demonstrate recovery from a permanent failure of durable media containing the Oracle recovery log data. The following steps were executed using 174,375 warehouses of the database:

1. The database was backed up to extra disks.
2. The total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
3. The RTE was started with 1,743,750 users.
4. The test was allowed to run for a minimum of 5 minutes after the system reached the steady state.
5. One of the log disks was removed from the cabinet to cause a log disk failure. Since the log was configured as RAID0+1, the transactions continued to run without interruption.
6. After the run for another minimum of 5 minutes, the RTE was finished successfully.
7. Step 2 was repeated and the difference between the first and second counts was noted.
8. The success file was used to determine the number of NEW_ORDERS successfully returned to the RTE.
9. The counts in step 7 and 8 were compared, verifying that all committed transactions were successfully recovered.
10. Data from the success file was used to query the database to demonstrate that successful transactions had corresponding rows in the ORDER table and that rolled back transactions did not.

This demonstration was executed under the configuration with the Oracle recovery log laid on one ETERNUS8000 storage subsystem and with all Oracle TPC-C tables on 8 ETERNUS6000 storage subsystems. A disk on the ETERNUS8000 storage was removed for the failure.

3.5.2 Loss of Data Disk

To demonstrate recovery from a permanent failure of durable media containing the Oracle TPC-C tables, the following steps were executed using 174,720 warehouses of the database:

1. The database was backed up to extra disks.
2. The total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
3. The RTE was started with 1,747,200 users.
4. The test was allowed to run for a minimum of 5 minutes after the system reached the steady state.
5. A disk array failure was caused by removing a disk from the disk array cabinet.
6. The RTE was shut down.
7. Oracle was shutdown abort.
8. New disks were returned into the disk cabinet to recover the RAID system.
9. Data from the backup disks was restored.
10. Oracle was restarted and the media recovery utility started.
11. Step 2 was repeated and the difference between the first and second counts was noted.
12. The success file was used to determine the number of NEW_ORDERS successfully returned to the RTE.

13. The counts in step 11 and 12 were compared, verifying that all committed transactions were successfully recovered.
14. Data from the success file was used to query the database to demonstrate that successful transactions had corresponding rows in the ORDER table and that rolled back transactions did not.

This demonstration was executed under the configuration with the Oracle recovery log laid on one ETERNUS6000 storage subsystem and with the Oracle TPC-C tables on 7 ETERNUS6000 and 1 ETERNUS8000 storage subsystems. A disk on the ETERNUS8000 storage was removed for the failure.

3.5.3 Instantaneous Interruption and Loss of Memory

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

This test was executed on a fully scaled database of 174,720 warehouses under a full load of 1,747,200 users. The following steps were executed:

1. The total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
2. The RTE was started with 1,747,200 users.
3. The test was allowed to run for a minimum of 5 minutes after the system reached the steady state.
4. The primary power to the server was shutdown.
5. Power was restored and the system performed an automatic recovery.
6. Oracle was restarted and performed an automatic recovery.
7. Step 1 was repeated and the difference between the first and second counts was noted.
8. The success file was used to determine the number of NEW-ORDERS successfully returned to the RTE.
9. The counts in step 7 and 8 were compared, verifying that all committed transactions had been successfully recovered.
10. Data from the success file was used to query the database to demonstrate successful transactions had corresponding rows in the ORDER table, and rolled back transactions did not.

This demonstration was executed under the configuration with Oracle recovery log laid on one ETERNUS6000 storage subsystem and with the Oracle TPC-C tables on 7 ETERNUS6000 and 1 ETERNUS8000 storage subsystem.

Clause 4 Related Items

4.1 Initial Cardinality of Tables

The cardinality (e.g. 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 initially configured with 174,720 warehouses.

Table 4.1 Number of Rows for Server

Table	Occurrences
Warehouse	174,720
District	1,747,200
Customer	5,241,600,000
History	5,241,600,000
Order	5,241,600,000
New Order	1,572,480,000
Order Line	52,417,988,736
Stock	17,472,000,000
Item	100,000

4.2 Database Layout

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

The following description depicts the data base configuration of the system tested.

1) Data files

We used 8 storage subsystems for data files. Each storage had 4 controllers and 8 FibreChannel ports (4 @ dual-port) to connect to the Database server. It had 384 disk drives, which were configured into 24 RAID0 volumes having 16 disks in each group.

2 logical volumes were allocated on each RAID0 volume for this benchmark test. One of them was configured with 11 partitions (Type A) and the other was configured with 5 partitions (Type B) to lay the TPC-C data files.

Type A:

- 4 partitions for stok
- 5 partitions for cust
- 1 partition for ordr
- 1 partition for hist

Type B:

- 1 partition for icust2
- 1 partition for iordr2
- 1 partition for temp
- 1 partition for icust1 or istok1
- 1 partition for nord or other tables and indexes

The software RAID tool was used to create the large volumes. The data files of nord, ordr and hist were on each large volume configured with partitions for them. Total 384 logical volumes (48 logical volumes for each storage) were used for the data files.

2) Log files

The database logs were configured with 16 RAID0+1 volumes. Each RAID0+1 volume consisted of 24 disks (12 disks + 12 disks mirrored). A log file was configured with 16 LUs (1 LU from each RAID0+1 volume) using Linux mdadm software RAID utility to spread accesses across all 16 volumes.

3) Partition type

The following list shows how partitioned the devices (logical volumes) were:

[Type A : the size of each device is 153,600MB]

sdc	sdfg	sdlr	sdrs	sdabx
sdg	sdfk	sdlw	sdrw	sdacb
sdk	sdfo	sdma	sdsa	sdacf
sdo	sdfs	sdme	sdwj	sdacj
sds	sdfw	sdmi	sdwn	sdacn
sdw	sdga	sdmm	sdwr	sdacr
sdaa	sdge	sdmq	sdwv	sdacv
sdae	sdgi	sdmu	sdwz	sdacz
sdai	sdgm	sdmy	sdxl	sdadd
sdam	sdhc	sdnc	sdxh	sdadh

sdaq	sdhg	sdng	sdxl	sdadl
sdau	sdhk	sdnk	sdxp	sdadp
sday	sdho	sdno	sdxt	sdadt
sdbc	sdhs	sdns	sdxs	sdadx
sdbg	sdhw	sdnw	sdyb	sdaeb
sdbk	sdia	sdoa	sdyf	sdaef
sdbo	sdie	sdoe	sdyj	sdaej
sdbb	sdii	sdoi	sdyn	sdaen
sdbw	sdim	sdom	sdyr	sdaer
sdca	sdii	sdoq	sdyv	sdaev
sdce	sdiu	sdou	sdyz	sdaez
sdc	sdiy	sdo	sdzd	sdafd
sdc	sdjc	sdpc	sdzh	sdafh
sdcq	sdjg	sdpg	sdzl	sdaf
sdcu	sdjk	sdpk	sdzp	sdafp
sdey	sdjo	sdpo	sdzt	sdaft
sddc	sdjs	sdps	sdzx	sdafx
sddg	sdjw	sdpw	sdaab	sdagb
sddk	sdka	sdqa	sdaaf	sdagf
sddo	sdke	sdqe	sdaaj	sdagj
sdds	sdki	sdqi	sdaan	sdagn
sddw	sdkm	sdqm	sdaar	sdagr
sdea	sdq	sdqq	sdaav	sdagv
sdee	sdku	sdqu	sdaaz	sdagz
sdei	sdky	sdqy	sdabd	sdahd
sdem	sdlc	sdr	sdabh	sdahh
sdeu	sdlg	sdr	sdabl	
sdey	sdlk	sdrk	sdabp	
sdfc	sdlo	sdro	sdabt	

[Type B : the size of each device is 102,400MB]

sdd	sdfh	sdl	sdr	sdaby
sdh	sdf	sdlx	sdrx	sdacc
sd	sdfp	sdlb	sdrb	sdacg
sdp	sdf	sdlm	sdr	sdack
sdt	sdfx	sdlj	sdr	sdaco
sdx	sdgb	sdlm	sdr	sdacs
sdab	sdgf	sdlr	sdrw	sdacw
sdaf	sdgj	sdlv	sdr	sdada
sdaj	sdgn	sdlz	sdr	sdade
sdan	sdhd	sdl	sdr	sdadi
sdar	sdhh	sdlh	sdr	sdadm
sdav	sdhl	sdl	sdr	sdadq
sdaz	sdhp	sdlp	sdr	sdadu
sdbd	sdht	sdl	sdr	sdady
sdbh	sdhx	sdlx	sdr	sdaec
sdbl	sdib	sdlb	sdr	sdaeg
sdbp	sdif	sdlf	sdr	sdaek
sdbt	sdij	sdlj	sdr	sdaeo
sdbx	sdin	sdl	sdr	sdaes
sdc	sd	sdr	sdr	sdaew
sdcf	sdiv	sdlv	sdr	sdafa
sdcj	sdiz	sdlz	sdr	sdafe
sdcn	sdjd	sdlj	sdr	sdafi
sdc	sdjh	sdlh	sdr	sdafm
sdcv	sdjl	sdlj	sdr	sdafq
sdcz	sdjp	sdlp	sdr	sdafu
sddd	sdjt	sdlj	sdr	sdafy
sddh	sdjx	sdlx	sdr	sdagc

```

sddl   sdkb   sdqb   sdaag  sdagg
sddp   sdkf   sdqf   sdaak  sdagk
sddt   sdkj   sdqj   sdaao  sdago
sddx   sdkn   sdqn   sdaas  sdags
sdeb   sdkr   sdqr   sdaaw  sdagw
sdef   sdkv   sdqv   sdaba  sdaha
sdejs  dkz    sdqz   sdabe  sdahe
sden   sdld   sdrd   sdabi  sdahi
sdev   sdll   sdrh   sdabm
sdez   sdll   sdrl   sdabq
sdfd   sdlp   sdrp   sdabu

```

4) Relation between data files and devices

The following list shows the relation between data files, raw device files and device names:

```

hist_0_96 --> raw2018 --> dm-0 | cust_0_74 --> raw844 --> sdey6
hist_0_97 --> raw2019 --> dm-1 | cust_0_75 --> raw845 --> sdey7
hist_0_106 --> raw2028 --> dm-10 | cust_0_495 --> raw1265 --> sdey8
hist_0_4 --> raw1926 --> dm-100 | cust_0_496 --> raw1266 --> sdey9
hist_0_5 --> raw1927 --> dm-101 | icust2_0_37 --> raw2151 --> sdez1
hist_0_6 --> raw1928 --> dm-102 | iordr2_0_37 --> raw2343 --> sdez2
hist_0_7 --> raw1929 --> dm-103 | temp_0_37 --> raw2535 --> sdez3
hist_0_8 --> raw1930 --> dm-104 | stok_0_152 --> raw153 --> sdfc1
hist_0_9 --> raw1931 --> dm-105 | cust_0_500 --> raw1270 --> sdfc10
hist_0_10 --> raw1932 --> dm-106 | stok_0_153 --> raw154 --> sdfc2
hist_0_11 --> raw1933 --> dm-107 | stok_0_154 --> raw155 --> sdfc3
hist_0_12 --> raw1934 --> dm-108 | stok_0_155 --> raw156 --> sdfc5
hist_0_13 --> raw1935 --> dm-109 | cust_0_76 --> raw846 --> sdfc6
hist_0_107 --> raw2029 --> dm-11 | cust_0_77 --> raw847 --> sdfc7
hist_0_14 --> raw1936 --> dm-110 | cust_0_498 --> raw1268 --> sdfc8
hist_0_15 --> raw1937 --> dm-111 | cust_0_499 --> raw1269 --> sdfc9
hist_0_16 --> raw1938 --> dm-112 | icust2_0_38 --> raw2152 --> sdfd1
hist_0_17 --> raw1939 --> dm-113 | iordr2_0_38 --> raw2344 --> sdfd2
hist_0_18 --> raw1940 --> dm-114 | temp_0_38 --> raw2536 --> sdfd3
hist_0_19 --> raw1941 --> dm-115 | stok_0_156 --> raw157 --> sdfg1
hist_0_20 --> raw1942 --> dm-116 | cust_0_503 --> raw1273 --> sdfg10
hist_0_21 --> raw1943 --> dm-117 | stok_0_157 --> raw158 --> sdfg2
hist_0_22 --> raw1944 --> dm-118 | stok_0_158 --> raw159 --> sdfg3
hist_0_23 --> raw1945 --> dm-119 | stok_0_159 --> raw160 --> sdfg5
hist_0_108 --> raw2030 --> dm-12 | cust_0_78 --> raw848 --> sdfg6
hist_0_24 --> raw1946 --> dm-120 | cust_0_79 --> raw849 --> sdfg7
hist_0_25 --> raw1947 --> dm-121 | cust_0_501 --> raw1271 --> sdfg8
hist_0_26 --> raw1948 --> dm-122 | cust_0_502 --> raw1272 --> sdfg9
hist_0_27 --> raw1949 --> dm-123 | icust2_0_39 --> raw2153 --> sdfh1
hist_0_28 --> raw1950 --> dm-124 | iordr2_0_39 --> raw2345 --> sdfh2
hist_0_29 --> raw1951 --> dm-125 | temp_0_39 --> raw2537 --> sdfh3
hist_0_30 --> raw1952 --> dm-126 | stok_0_160 --> raw161 --> sdfk1
hist_0_31 --> raw1953 --> dm-127 | cust_0_506 --> raw1276 --> sdfk10
hist_0_32 --> raw1954 --> dm-128 | stok_0_161 --> raw162 --> sdfk2
hist_0_33 --> raw1955 --> dm-129 | stok_0_162 --> raw163 --> sdfk3
hist_0_109 --> raw2031 --> dm-13 | stok_0_163 --> raw164 --> sdfk5
hist_0_34 --> raw1956 --> dm-130 | cust_0_80 --> raw850 --> sdfk6
hist_0_35 --> raw1957 --> dm-131 | cust_0_81 --> raw851 --> sdfk7
hist_0_36 --> raw1958 --> dm-132 | cust_0_504 --> raw1274 --> sdfk8
hist_0_37 --> raw1959 --> dm-133 | cust_0_505 --> raw1275 --> sdfk9
hist_0_38 --> raw1960 --> dm-134 | icust2_0_40 --> raw2154 --> sdfl1
hist_0_39 --> raw1961 --> dm-135 | iordr2_0_40 --> raw2346 --> sdfl2
hist_0_40 --> raw1962 --> dm-136 | temp_0_40 --> raw2538 --> sdfl3
hist_0_41 --> raw1963 --> dm-137 | stok_0_164 --> raw165 --> sdfo1
hist_0_42 --> raw1964 --> dm-138 | cust_0_509 --> raw1279 --> sdfo10
hist_0_43 --> raw1965 --> dm-139 | stok_0_165 --> raw166 --> sdfo2

```

hist_0_110	-->	raw2032	-->	dm-14		stok_0_166	-->	raw167	-->	sdfo3
hist_0_44	-->	raw1966	-->	dm-140		stok_0_167	-->	raw168	-->	sdfo5
hist_0_45	-->	raw1967	-->	dm-141		cust_0_82	-->	raw852	-->	sdfo6
hist_0_46	-->	raw1968	-->	dm-142		cust_0_83	-->	raw853	-->	sdfo7
hist_0_47	-->	raw1969	-->	dm-143		cust_0_507	-->	raw1277	-->	sdfo8
hist_0_48	-->	raw1970	-->	dm-144		cust_0_508	-->	raw1278	-->	sdfo9
hist_0_49	-->	raw1971	-->	dm-145		icust2_0_41	-->	raw2155	-->	sdfp1
hist_0_50	-->	raw1972	-->	dm-146		iordr2_0_41	-->	raw2347	-->	sdfp2
hist_0_51	-->	raw1973	-->	dm-147		temp_0_41	-->	raw2539	-->	sdfp3
hist_0_52	-->	raw1974	-->	dm-148		stok_0_168	-->	raw169	-->	sdfs1
hist_0_53	-->	raw1975	-->	dm-149		cust_0_512	-->	raw1282	-->	sdfs10
hist_0_111	-->	raw2033	-->	dm-15		stok_0_169	-->	raw170	-->	sdfs2
hist_0_54	-->	raw1976	-->	dm-150		stok_0_170	-->	raw171	-->	sdfs3
hist_0_55	-->	raw1977	-->	dm-151		stok_0_171	-->	raw172	-->	sdfs5
hist_0_56	-->	raw1978	-->	dm-152		cust_0_84	-->	raw854	-->	sdfs6
hist_0_57	-->	raw1979	-->	dm-153		cust_0_85	-->	raw855	-->	sdfs7
hist_0_58	-->	raw1980	-->	dm-154		cust_0_510	-->	raw1280	-->	sdfs8
hist_0_59	-->	raw1981	-->	dm-155		cust_0_511	-->	raw1281	-->	sdfs9
hist_0_60	-->	raw1982	-->	dm-156		icust2_0_42	-->	raw2156	-->	sdf1
hist_0_61	-->	raw1983	-->	dm-157		iordr2_0_42	-->	raw2348	-->	sdf2
hist_0_62	-->	raw1984	-->	dm-158		temp_0_42	-->	raw2540	-->	sdf3
hist_0_63	-->	raw1985	-->	dm-159		stok_0_172	-->	raw173	-->	sdfw1
hist_0_112	-->	raw2034	-->	dm-16		cust_0_515	-->	raw1285	-->	sdfw10
hist_0_64	-->	raw1986	-->	dm-160		stok_0_173	-->	raw174	-->	sdfw2
hist_0_65	-->	raw1987	-->	dm-161		stok_0_174	-->	raw175	-->	sdfw3
hist_0_66	-->	raw1988	-->	dm-162		stok_0_175	-->	raw176	-->	sdfw5
hist_0_67	-->	raw1989	-->	dm-163		cust_0_86	-->	raw856	-->	sdfw6
hist_0_68	-->	raw1990	-->	dm-164		cust_0_87	-->	raw857	-->	sdfw7
hist_0_69	-->	raw1991	-->	dm-165		cust_0_513	-->	raw1283	-->	sdfw8
hist_0_70	-->	raw1992	-->	dm-166		cust_0_514	-->	raw1284	-->	sdfw9
hist_0_71	-->	raw1993	-->	dm-167		icust2_0_43	-->	raw2157	-->	sdfx1
hist_0_72	-->	raw1994	-->	dm-168		iordr2_0_43	-->	raw2349	-->	sdfx2
hist_0_73	-->	raw1995	-->	dm-169		temp_0_43	-->	raw2541	-->	sdfx3
hist_0_113	-->	raw2035	-->	dm-17		stok_0_4	-->	raw5	-->	sdg1
hist_0_74	-->	raw1996	-->	dm-170		cust_0_389	-->	raw1159	-->	sdg10
hist_0_75	-->	raw1997	-->	dm-171		stok_0_5	-->	raw6	-->	sdg2
hist_0_76	-->	raw1998	-->	dm-172		stok_0_6	-->	raw7	-->	sdg3
hist_0_77	-->	raw1999	-->	dm-173		stok_0_7	-->	raw8	-->	sdg5
hist_0_78	-->	raw2000	-->	dm-174		cust_0_2	-->	raw772	-->	sdg6
hist_0_79	-->	raw2001	-->	dm-175		cust_0_3	-->	raw773	-->	sdg7
hist_0_80	-->	raw2002	-->	dm-176		cust_0_387	-->	raw1157	-->	sdg8
hist_0_81	-->	raw2003	-->	dm-177		cust_0_388	-->	raw1158	-->	sdg9
hist_0_82	-->	raw2004	-->	dm-178		stok_0_176	-->	raw177	-->	sdga1
hist_0_83	-->	raw2005	-->	dm-179		cust_0_518	-->	raw1288	-->	sdga10
hist_0_114	-->	raw2036	-->	dm-18		stok_0_177	-->	raw178	-->	sdga2
hist_0_84	-->	raw2006	-->	dm-180		stok_0_178	-->	raw179	-->	sdga3
hist_0_85	-->	raw2007	-->	dm-181		stok_0_179	-->	raw180	-->	sdga5
hist_0_86	-->	raw2008	-->	dm-182		cust_0_88	-->	raw858	-->	sdga6
hist_0_87	-->	raw2009	-->	dm-183		cust_0_89	-->	raw859	-->	sdga7
hist_0_88	-->	raw2010	-->	dm-184		cust_0_516	-->	raw1286	-->	sdga8
hist_0_89	-->	raw2011	-->	dm-185		cust_0_517	-->	raw1287	-->	sdga9
hist_0_90	-->	raw2012	-->	dm-186		icust2_0_44	-->	raw2158	-->	sdgb1
hist_0_91	-->	raw2013	-->	dm-187		iordr2_0_44	-->	raw2350	-->	sdgb2
hist_0_92	-->	raw2014	-->	dm-188		temp_0_44	-->	raw2542	-->	sdgb3
hist_0_93	-->	raw2015	-->	dm-189		stok_0_180	-->	raw181	-->	sdge1
hist_0_115	-->	raw2037	-->	dm-19		cust_0_521	-->	raw1291	-->	sdge10
hist_0_94	-->	raw2016	-->	dm-190		stok_0_181	-->	raw182	-->	sdge2
hist_0_95	-->	raw2017	-->	dm-191		stok_0_182	-->	raw183	-->	sdge3
ordr_0_96	-->	raw1826	-->	dm-192		stok_0_183	-->	raw184	-->	sdge5
ordr_0_97	-->	raw1827	-->	dm-193		cust_0_90	-->	raw860	-->	sdge6
ordr_0_98	-->	raw1828	-->	dm-194		cust_0_91	-->	raw861	-->	sdge7
ordr_0_99	-->	raw1829	-->	dm-195		cust_0_519	-->	raw1289	-->	sdge8
ordr_0_100	-->	raw1830	-->	dm-196		cust_0_520	-->	raw1290	-->	sdge9
ordr_0_101	-->	raw1831	-->	dm-197		icust2_0_45	-->	raw2159	-->	sdgf1
ordr_0_102	-->	raw1832	-->	dm-198		iordr2_0_45	-->	raw2351	-->	sdgf2

ordr_0_103	-->	raw1833	-->	dm-199		temp_0_45	-->	raw2543	-->	sdgf3
hist_0_98	-->	raw2020	-->	dm-2		stok_0_184	-->	raw185	-->	sdgi1
hist_0_116	-->	raw2038	-->	dm-20		cust_0_524	-->	raw1294	-->	sdgi10
ordr_0_104	-->	raw1834	-->	dm-200		stok_0_185	-->	raw186	-->	sdgi2
ordr_0_105	-->	raw1835	-->	dm-201		stok_0_186	-->	raw187	-->	sdgi3
ordr_0_106	-->	raw1836	-->	dm-202		stok_0_187	-->	raw188	-->	sdgi5
ordr_0_107	-->	raw1837	-->	dm-203		cust_0_92	-->	raw862	-->	sdgi6
ordr_0_108	-->	raw1838	-->	dm-204		cust_0_93	-->	raw863	-->	sdgi7
ordr_0_109	-->	raw1839	-->	dm-205		cust_0_522	-->	raw1292	-->	sdgi8
ordr_0_110	-->	raw1840	-->	dm-206		cust_0_523	-->	raw1293	-->	sdgi9
ordr_0_111	-->	raw1841	-->	dm-207		icust2_0_46	-->	raw2160	-->	sdgj1
ordr_0_112	-->	raw1842	-->	dm-208		iordr2_0_46	-->	raw2352	-->	sdgj2
ordr_0_113	-->	raw1843	-->	dm-209		temp_0_46	-->	raw2544	-->	sdgj3
hist_0_117	-->	raw2039	-->	dm-21		stok_0_188	-->	raw189	-->	sdgm1
ordr_0_114	-->	raw1844	-->	dm-210		cust_0_527	-->	raw1297	-->	sdgm10
ordr_0_115	-->	raw1845	-->	dm-211		stok_0_189	-->	raw190	-->	sdgm2
ordr_0_116	-->	raw1846	-->	dm-212		stok_0_190	-->	raw191	-->	sdgm3
ordr_0_117	-->	raw1847	-->	dm-213		stok_0_191	-->	raw192	-->	sdgm5
ordr_0_118	-->	raw1848	-->	dm-214		cust_0_94	-->	raw864	-->	sdgm6
ordr_0_119	-->	raw1849	-->	dm-215		cust_0_95	-->	raw865	-->	sdgm7
ordr_0_120	-->	raw1850	-->	dm-216		cust_0_525	-->	raw1295	-->	sdgm8
ordr_0_121	-->	raw1851	-->	dm-217		cust_0_526	-->	raw1296	-->	sdgm9
ordr_0_122	-->	raw1852	-->	dm-218		icust2_0_47	-->	raw2161	-->	sdgn1
ordr_0_123	-->	raw1853	-->	dm-219		iordr2_0_47	-->	raw2353	-->	sdgn2
hist_0_118	-->	raw2040	-->	dm-22		temp_0_47	-->	raw2545	-->	sdgn3
ordr_0_124	-->	raw1854	-->	dm-220		icust2_0_1	-->	raw2115	-->	sdh1
ordr_0_125	-->	raw1855	-->	dm-221		iordr2_0_1	-->	raw2307	-->	sdh2
ordr_0_126	-->	raw1856	-->	dm-222		temp_0_1	-->	raw2499	-->	sdh3
ordr_0_127	-->	raw1857	-->	dm-223		icust1_0_1	-->	raw2691	-->	sdh5
ordr_0_128	-->	raw1858	-->	dm-224		stok_0_192	-->	raw193	-->	sdhc1
ordr_0_129	-->	raw1859	-->	dm-225		cust_0_530	-->	raw1300	-->	sdhc10
ordr_0_130	-->	raw1860	-->	dm-226		stok_0_193	-->	raw194	-->	sdhc2
ordr_0_131	-->	raw1861	-->	dm-227		stok_0_194	-->	raw195	-->	sdhc3
ordr_0_132	-->	raw1862	-->	dm-228		stok_0_195	-->	raw196	-->	sdhc5
ordr_0_133	-->	raw1863	-->	dm-229		cust_0_96	-->	raw866	-->	sdhc6
hist_0_119	-->	raw2041	-->	dm-23		cust_0_97	-->	raw867	-->	sdhc7
ordr_0_134	-->	raw1864	-->	dm-230		cust_0_528	-->	raw1298	-->	sdhc8
ordr_0_135	-->	raw1865	-->	dm-231		cust_0_529	-->	raw1299	-->	sdhc9
ordr_0_136	-->	raw1866	-->	dm-232		icust2_0_48	-->	raw2162	-->	sdhd1
ordr_0_137	-->	raw1867	-->	dm-233		iordr2_0_48	-->	raw2354	-->	sdhd2
ordr_0_138	-->	raw1868	-->	dm-234		temp_0_48	-->	raw2546	-->	sdhd3
ordr_0_139	-->	raw1869	-->	dm-235		stok_0_196	-->	raw197	-->	sdhg1
ordr_0_140	-->	raw1870	-->	dm-236		cust_0_533	-->	raw1303	-->	sdhg10
ordr_0_141	-->	raw1871	-->	dm-237		stok_0_197	-->	raw198	-->	sdhg2
ordr_0_142	-->	raw1872	-->	dm-238		stok_0_198	-->	raw199	-->	sdhg3
ordr_0_143	-->	raw1873	-->	dm-239		stok_0_199	-->	raw200	-->	sdhg5
hist_0_120	-->	raw2042	-->	dm-24		cust_0_98	-->	raw868	-->	sdhg6
ordr_0_144	-->	raw1874	-->	dm-240		cust_0_99	-->	raw869	-->	sdhg7
ordr_0_145	-->	raw1875	-->	dm-241		cust_0_531	-->	raw1301	-->	sdhg8
ordr_0_146	-->	raw1876	-->	dm-242		cust_0_532	-->	raw1302	-->	sdhg9
ordr_0_147	-->	raw1877	-->	dm-243		icust2_0_49	-->	raw2163	-->	sdhh1
ordr_0_148	-->	raw1878	-->	dm-244		iordr2_0_49	-->	raw2355	-->	sdhh2
ordr_0_149	-->	raw1879	-->	dm-245		temp_0_49	-->	raw2547	-->	sdhh3
ordr_0_150	-->	raw1880	-->	dm-246		stok_0_200	-->	raw201	-->	sdhk1
ordr_0_151	-->	raw1881	-->	dm-247		cust_0_536	-->	raw1306	-->	sdhk10
ordr_0_152	-->	raw1882	-->	dm-248		stok_0_201	-->	raw202	-->	sdhk2
ordr_0_153	-->	raw1883	-->	dm-249		stok_0_202	-->	raw203	-->	sdhk3
hist_0_121	-->	raw2043	-->	dm-25		stok_0_203	-->	raw204	-->	sdhk5
ordr_0_154	-->	raw1884	-->	dm-250		cust_0_100	-->	raw870	-->	sdhk6
ordr_0_155	-->	raw1885	-->	dm-251		cust_0_101	-->	raw871	-->	sdhk7
ordr_0_156	-->	raw1886	-->	dm-252		cust_0_534	-->	raw1304	-->	sdhk8
ordr_0_157	-->	raw1887	-->	dm-253		cust_0_535	-->	raw1305	-->	sdhk9
ordr_0_158	-->	raw1888	-->	dm-254		icust2_0_50	-->	raw2164	-->	sdhl1
ordr_0_159	-->	raw1889	-->	dm-255		iordr2_0_50	-->	raw2356	-->	sdhl2
ordr_0_160	-->	raw1890	-->	dm-256		temp_0_50	-->	raw2548	-->	sdhl3

ordr_0_161	-->	raw1891	-->	dm-257		stok_0_204	-->	raw205	-->	sdho1
ordr_0_162	-->	raw1892	-->	dm-258		cust_0_539	-->	raw1309	-->	sdho10
ordr_0_163	-->	raw1893	-->	dm-259		stok_0_205	-->	raw206	-->	sdho2
hist_0_122	-->	raw2044	-->	dm-26		stok_0_206	-->	raw207	-->	sdho3
ordr_0_164	-->	raw1894	-->	dm-260		stok_0_207	-->	raw208	-->	sdho5
ordr_0_165	-->	raw1895	-->	dm-261		cust_0_102	-->	raw872	-->	sdho6
ordr_0_166	-->	raw1896	-->	dm-262		cust_0_103	-->	raw873	-->	sdho7
ordr_0_167	-->	raw1897	-->	dm-263		cust_0_537	-->	raw1307	-->	sdho8
ordr_0_168	-->	raw1898	-->	dm-264		cust_0_538	-->	raw1308	-->	sdho9
ordr_0_169	-->	raw1899	-->	dm-265		icust2_0_51	-->	raw2165	-->	sdhp1
ordr_0_170	-->	raw1900	-->	dm-266		iordr2_0_51	-->	raw2357	-->	sdhp2
ordr_0_171	-->	raw1901	-->	dm-267		temp_0_51	-->	raw2549	-->	sdhp3
ordr_0_172	-->	raw1902	-->	dm-268		stok_0_208	-->	raw209	-->	sdhs1
ordr_0_173	-->	raw1903	-->	dm-269		cust_0_542	-->	raw1312	-->	sdhs10
hist_0_123	-->	raw2045	-->	dm-27		stok_0_209	-->	raw210	-->	sdhs2
ordr_0_174	-->	raw1904	-->	dm-270		stok_0_210	-->	raw211	-->	sdhs3
ordr_0_175	-->	raw1905	-->	dm-271		stok_0_211	-->	raw212	-->	sdhs5
ordr_0_176	-->	raw1906	-->	dm-272		cust_0_104	-->	raw874	-->	sdhs6
ordr_0_177	-->	raw1907	-->	dm-273		cust_0_105	-->	raw875	-->	sdhs7
ordr_0_178	-->	raw1908	-->	dm-274		cust_0_540	-->	raw1310	-->	sdhs8
ordr_0_179	-->	raw1909	-->	dm-275		cust_0_541	-->	raw1311	-->	sdhs9
ordr_0_180	-->	raw1910	-->	dm-276		icust2_0_52	-->	raw2166	-->	sdht1
ordr_0_181	-->	raw1911	-->	dm-277		iordr2_0_52	-->	raw2358	-->	sdht2
ordr_0_182	-->	raw1912	-->	dm-278		temp_0_52	-->	raw2550	-->	sdht3
ordr_0_183	-->	raw1913	-->	dm-279		stok_0_212	-->	raw213	-->	sdhw1
hist_0_124	-->	raw2046	-->	dm-28		cust_0_545	-->	raw1315	-->	sdhw10
ordr_0_184	-->	raw1914	-->	dm-280		stok_0_213	-->	raw214	-->	sdhw2
ordr_0_185	-->	raw1915	-->	dm-281		stok_0_214	-->	raw215	-->	sdhw3
ordr_0_186	-->	raw1916	-->	dm-282		stok_0_215	-->	raw216	-->	sdhw5
ordr_0_187	-->	raw1917	-->	dm-283		cust_0_106	-->	raw876	-->	sdhw6
ordr_0_188	-->	raw1918	-->	dm-284		cust_0_107	-->	raw877	-->	sdhw7
ordr_0_189	-->	raw1919	-->	dm-285		cust_0_543	-->	raw1313	-->	sdhw8
ordr_0_190	-->	raw1920	-->	dm-286		cust_0_544	-->	raw1314	-->	sdhw9
ordr_0_191	-->	raw1921	-->	dm-287		icust2_0_53	-->	raw2167	-->	sdhx1
ordr_0_0	-->	raw1730	-->	dm-288		iordr2_0_53	-->	raw2359	-->	sdhx2
ordr_0_1	-->	raw1731	-->	dm-289		temp_0_53	-->	raw2551	-->	sdhx3
hist_0_125	-->	raw2047	-->	dm-29		stok_0_216	-->	raw217	-->	sdia1
ordr_0_2	-->	raw1732	-->	dm-290		cust_0_548	-->	raw1318	-->	sdia10
ordr_0_3	-->	raw1733	-->	dm-291		stok_0_217	-->	raw218	-->	sdia2
ordr_0_4	-->	raw1734	-->	dm-292		stok_0_218	-->	raw219	-->	sdia3
ordr_0_5	-->	raw1735	-->	dm-293		stok_0_219	-->	raw220	-->	sdia5
ordr_0_6	-->	raw1736	-->	dm-294		cust_0_108	-->	raw878	-->	sdia6
ordr_0_7	-->	raw1737	-->	dm-295		cust_0_109	-->	raw879	-->	sdia7
ordr_0_8	-->	raw1738	-->	dm-296		cust_0_546	-->	raw1316	-->	sdia8
ordr_0_9	-->	raw1739	-->	dm-297		cust_0_547	-->	raw1317	-->	sdia9
ordr_0_10	-->	raw1740	-->	dm-298		icust2_0_54	-->	raw2168	-->	sdib1
ordr_0_11	-->	raw1741	-->	dm-299		iordr2_0_54	-->	raw2360	-->	sdib2
hist_0_99	-->	raw2021	-->	dm-3		temp_0_54	-->	raw2552	-->	sdib3
hist_0_126	-->	raw2048	-->	dm-30		stok_0_220	-->	raw221	-->	sdie1
ordr_0_12	-->	raw1742	-->	dm-300		cust_0_551	-->	raw1321	-->	sdie10
ordr_0_13	-->	raw1743	-->	dm-301		stok_0_221	-->	raw222	-->	sdie2
ordr_0_14	-->	raw1744	-->	dm-302		stok_0_222	-->	raw223	-->	sdie3
ordr_0_15	-->	raw1745	-->	dm-303		stok_0_223	-->	raw224	-->	sdie5
ordr_0_16	-->	raw1746	-->	dm-304		cust_0_110	-->	raw880	-->	sdie6
ordr_0_17	-->	raw1747	-->	dm-305		cust_0_111	-->	raw881	-->	sdie7
ordr_0_18	-->	raw1748	-->	dm-306		cust_0_549	-->	raw1319	-->	sdie8
ordr_0_19	-->	raw1749	-->	dm-307		cust_0_550	-->	raw1320	-->	sdie9
ordr_0_20	-->	raw1750	-->	dm-308		icust2_0_55	-->	raw2169	-->	sdif1
ordr_0_21	-->	raw1751	-->	dm-309		iordr2_0_55	-->	raw2361	-->	sdif2
hist_0_127	-->	raw2049	-->	dm-31		temp_0_55	-->	raw2553	-->	sdif3
ordr_0_22	-->	raw1752	-->	dm-310		stok_0_224	-->	raw225	-->	sdii1
ordr_0_23	-->	raw1753	-->	dm-311		cust_0_554	-->	raw1324	-->	sdii10
ordr_0_24	-->	raw1754	-->	dm-312		stok_0_225	-->	raw226	-->	sdii2
ordr_0_25	-->	raw1755	-->	dm-313		stok_0_226	-->	raw227	-->	sdii3
ordr_0_26	-->	raw1756	-->	dm-314		stok_0_227	-->	raw228	-->	sdii5

ordr_0_27	-->	raw1757	-->	dm-315		cust_0_112	-->	raw882	-->	sdiid6
ordr_0_28	-->	raw1758	-->	dm-316		cust_0_113	-->	raw883	-->	sdiid7
ordr_0_29	-->	raw1759	-->	dm-317		cust_0_552	-->	raw1322	-->	sdiid8
ordr_0_30	-->	raw1760	-->	dm-318		cust_0_553	-->	raw1323	-->	sdiid9
ordr_0_31	-->	raw1761	-->	dm-319		icust2_0_56	-->	raw2170	-->	sdij1
hist_0_128	-->	raw2050	-->	dm-32		iordr2_0_56	-->	raw2362	-->	sdij2
ordr_0_32	-->	raw1762	-->	dm-320		temp_0_56	-->	raw2554	-->	sdij3
ordr_0_33	-->	raw1763	-->	dm-321		stok_0_228	-->	raw229	-->	sdim1
ordr_0_34	-->	raw1764	-->	dm-322		cust_0_557	-->	raw1327	-->	sdim10
ordr_0_35	-->	raw1765	-->	dm-323		stok_0_229	-->	raw230	-->	sdim2
ordr_0_36	-->	raw1766	-->	dm-324		stok_0_230	-->	raw231	-->	sdim3
ordr_0_37	-->	raw1767	-->	dm-325		stok_0_231	-->	raw232	-->	sdim5
ordr_0_38	-->	raw1768	-->	dm-326		cust_0_114	-->	raw884	-->	sdim6
ordr_0_39	-->	raw1769	-->	dm-327		cust_0_115	-->	raw885	-->	sdim7
ordr_0_40	-->	raw1770	-->	dm-328		cust_0_555	-->	raw1325	-->	sdim8
ordr_0_41	-->	raw1771	-->	dm-329		cust_0_556	-->	raw1326	-->	sdim9
hist_0_129	-->	raw2051	-->	dm-33		icust2_0_57	-->	raw2171	-->	sdin1
ordr_0_42	-->	raw1772	-->	dm-330		iordr2_0_57	-->	raw2363	-->	sdin2
ordr_0_43	-->	raw1773	-->	dm-331		temp_0_57	-->	raw2555	-->	sdin3
ordr_0_44	-->	raw1774	-->	dm-332		stok_0_232	-->	raw233	-->	sdiiq1
ordr_0_45	-->	raw1775	-->	dm-333		cust_0_560	-->	raw1330	-->	sdiiq10
ordr_0_46	-->	raw1776	-->	dm-334		stok_0_233	-->	raw234	-->	sdiiq2
ordr_0_47	-->	raw1777	-->	dm-335		stok_0_234	-->	raw235	-->	sdiiq3
ordr_0_48	-->	raw1778	-->	dm-336		stok_0_235	-->	raw236	-->	sdiiq5
ordr_0_49	-->	raw1779	-->	dm-337		cust_0_116	-->	raw886	-->	sdiiq6
ordr_0_50	-->	raw1780	-->	dm-338		cust_0_117	-->	raw887	-->	sdiiq7
ordr_0_51	-->	raw1781	-->	dm-339		cust_0_558	-->	raw1328	-->	sdiiq8
hist_0_130	-->	raw2052	-->	dm-34		cust_0_559	-->	raw1329	-->	sdiiq9
ordr_0_52	-->	raw1782	-->	dm-340		icust2_0_58	-->	raw2172	-->	sdir1
ordr_0_53	-->	raw1783	-->	dm-341		iordr2_0_58	-->	raw2364	-->	sdir2
ordr_0_54	-->	raw1784	-->	dm-342		temp_0_58	-->	raw2556	-->	sdir3
ordr_0_55	-->	raw1785	-->	dm-343		stok_0_236	-->	raw237	-->	sdiu1
ordr_0_56	-->	raw1786	-->	dm-344		cust_0_563	-->	raw1333	-->	sdiu10
ordr_0_57	-->	raw1787	-->	dm-345		stok_0_237	-->	raw238	-->	sdiu2
ordr_0_58	-->	raw1788	-->	dm-346		stok_0_238	-->	raw239	-->	sdiu3
ordr_0_59	-->	raw1789	-->	dm-347		stok_0_239	-->	raw240	-->	sdiu5
ordr_0_60	-->	raw1790	-->	dm-348		cust_0_118	-->	raw888	-->	sdiu6
ordr_0_61	-->	raw1791	-->	dm-349		cust_0_119	-->	raw889	-->	sdiu7
hist_0_131	-->	raw2053	-->	dm-35		cust_0_561	-->	raw1331	-->	sdiu8
ordr_0_62	-->	raw1792	-->	dm-350		cust_0_562	-->	raw1332	-->	sdiu9
ordr_0_63	-->	raw1793	-->	dm-351		icust2_0_59	-->	raw2173	-->	sdiv1
ordr_0_64	-->	raw1794	-->	dm-352		iordr2_0_59	-->	raw2365	-->	sdiv2
ordr_0_65	-->	raw1795	-->	dm-353		temp_0_59	-->	raw2557	-->	sdiv3
ordr_0_66	-->	raw1796	-->	dm-354		stok_0_240	-->	raw241	-->	sdiy1
ordr_0_67	-->	raw1797	-->	dm-355		cust_0_566	-->	raw1336	-->	sdiy10
ordr_0_68	-->	raw1798	-->	dm-356		stok_0_241	-->	raw242	-->	sdiy2
ordr_0_69	-->	raw1799	-->	dm-357		stok_0_242	-->	raw243	-->	sdiy3
ordr_0_70	-->	raw1800	-->	dm-358		stok_0_243	-->	raw244	-->	sdiy5
ordr_0_71	-->	raw1801	-->	dm-359		cust_0_120	-->	raw890	-->	sdiy6
hist_0_132	-->	raw2054	-->	dm-36		cust_0_121	-->	raw891	-->	sdiy7
ordr_0_72	-->	raw1802	-->	dm-360		cust_0_564	-->	raw1334	-->	sdiy8
ordr_0_73	-->	raw1803	-->	dm-361		cust_0_565	-->	raw1335	-->	sdiy9
ordr_0_74	-->	raw1804	-->	dm-362		icust2_0_60	-->	raw2174	-->	sdiz1
ordr_0_75	-->	raw1805	-->	dm-363		iordr2_0_60	-->	raw2366	-->	sdiz2
ordr_0_76	-->	raw1806	-->	dm-364		temp_0_60	-->	raw2558	-->	sdiz3
ordr_0_77	-->	raw1807	-->	dm-365		stok_0_244	-->	raw245	-->	sdjc1
ordr_0_78	-->	raw1808	-->	dm-366		cust_0_569	-->	raw1339	-->	sdjc10
ordr_0_79	-->	raw1809	-->	dm-367		stok_0_245	-->	raw246	-->	sdjc2
ordr_0_80	-->	raw1810	-->	dm-368		stok_0_246	-->	raw247	-->	sdjc3
ordr_0_81	-->	raw1811	-->	dm-369		stok_0_247	-->	raw248	-->	sdjc5
hist_0_133	-->	raw2055	-->	dm-37		cust_0_122	-->	raw892	-->	sdjc6
ordr_0_82	-->	raw1812	-->	dm-370		cust_0_123	-->	raw893	-->	sdjc7
ordr_0_83	-->	raw1813	-->	dm-371		cust_0_567	-->	raw1337	-->	sdjc8
ordr_0_84	-->	raw1814	-->	dm-372		cust_0_568	-->	raw1338	-->	sdjc9
ordr_0_85	-->	raw1815	-->	dm-373		icust2_0_61	-->	raw2175	-->	sdjd1

ordr_0_86	-->	raw1816	-->	dm-374		iordr2_0_61	-->	raw2367	-->	sdjd2
ordr_0_87	-->	raw1817	-->	dm-375		temp_0_61	-->	raw2559	-->	sdjd3
ordr_0_88	-->	raw1818	-->	dm-376		stok_0_248	-->	raw249	-->	sdjg1
ordr_0_89	-->	raw1819	-->	dm-377		cust_0_572	-->	raw1342	-->	sdjg10
ordr_0_90	-->	raw1820	-->	dm-378		stok_0_249	-->	raw250	-->	sdjg2
ordr_0_91	-->	raw1821	-->	dm-379		stok_0_250	-->	raw251	-->	sdjg3
hist_0_134	-->	raw2056	-->	dm-38		stok_0_251	-->	raw252	-->	sdjg5
ordr_0_92	-->	raw1822	-->	dm-380		cust_0_124	-->	raw894	-->	sdjg6
ordr_0_93	-->	raw1823	-->	dm-381		cust_0_125	-->	raw895	-->	sdjg7
ordr_0_94	-->	raw1824	-->	dm-382		cust_0_570	-->	raw1340	-->	sdjg8
ordr_0_95	-->	raw1825	-->	dm-383		cust_0_571	-->	raw1341	-->	sdjg9
hist_0_135	-->	raw2057	-->	dm-39		icust2_0_62	-->	raw2176	-->	sdjh1
hist_0_100	-->	raw2022	-->	dm-4		iordr2_0_62	-->	raw2368	-->	sdjh2
hist_0_136	-->	raw2058	-->	dm-40		temp_0_62	-->	raw2560	-->	sdjh3
hist_0_137	-->	raw2059	-->	dm-41		stok_0_252	-->	raw253	-->	sdjk1
hist_0_138	-->	raw2060	-->	dm-42		cust_0_575	-->	raw1345	-->	sdjk10
hist_0_139	-->	raw2061	-->	dm-43		stok_0_253	-->	raw254	-->	sdjk2
hist_0_140	-->	raw2062	-->	dm-44		stok_0_254	-->	raw255	-->	sdjk3
hist_0_141	-->	raw2063	-->	dm-45		stok_0_255	-->	raw256	-->	sdjk5
hist_0_142	-->	raw2064	-->	dm-46		cust_0_126	-->	raw896	-->	sdjk6
hist_0_143	-->	raw2065	-->	dm-47		cust_0_127	-->	raw897	-->	sdjk7
hist_0_144	-->	raw2066	-->	dm-48		cust_0_573	-->	raw1343	-->	sdjk8
hist_0_145	-->	raw2067	-->	dm-49		cust_0_574	-->	raw1344	-->	sdjk9
hist_0_101	-->	raw2023	-->	dm-5		icust2_0_63	-->	raw2177	-->	sdjl1
hist_0_146	-->	raw2068	-->	dm-50		iordr2_0_63	-->	raw2369	-->	sdjl2
hist_0_147	-->	raw2069	-->	dm-51		temp_0_63	-->	raw2561	-->	sdjl3
hist_0_148	-->	raw2070	-->	dm-52		stok_0_256	-->	raw257	-->	sdjo1
hist_0_149	-->	raw2071	-->	dm-53		cust_0_578	-->	raw1348	-->	sdjo10
hist_0_150	-->	raw2072	-->	dm-54		stok_0_257	-->	raw258	-->	sdjo2
hist_0_151	-->	raw2073	-->	dm-55		stok_0_258	-->	raw259	-->	sdjo3
hist_0_152	-->	raw2074	-->	dm-56		stok_0_259	-->	raw260	-->	sdjo5
hist_0_153	-->	raw2075	-->	dm-57		cust_0_128	-->	raw898	-->	sdjo6
hist_0_154	-->	raw2076	-->	dm-58		cust_0_129	-->	raw899	-->	sdjo7
hist_0_155	-->	raw2077	-->	dm-59		cust_0_576	-->	raw1346	-->	sdjo8
hist_0_102	-->	raw2024	-->	dm-6		cust_0_577	-->	raw1347	-->	sdjo9
hist_0_156	-->	raw2078	-->	dm-60		icust2_0_64	-->	raw2178	-->	sdjp1
hist_0_157	-->	raw2079	-->	dm-61		iordr2_0_64	-->	raw2370	-->	sdjp2
hist_0_158	-->	raw2080	-->	dm-62		temp_0_64	-->	raw2562	-->	sdjp3
hist_0_159	-->	raw2081	-->	dm-63		stok_0_260	-->	raw261	-->	sdjs1
hist_0_160	-->	raw2082	-->	dm-64		cust_0_581	-->	raw1351	-->	sdjs10
hist_0_161	-->	raw2083	-->	dm-65		stok_0_261	-->	raw262	-->	sdjs2
hist_0_162	-->	raw2084	-->	dm-66		stok_0_262	-->	raw263	-->	sdjs3
hist_0_163	-->	raw2085	-->	dm-67		stok_0_263	-->	raw264	-->	sdjs5
hist_0_164	-->	raw2086	-->	dm-68		cust_0_130	-->	raw900	-->	sdjs6
hist_0_165	-->	raw2087	-->	dm-69		cust_0_131	-->	raw901	-->	sdjs7
hist_0_103	-->	raw2025	-->	dm-7		cust_0_579	-->	raw1349	-->	sdjs8
hist_0_166	-->	raw2088	-->	dm-70		cust_0_580	-->	raw1350	-->	sdjs9
hist_0_167	-->	raw2089	-->	dm-71		icust2_0_65	-->	raw2179	-->	sdjt1
hist_0_168	-->	raw2090	-->	dm-72		iordr2_0_65	-->	raw2371	-->	sdjt2
hist_0_169	-->	raw2091	-->	dm-73		temp_0_65	-->	raw2563	-->	sdjt3
hist_0_170	-->	raw2092	-->	dm-74		stok_0_264	-->	raw265	-->	sdjw1
hist_0_171	-->	raw2093	-->	dm-75		cust_0_584	-->	raw1354	-->	sdjw10
hist_0_172	-->	raw2094	-->	dm-76		stok_0_265	-->	raw266	-->	sdjw2
hist_0_173	-->	raw2095	-->	dm-77		stok_0_266	-->	raw267	-->	sdjw3
hist_0_174	-->	raw2096	-->	dm-78		stok_0_267	-->	raw268	-->	sdjw5
hist_0_175	-->	raw2097	-->	dm-79		cust_0_132	-->	raw902	-->	sdjw6
hist_0_104	-->	raw2026	-->	dm-8		cust_0_133	-->	raw903	-->	sdjw7
hist_0_176	-->	raw2098	-->	dm-80		cust_0_582	-->	raw1352	-->	sdjw8
hist_0_177	-->	raw2099	-->	dm-81		cust_0_583	-->	raw1353	-->	sdjw9
hist_0_178	-->	raw2100	-->	dm-82		icust2_0_66	-->	raw2180	-->	sdjx1
hist_0_179	-->	raw2101	-->	dm-83		iordr2_0_66	-->	raw2372	-->	sdjx2
hist_0_180	-->	raw2102	-->	dm-84		temp_0_66	-->	raw2564	-->	sdjx3
hist_0_181	-->	raw2103	-->	dm-85		stok_0_8	-->	raw9	-->	sdk1
hist_0_182	-->	raw2104	-->	dm-86		cust_0_392	-->	raw1162	-->	sdk10
hist_0_183	-->	raw2105	-->	dm-87		stok_0_9	-->	raw10	-->	sdk2

hist_0_184	-->	raw2106	-->	dm-88		stok_0_10	-->	raw11	-->	sdk3
hist_0_185	-->	raw2107	-->	dm-89		stok_0_11	-->	raw12	-->	sdk5
hist_0_105	-->	raw2027	-->	dm-9		cust_0_4	-->	raw774	-->	sdk6
hist_0_186	-->	raw2108	-->	dm-90		cust_0_5	-->	raw775	-->	sdk7
hist_0_187	-->	raw2109	-->	dm-91		cust_0_390	-->	raw1160	-->	sdk8
hist_0_188	-->	raw2110	-->	dm-92		cust_0_391	-->	raw1161	-->	sdk9
hist_0_189	-->	raw2111	-->	dm-93		stok_0_268	-->	raw269	-->	sdka1
hist_0_190	-->	raw2112	-->	dm-94		cust_0_587	-->	raw1357	-->	sdka10
hist_0_191	-->	raw2113	-->	dm-95		stok_0_269	-->	raw270	-->	sdka2
hist_0_0	-->	raw1922	-->	dm-96		stok_0_270	-->	raw271	-->	sdka3
hist_0_1	-->	raw1923	-->	dm-97		stok_0_271	-->	raw272	-->	sdka5
hist_0_2	-->	raw1924	-->	dm-98		cust_0_134	-->	raw904	-->	sdka6
hist_0_3	-->	raw1925	-->	dm-99		cust_0_135	-->	raw905	-->	sdka7
nord_0_0	-->	raw2882	-->	md0		cust_0_585	-->	raw1355	-->	sdka8
nord_0_1	-->	raw2883	-->	md1		cust_0_586	-->	raw1356	-->	sdka9
nord_0_9	-->	raw2912	-->	md18		icust2_0_67	-->	raw2181	-->	sdkb1
nord_0_2	-->	raw2884	-->	md2		iordr2_0_67	-->	raw2373	-->	sdkb2
nord_0_3	-->	raw2885	-->	md3		temp_0_67	-->	raw2565	-->	sdkb3
nord_0_4	-->	raw2886	-->	md4		stok_0_272	-->	raw273	-->	sdke1
nord_0_5	-->	raw2887	-->	md5		cust_0_590	-->	raw1360	-->	sdke10
nord_0_6	-->	raw2888	-->	md6		stok_0_273	-->	raw274	-->	sdke2
nord_0_7	-->	raw2889	-->	md7		stok_0_274	-->	raw275	-->	sdke3
nord_0_8	-->	raw2890	-->	md8		stok_0_275	-->	raw276	-->	sdke5
stok_0_24	-->	raw25	-->	sdaa1		cust_0_136	-->	raw906	-->	sdke6
cust_0_404	-->	raw1174	-->	sdaa10		cust_0_137	-->	raw907	-->	sdke7
stok_0_25	-->	raw26	-->	sdaa2		cust_0_588	-->	raw1358	-->	sdke8
stok_0_26	-->	raw27	-->	sdaa3		cust_0_589	-->	raw1359	-->	sdke9
stok_0_27	-->	raw28	-->	sdaa5		icust2_0_68	-->	raw2182	-->	sdkf1
cust_0_12	-->	raw782	-->	sdaa6		iordr2_0_68	-->	raw2374	-->	sdkf2
cust_0_13	-->	raw783	-->	sdaa7		temp_0_68	-->	raw2566	-->	sdkf3
cust_0_402	-->	raw1172	-->	sdaa8		stok_0_276	-->	raw277	-->	sdki1
cust_0_403	-->	raw1173	-->	sdaa9		cust_0_593	-->	raw1363	-->	sdki10
stok_0_576	-->	raw577	-->	sdaab1		stok_0_277	-->	raw278	-->	sdki2
cust_0_818	-->	raw1588	-->	sdaab10		stok_0_278	-->	raw279	-->	sdki3
stok_0_577	-->	raw578	-->	sdaab2		stok_0_279	-->	raw280	-->	sdki5
stok_0_578	-->	raw579	-->	sdaab3		cust_0_138	-->	raw908	-->	sdki6
stok_0_579	-->	raw580	-->	sdaab5		cust_0_139	-->	raw909	-->	sdki7
cust_0_288	-->	raw1058	-->	sdaab6		cust_0_591	-->	raw1361	-->	sdki8
cust_0_289	-->	raw1059	-->	sdaab7		cust_0_592	-->	raw1362	-->	sdki9
cust_0_816	-->	raw1586	-->	sdaab8		icust2_0_69	-->	raw2183	-->	sdkj1
cust_0_817	-->	raw1587	-->	sdaab9		iordr2_0_69	-->	raw2375	-->	sdkj2
icust2_0_144	-->	raw2258	-->	sdaac1		temp_0_69	-->	raw2567	-->	sdkj3
iordr2_0_144	-->	raw2450	-->	sdaac2		stok_0_280	-->	raw281	-->	sdkm1
temp_0_144	-->	raw2642	-->	sdaac3		cust_0_596	-->	raw1366	-->	sdkm10
stok_0_580	-->	raw581	-->	sdaaf1		stok_0_281	-->	raw282	-->	sdkm2
cust_0_821	-->	raw1591	-->	sdaaf10		stok_0_282	-->	raw283	-->	sdkm3
stok_0_581	-->	raw582	-->	sdaaf2		stok_0_283	-->	raw284	-->	sdkm5
stok_0_582	-->	raw583	-->	sdaaf3		cust_0_140	-->	raw910	-->	sdkm6
stok_0_583	-->	raw584	-->	sdaaf5		cust_0_141	-->	raw911	-->	sdkm7
cust_0_290	-->	raw1060	-->	sdaaf6		cust_0_594	-->	raw1364	-->	sdkm8
cust_0_291	-->	raw1061	-->	sdaaf7		cust_0_595	-->	raw1365	-->	sdkm9
cust_0_819	-->	raw1589	-->	sdaaf8		icust2_0_70	-->	raw2184	-->	sdkn1
cust_0_820	-->	raw1590	-->	sdaaf9		iordr2_0_70	-->	raw2376	-->	sdkn2
icust2_0_145	-->	raw2259	-->	sdaag1		temp_0_70	-->	raw2568	-->	sdkn3
iordr2_0_145	-->	raw2451	-->	sdaag2		stok_0_284	-->	raw285	-->	sdq1
temp_0_145	-->	raw2643	-->	sdaag3		cust_0_599	-->	raw1369	-->	sdq10
stok_0_584	-->	raw585	-->	sdaaj1		stok_0_285	-->	raw286	-->	sdq2
cust_0_824	-->	raw1594	-->	sdaaj10		stok_0_286	-->	raw287	-->	sdq3
stok_0_585	-->	raw586	-->	sdaaj2		stok_0_287	-->	raw288	-->	sdq5
stok_0_586	-->	raw587	-->	sdaaj3		cust_0_142	-->	raw912	-->	sdq6
stok_0_587	-->	raw588	-->	sdaaj5		cust_0_143	-->	raw913	-->	sdq7
cust_0_292	-->	raw1062	-->	sdaaj6		cust_0_597	-->	raw1367	-->	sdq8
cust_0_293	-->	raw1063	-->	sdaaj7		cust_0_598	-->	raw1368	-->	sdq9
cust_0_822	-->	raw1592	-->	sdaaj8		icust2_0_71	-->	raw2185	-->	sdkr1
cust_0_823	-->	raw1593	-->	sdaaj9		iordr2_0_71	-->	raw2377	-->	sdkr2

icust2_0_146 --> raw2260 --> sdaak1	temp_0_71 --> raw2569 --> sdkr3
iordr2_0_146 --> raw2452 --> sdaak2	stok_0_288 --> raw289 --> sdku1
temp_0_146 --> raw2644 --> sdaak3	cust_0_602 --> raw1372 --> sdku10
stok_0_588 --> raw589 --> sdaan1	stok_0_289 --> raw290 --> sdku2
cust_0_827 --> raw1597 --> sdaan10	stok_0_290 --> raw291 --> sdku3
stok_0_589 --> raw590 --> sdaan2	stok_0_291 --> raw292 --> sdku5
stok_0_590 --> raw591 --> sdaan3	cust_0_144 --> raw914 --> sdku6
stok_0_591 --> raw592 --> sdaan5	cust_0_145 --> raw915 --> sdku7
cust_0_294 --> raw1064 --> sdaan6	cust_0_600 --> raw1370 --> sdku8
cust_0_295 --> raw1065 --> sdaan7	cust_0_601 --> raw1371 --> sdku9
cust_0_825 --> raw1595 --> sdaan8	icust2_0_72 --> raw2186 --> sdkv1
cust_0_826 --> raw1596 --> sdaan9	iordr2_0_72 --> raw2378 --> sdkv2
icust2_0_147 --> raw2261 --> sdaao1	temp_0_72 --> raw2570 --> sdkv3
iordr2_0_147 --> raw2453 --> sdaao2	stok_0_292 --> raw293 --> sdky1
temp_0_147 --> raw2645 --> sdaao3	cust_0_605 --> raw1375 --> sdky10
stok_0_592 --> raw593 --> sdaar1	stok_0_293 --> raw294 --> sdky2
cust_0_830 --> raw1600 --> sdaar10	stok_0_294 --> raw295 --> sdky3
stok_0_593 --> raw594 --> sdaar2	stok_0_295 --> raw296 --> sdky5
stok_0_594 --> raw595 --> sdaar3	cust_0_146 --> raw916 --> sdky6
stok_0_595 --> raw596 --> sdaar5	cust_0_147 --> raw917 --> sdky7
cust_0_296 --> raw1066 --> sdaar6	cust_0_603 --> raw1373 --> sdky8
cust_0_297 --> raw1067 --> sdaar7	cust_0_604 --> raw1374 --> sdky9
cust_0_828 --> raw1598 --> sdaar8	icust2_0_73 --> raw2187 --> sdkz1
cust_0_829 --> raw1599 --> sdaar9	iordr2_0_73 --> raw2379 --> sdkz2
icust2_0_148 --> raw2262 --> sdaas1	temp_0_73 --> raw2571 --> sdkz3
iordr2_0_148 --> raw2454 --> sdaas2	icust2_0_2 --> raw2116 --> sdl1
temp_0_148 --> raw2646 --> sdaas3	iordr2_0_2 --> raw2308 --> sdl2
stok_0_596 --> raw597 --> sdaav1	temp_0_2 --> raw2500 --> sdl3
cust_0_833 --> raw1603 --> sdaav10	icust1_0_2 --> raw2692 --> sdl5
stok_0_597 --> raw598 --> sdaav2	stok_0_296 --> raw297 --> sdlc1
stok_0_598 --> raw599 --> sdaav3	cust_0_608 --> raw1378 --> sdlc10
stok_0_599 --> raw600 --> sdaav5	stok_0_297 --> raw298 --> sdlc2
cust_0_298 --> raw1068 --> sdaav6	stok_0_298 --> raw299 --> sdlc3
cust_0_299 --> raw1069 --> sdaav7	stok_0_299 --> raw300 --> sdlc5
cust_0_831 --> raw1601 --> sdaav8	cust_0_148 --> raw918 --> sdlc6
cust_0_832 --> raw1602 --> sdaav9	cust_0_149 --> raw919 --> sdlc7
icust2_0_149 --> raw2263 --> sdaaw1	cust_0_606 --> raw1376 --> sdlc8
iordr2_0_149 --> raw2455 --> sdaaw2	cust_0_607 --> raw1377 --> sdlc9
temp_0_149 --> raw2647 --> sdaaw3	icust2_0_74 --> raw2188 --> sddl1
stok_0_600 --> raw601 --> sdaaz1	iordr2_0_74 --> raw2380 --> sddl2
cust_0_836 --> raw1606 --> sdaaz10	temp_0_74 --> raw2572 --> sddl3
stok_0_601 --> raw602 --> sdaaz2	stok_0_300 --> raw301 --> sdlg1
stok_0_602 --> raw603 --> sdaaz3	cust_0_611 --> raw1381 --> sdlg10
stok_0_603 --> raw604 --> sdaaz5	stok_0_301 --> raw302 --> sdlg2
cust_0_300 --> raw1070 --> sdaaz6	stok_0_302 --> raw303 --> sdlg3
cust_0_301 --> raw1071 --> sdaaz7	stok_0_303 --> raw304 --> sdlg5
cust_0_834 --> raw1604 --> sdaaz8	cust_0_150 --> raw920 --> sdlg6
cust_0_835 --> raw1605 --> sdaaz9	cust_0_151 --> raw921 --> sdlg7
icust2_0_6 --> raw2120 --> sdab1	cust_0_609 --> raw1379 --> sdlg8
iordr2_0_6 --> raw2312 --> sdab2	cust_0_610 --> raw1380 --> sdlg9
temp_0_6 --> raw2504 --> sdab3	icust2_0_75 --> raw2189 --> sdlh1
istok_0_3 --> raw2696 --> sdab5	iordr2_0_75 --> raw2381 --> sdlh2
icust2_0_150 --> raw2264 --> sdaba1	temp_0_75 --> raw2573 --> sdlh3
iordr2_0_150 --> raw2456 --> sdaba2	stok_0_304 --> raw305 --> sdlk1
temp_0_150 --> raw2648 --> sdaba3	cust_0_614 --> raw1384 --> sdlk10
stok_0_604 --> raw605 --> sdabd1	stok_0_305 --> raw306 --> sdlk2
cust_0_839 --> raw1609 --> sdabd10	stok_0_306 --> raw307 --> sdlk3
stok_0_605 --> raw606 --> sdabd2	stok_0_307 --> raw308 --> sdlk5
stok_0_606 --> raw607 --> sdabd3	cust_0_152 --> raw922 --> sdlk6
stok_0_607 --> raw608 --> sdabd5	cust_0_153 --> raw923 --> sdlk7
cust_0_302 --> raw1072 --> sdabd6	cust_0_612 --> raw1382 --> sdlk8
cust_0_303 --> raw1073 --> sdabd7	cust_0_613 --> raw1383 --> sdlk9
cust_0_837 --> raw1607 --> sdabd8	icust2_0_76 --> raw2190 --> sdll1
cust_0_838 --> raw1608 --> sdabd9	iordr2_0_76 --> raw2382 --> sdll2
icust2_0_151 --> raw2265 --> sdabe1	temp_0_76 --> raw2574 --> sdll3

iordr2_0_151	-->	raw2457	-->	sdabe2		stok_0_308	-->	raw309	-->	sdlo1
temp_0_151	-->	raw2649	-->	sdabe3		cust_0_617	-->	raw1387	-->	sdlo10
stok_0_608	-->	raw609	-->	sdabh1		stok_0_309	-->	raw310	-->	sdlo2
cust_0_842	-->	raw1612	-->	sdabh10		stok_0_310	-->	raw311	-->	sdlo3
stok_0_609	-->	raw610	-->	sdabh2		stok_0_311	-->	raw312	-->	sdlo5
stok_0_610	-->	raw611	-->	sdabh3		cust_0_154	-->	raw924	-->	sdlo6
stok_0_611	-->	raw612	-->	sdabh5		cust_0_155	-->	raw925	-->	sdlo7
cust_0_304	-->	raw1074	-->	sdabh6		cust_0_615	-->	raw1385	-->	sdlo8
cust_0_305	-->	raw1075	-->	sdabh7		cust_0_616	-->	raw1386	-->	sdlo9
cust_0_840	-->	raw1610	-->	sdabh8		icust2_0_77	-->	raw2191	-->	sdlp1
cust_0_841	-->	raw1611	-->	sdabh9		iordr2_0_77	-->	raw2383	-->	sdlp2
icust2_0_152	-->	raw2266	-->	sdabi1		temp_0_77	-->	raw2575	-->	sdlp3
iordr2_0_152	-->	raw2458	-->	sdabi2		stok_0_312	-->	raw313	-->	sdls1
temp_0_152	-->	raw2650	-->	sdabi3		cust_0_620	-->	raw1390	-->	sdls10
stok_0_612	-->	raw613	-->	sdabl1		stok_0_313	-->	raw314	-->	sdls2
cust_0_845	-->	raw1615	-->	sdabl10		stok_0_314	-->	raw315	-->	sdls3
stok_0_613	-->	raw614	-->	sdabl2		stok_0_315	-->	raw316	-->	sdls5
stok_0_614	-->	raw615	-->	sdabl3		cust_0_156	-->	raw926	-->	sdls6
stok_0_615	-->	raw616	-->	sdabl5		cust_0_157	-->	raw927	-->	sdls7
cust_0_306	-->	raw1076	-->	sdabl6		cust_0_618	-->	raw1388	-->	sdls8
cust_0_307	-->	raw1077	-->	sdabl7		cust_0_619	-->	raw1389	-->	sdls9
cust_0_843	-->	raw1613	-->	sdabl8		icust2_0_78	-->	raw2192	-->	sdlt1
cust_0_844	-->	raw1614	-->	sdabl9		iordr2_0_78	-->	raw2384	-->	sdlt2
icust2_0_153	-->	raw2267	-->	sdabm1		temp_0_78	-->	raw2576	-->	sdlt3
iordr2_0_153	-->	raw2459	-->	sdabm2		stok_0_316	-->	raw317	-->	sdlw1
temp_0_153	-->	raw2651	-->	sdabm3		cust_0_623	-->	raw1393	-->	sdlw10
stok_0_616	-->	raw617	-->	sdabp1		stok_0_317	-->	raw318	-->	sdlw2
cust_0_848	-->	raw1618	-->	sdabp10		stok_0_318	-->	raw319	-->	sdlw3
stok_0_617	-->	raw618	-->	sdabp2		stok_0_319	-->	raw320	-->	sdlw5
stok_0_618	-->	raw619	-->	sdabp3		cust_0_158	-->	raw928	-->	sdlw6
stok_0_619	-->	raw620	-->	sdabp5		cust_0_159	-->	raw929	-->	sdlw7
cust_0_308	-->	raw1078	-->	sdabp6		cust_0_621	-->	raw1391	-->	sdlw8
cust_0_309	-->	raw1079	-->	sdabp7		cust_0_622	-->	raw1392	-->	sdlw9
cust_0_846	-->	raw1616	-->	sdabp8		icust2_0_79	-->	raw2193	-->	sdlx1
cust_0_847	-->	raw1617	-->	sdabp9		iordr2_0_79	-->	raw2385	-->	sdlx2
icust2_0_154	-->	raw2268	-->	sdabq1		temp_0_79	-->	raw2577	-->	sdlx3
iordr2_0_154	-->	raw2460	-->	sdabq2		stok_0_320	-->	raw321	-->	sdma1
temp_0_154	-->	raw2652	-->	sdabq3		cust_0_626	-->	raw1396	-->	sdma10
stok_0_620	-->	raw621	-->	sdabt1		stok_0_321	-->	raw322	-->	sdma2
cust_0_851	-->	raw1621	-->	sdabt10		stok_0_322	-->	raw323	-->	sdma3
stok_0_621	-->	raw622	-->	sdabt2		stok_0_323	-->	raw324	-->	sdma5
stok_0_622	-->	raw623	-->	sdabt3		cust_0_160	-->	raw930	-->	sdma6
stok_0_623	-->	raw624	-->	sdabt5		cust_0_161	-->	raw931	-->	sdma7
cust_0_310	-->	raw1080	-->	sdabt6		cust_0_624	-->	raw1394	-->	sdma8
cust_0_311	-->	raw1081	-->	sdabt7		cust_0_625	-->	raw1395	-->	sdma9
cust_0_849	-->	raw1619	-->	sdabt8		icust2_0_80	-->	raw2194	-->	sdmb1
cust_0_850	-->	raw1620	-->	sdabt9		iordr2_0_80	-->	raw2386	-->	sdmb2
icust2_0_155	-->	raw2269	-->	sdabu1		temp_0_80	-->	raw2578	-->	sdmb3
iordr2_0_155	-->	raw2461	-->	sdabu2		ware_0_0	-->	raw2891	-->	sdmb6
temp_0_155	-->	raw2653	-->	sdabu3		stok_0_324	-->	raw325	-->	sdme1
stok_0_624	-->	raw625	-->	sdabx1		cust_0_629	-->	raw1399	-->	sdme10
cust_0_854	-->	raw1624	-->	sdabx10		stok_0_325	-->	raw326	-->	sdme2
stok_0_625	-->	raw626	-->	sdabx2		stok_0_326	-->	raw327	-->	sdme3
stok_0_626	-->	raw627	-->	sdabx3		stok_0_327	-->	raw328	-->	sdme5
stok_0_627	-->	raw628	-->	sdabx5		cust_0_162	-->	raw932	-->	sdme6
cust_0_312	-->	raw1082	-->	sdabx6		cust_0_163	-->	raw933	-->	sdme7
cust_0_313	-->	raw1083	-->	sdabx7		cust_0_627	-->	raw1397	-->	sdme8
cust_0_852	-->	raw1622	-->	sdabx8		cust_0_628	-->	raw1398	-->	sdme9
cust_0_853	-->	raw1623	-->	sdabx9		icust2_0_81	-->	raw2195	-->	sdmf1
icust2_0_156	-->	raw2270	-->	sdaby1		iordr2_0_81	-->	raw2387	-->	sdmf2
iordr2_0_156	-->	raw2462	-->	sdaby2		temp_0_81	-->	raw2579	-->	sdmf3
temp_0_156	-->	raw2654	-->	sdaby3		dist_0_0	-->	raw2892	-->	sdmf6
stok_0_628	-->	raw629	-->	sdacb1		stok_0_328	-->	raw329	-->	sdmi1
cust_0_857	-->	raw1627	-->	sdacb10		cust_0_632	-->	raw1402	-->	sdmi10
stok_0_629	-->	raw630	-->	sdacb2		stok_0_329	-->	raw330	-->	sdmi2

stok_0_630	-->	raw631	-->	sdacb3		stok_0_330	-->	raw331	-->	sdmi3
stok_0_631	-->	raw632	-->	sdacb5		stok_0_331	-->	raw332	-->	sdmi5
cust_0_314	-->	raw1084	-->	sdacb6		cust_0_164	-->	raw934	-->	sdmi6
cust_0_315	-->	raw1085	-->	sdacb7		cust_0_165	-->	raw935	-->	sdmi7
cust_0_855	-->	raw1625	-->	sdacb8		cust_0_630	-->	raw1400	-->	sdmi8
cust_0_856	-->	raw1626	-->	sdacb9		cust_0_631	-->	raw1401	-->	sdmi9
icust2_0_157	-->	raw2271	-->	sdacc1		icust2_0_82	-->	raw2196	-->	sdmj1
iordr2_0_157	-->	raw2463	-->	sdacc2		iordr2_0_82	-->	raw2388	-->	sdmj2
temp_0_157	-->	raw2655	-->	sdacc3		temp_0_82	-->	raw2580	-->	sdmj3
stok_0_632	-->	raw633	-->	sdacf1		item_0_0	-->	raw2893	-->	sdmj6
cust_0_860	-->	raw1630	-->	sdacf10		stok_0_332	-->	raw333	-->	sdmm1
stok_0_633	-->	raw634	-->	sdacf2		cust_0_635	-->	raw1405	-->	sdmm10
stok_0_634	-->	raw635	-->	sdacf3		stok_0_333	-->	raw334	-->	sdmm2
stok_0_635	-->	raw636	-->	sdacf5		stok_0_334	-->	raw335	-->	sdmm3
cust_0_316	-->	raw1086	-->	sdacf6		stok_0_335	-->	raw336	-->	sdmm5
cust_0_317	-->	raw1087	-->	sdacf7		cust_0_166	-->	raw936	-->	sdmm6
cust_0_858	-->	raw1628	-->	sdacf8		cust_0_167	-->	raw937	-->	sdmm7
cust_0_859	-->	raw1629	-->	sdacf9		cust_0_633	-->	raw1403	-->	sdmm8
icust2_0_158	-->	raw2272	-->	sdacg1		cust_0_634	-->	raw1404	-->	sdmm9
iordr2_0_158	-->	raw2464	-->	sdacg2		icust2_0_83	-->	raw2197	-->	sdmn1
temp_0_158	-->	raw2656	-->	sdacg3		iordr2_0_83	-->	raw2389	-->	sdmn2
stok_0_636	-->	raw637	-->	sdacj1		temp_0_83	-->	raw2581	-->	sdmn3
cust_0_863	-->	raw1633	-->	sdacj10		iware_0_0	-->	raw2894	-->	sdmm6
stok_0_637	-->	raw638	-->	sdacj2		stok_0_336	-->	raw337	-->	sdmq1
stok_0_638	-->	raw639	-->	sdacj3		cust_0_638	-->	raw1408	-->	sdmq10
stok_0_639	-->	raw640	-->	sdacj5		stok_0_337	-->	raw338	-->	sdmq2
cust_0_318	-->	raw1088	-->	sdacj6		stok_0_338	-->	raw339	-->	sdmq3
cust_0_319	-->	raw1089	-->	sdacj7		stok_0_339	-->	raw340	-->	sdmq5
cust_0_861	-->	raw1631	-->	sdacj8		cust_0_168	-->	raw938	-->	sdmq6
cust_0_862	-->	raw1632	-->	sdacj9		cust_0_169	-->	raw939	-->	sdmq7
icust2_0_159	-->	raw2273	-->	sdack1		cust_0_636	-->	raw1406	-->	sdmq8
iordr2_0_159	-->	raw2465	-->	sdack2		cust_0_637	-->	raw1407	-->	sdmq9
temp_0_159	-->	raw2657	-->	sdack3		icust2_0_84	-->	raw2198	-->	sdmr1
stok_0_640	-->	raw641	-->	sdacn1		iordr2_0_84	-->	raw2390	-->	sdmr2
cust_0_866	-->	raw1636	-->	sdacn10		temp_0_84	-->	raw2582	-->	sdmr3
stok_0_641	-->	raw642	-->	sdacn2		idist_0_0	-->	raw2895	-->	sdmr6
stok_0_642	-->	raw643	-->	sdacn3		stok_0_340	-->	raw341	-->	sdmu1
stok_0_643	-->	raw644	-->	sdacn5		cust_0_641	-->	raw1411	-->	sdmu10
cust_0_320	-->	raw1090	-->	sdacn6		stok_0_341	-->	raw342	-->	sdmu2
cust_0_321	-->	raw1091	-->	sdacn7		stok_0_342	-->	raw343	-->	sdmu3
cust_0_864	-->	raw1634	-->	sdacn8		stok_0_343	-->	raw344	-->	sdmu5
cust_0_865	-->	raw1635	-->	sdacn9		cust_0_170	-->	raw940	-->	sdmu6
icust2_0_160	-->	raw2274	-->	sdaco1		cust_0_171	-->	raw941	-->	sdmu7
iordr2_0_160	-->	raw2466	-->	sdaco2		cust_0_639	-->	raw1409	-->	sdmu8
temp_0_160	-->	raw2658	-->	sdaco3		cust_0_640	-->	raw1410	-->	sdmu9
stok_0_644	-->	raw645	-->	sdacr1		icust2_0_85	-->	raw2199	-->	sdmv1
cust_0_869	-->	raw1639	-->	sdacr10		iordr2_0_85	-->	raw2391	-->	sdmv2
stok_0_645	-->	raw646	-->	sdacr2		temp_0_85	-->	raw2583	-->	sdmv3
stok_0_646	-->	raw647	-->	sdacr3		iitem_0_0	-->	raw2896	-->	sdmv6
stok_0_647	-->	raw648	-->	sdacr5		stok_0_344	-->	raw345	-->	sdmy1
cust_0_322	-->	raw1092	-->	sdacr6		cust_0_644	-->	raw1414	-->	sdmy10
cust_0_323	-->	raw1093	-->	sdacr7		stok_0_345	-->	raw346	-->	sdmy2
cust_0_867	-->	raw1637	-->	sdacr8		stok_0_346	-->	raw347	-->	sdmy3
cust_0_868	-->	raw1638	-->	sdacr9		stok_0_347	-->	raw348	-->	sdmy5
icust2_0_161	-->	raw2275	-->	sdacs1		cust_0_172	-->	raw942	-->	sdmy6
iordr2_0_161	-->	raw2467	-->	sdacs2		cust_0_173	-->	raw943	-->	sdmy7
temp_0_161	-->	raw2659	-->	sdacs3		cust_0_642	-->	raw1412	-->	sdmy8
stok_0_648	-->	raw649	-->	sdacv1		cust_0_643	-->	raw1413	-->	sdmy9
cust_0_872	-->	raw1642	-->	sdacv10		icust2_0_86	-->	raw2200	-->	sdmz1
stok_0_649	-->	raw650	-->	sdacv2		iordr2_0_86	-->	raw2392	-->	sdmz2
stok_0_650	-->	raw651	-->	sdacv3		temp_0_86	-->	raw2584	-->	sdmz3
stok_0_651	-->	raw652	-->	sdacv5		sp_0	-->	raw2897	-->	sdmz6
cust_0_324	-->	raw1094	-->	sdacv6		stok_0_348	-->	raw349	-->	sdnc1
cust_0_325	-->	raw1095	-->	sdacv7		cust_0_647	-->	raw1417	-->	sdnc10
cust_0_870	-->	raw1640	-->	sdacv8		stok_0_349	-->	raw350	-->	sdnc2

cust_0_871	-->	raw1641	-->	sdacv9		stok_0_350	-->	raw351	-->	sdnc3
icust2_0_162	-->	raw2276	-->	sdacw1		stok_0_351	-->	raw352	-->	sdnc5
iordr2_0_162	-->	raw2468	-->	sdacw2		cust_0_174	-->	raw944	-->	sdnc6
temp_0_162	-->	raw2660	-->	sdacw3		cust_0_175	-->	raw945	-->	sdnc7
stok_0_652	-->	raw653	-->	sdacz1		cust_0_645	-->	raw1415	-->	sdnc8
cust_0_875	-->	raw1645	-->	sdacz10		cust_0_646	-->	raw1416	-->	sdnc9
stok_0_653	-->	raw654	-->	sdacz2		icust2_0_87	-->	raw2201	-->	sdnd1
stok_0_654	-->	raw655	-->	sdacz3		iordr2_0_87	-->	raw2393	-->	sdnd2
stok_0_655	-->	raw656	-->	sdacz5		temp_0_87	-->	raw2585	-->	sdnd3
cust_0_326	-->	raw1096	-->	sdacz6		system_1	-->	raw2898	-->	sdnd6
cust_0_327	-->	raw1097	-->	sdacz7		stok_0_352	-->	raw353	-->	sdng1
cust_0_873	-->	raw1643	-->	sdacz8		cust_0_650	-->	raw1420	-->	sdng10
cust_0_874	-->	raw1644	-->	sdacz9		stok_0_353	-->	raw354	-->	sdng2
icust2_0_163	-->	raw2277	-->	sdada1		stok_0_354	-->	raw355	-->	sdng3
iordr2_0_163	-->	raw2469	-->	sdada2		stok_0_355	-->	raw356	-->	sdng5
temp_0_163	-->	raw2661	-->	sdada3		cust_0_176	-->	raw946	-->	sdng6
stok_0_656	-->	raw657	-->	sdadd1		cust_0_177	-->	raw947	-->	sdng7
cust_0_878	-->	raw1648	-->	sdadd10		cust_0_648	-->	raw1418	-->	sdng8
stok_0_657	-->	raw658	-->	sdadd2		cust_0_649	-->	raw1419	-->	sdng9
stok_0_658	-->	raw659	-->	sdadd3		icust2_0_88	-->	raw2202	-->	sdnh1
stok_0_659	-->	raw660	-->	sdadd5		iordr2_0_88	-->	raw2394	-->	sdnh2
cust_0_328	-->	raw1098	-->	sdadd6		temp_0_88	-->	raw2586	-->	sdnh3
cust_0_329	-->	raw1099	-->	sdadd7		tpccaux	-->	raw2899	-->	sdnh6
cust_0_876	-->	raw1646	-->	sdadd8		stok_0_356	-->	raw357	-->	sdnk1
cust_0_877	-->	raw1647	-->	sdadd9		cust_0_653	-->	raw1423	-->	sdnk10
icust2_0_164	-->	raw2278	-->	sdade1		stok_0_357	-->	raw358	-->	sdnk2
iordr2_0_164	-->	raw2470	-->	sdade2		stok_0_358	-->	raw359	-->	sdnk3
temp_0_164	-->	raw2662	-->	sdade3		stok_0_359	-->	raw360	-->	sdnk5
stok_0_660	-->	raw661	-->	sdadh1		cust_0_178	-->	raw948	-->	sdnk6
cust_0_881	-->	raw1651	-->	sdadh10		cust_0_179	-->	raw949	-->	sdnk7
stok_0_661	-->	raw662	-->	sdadh2		cust_0_651	-->	raw1421	-->	sdnk8
stok_0_662	-->	raw663	-->	sdadh3		cust_0_652	-->	raw1422	-->	sdnk9
stok_0_663	-->	raw664	-->	sdadh5		icust2_0_89	-->	raw2203	-->	sdnl1
cust_0_330	-->	raw1100	-->	sdadh6		iordr2_0_89	-->	raw2395	-->	sdnl2
cust_0_331	-->	raw1101	-->	sdadh7		temp_0_89	-->	raw2587	-->	sdnl3
cust_0_879	-->	raw1649	-->	sdadh8		roll1	-->	raw2900	-->	sdnl6
cust_0_880	-->	raw1650	-->	sdadh9		stok_0_360	-->	raw361	-->	sdno1
icust2_0_165	-->	raw2279	-->	sdadi1		cust_0_656	-->	raw1426	-->	sdno10
iordr2_0_165	-->	raw2471	-->	sdadi2		stok_0_361	-->	raw362	-->	sdno2
temp_0_165	-->	raw2663	-->	sdadi3		stok_0_362	-->	raw363	-->	sdno3
stok_0_664	-->	raw665	-->	sdadi11		stok_0_363	-->	raw364	-->	sdno5
cust_0_884	-->	raw1654	-->	sdadi10		cust_0_180	-->	raw950	-->	sdno6
stok_0_665	-->	raw666	-->	sdadi2		cust_0_181	-->	raw951	-->	sdno7
stok_0_666	-->	raw667	-->	sdadi3		cust_0_654	-->	raw1424	-->	sdno8
stok_0_667	-->	raw668	-->	sdadi5		cust_0_655	-->	raw1425	-->	sdno9
cust_0_332	-->	raw1102	-->	sdadi6		icust2_0_90	-->	raw2204	-->	sdnp1
cust_0_333	-->	raw1103	-->	sdadi7		iordr2_0_90	-->	raw2396	-->	sdnp2
cust_0_882	-->	raw1652	-->	sdadi8		temp_0_90	-->	raw2588	-->	sdnp3
cust_0_883	-->	raw1653	-->	sdadi9		log_1_2	-->	raw2902	-->	sdnp6
icust2_0_166	-->	raw2280	-->	sdadm1		stok_0_364	-->	raw365	-->	sdns1
iordr2_0_166	-->	raw2472	-->	sdadm2		cust_0_659	-->	raw1429	-->	sdns10
temp_0_166	-->	raw2664	-->	sdadm3		stok_0_365	-->	raw366	-->	sdns2
stok_0_668	-->	raw669	-->	sdadp1		stok_0_366	-->	raw367	-->	sdns3
cust_0_887	-->	raw1657	-->	sdadp10		stok_0_367	-->	raw368	-->	sdns5
stok_0_669	-->	raw670	-->	sdadp2		cust_0_182	-->	raw952	-->	sdns6
stok_0_670	-->	raw671	-->	sdadp3		cust_0_183	-->	raw953	-->	sdns7
stok_0_671	-->	raw672	-->	sdadp5		cust_0_657	-->	raw1427	-->	sdns8
cust_0_334	-->	raw1104	-->	sdadp6		cust_0_658	-->	raw1428	-->	sdns9
cust_0_335	-->	raw1105	-->	sdadp7		icust2_0_91	-->	raw2205	-->	sdnt1
cust_0_885	-->	raw1655	-->	sdadp8		iordr2_0_91	-->	raw2397	-->	sdnt2
cust_0_886	-->	raw1656	-->	sdadp9		temp_0_91	-->	raw2589	-->	sdnt3
icust2_0_167	-->	raw2281	-->	sdadq1		system_2	-->	raw2913	-->	sdnt6
iordr2_0_167	-->	raw2473	-->	sdadq2		stok_0_368	-->	raw369	-->	sdnw1
temp_0_167	-->	raw2665	-->	sdadq3		cust_0_662	-->	raw1432	-->	sdnw10
stok_0_672	-->	raw673	-->	sdadt1		stok_0_369	-->	raw370	-->	sdnw2

cust_0_890	-->	raw1660	-->	sdadt10		stok_0_370	-->	raw371	-->	sdnw3
stok_0_673	-->	raw674	-->	sdadt2		stok_0_371	-->	raw372	-->	sdnw5
stok_0_674	-->	raw675	-->	sdadt3		cust_0_184	-->	raw954	-->	sdnw6
stok_0_675	-->	raw676	-->	sdadt5		cust_0_185	-->	raw955	-->	sdnw7
cust_0_336	-->	raw1106	-->	sdadt6		cust_0_660	-->	raw1430	-->	sdnw8
cust_0_337	-->	raw1107	-->	sdadt7		cust_0_661	-->	raw1431	-->	sdnw9
cust_0_888	-->	raw1658	-->	sdadt8		icust2_0_92	-->	raw2206	-->	sdnx1
cust_0_889	-->	raw1659	-->	sdadt9		iordr2_0_92	-->	raw2398	-->	sdnx2
icust2_0_168	-->	raw2282	-->	sdadu1		temp_0_92	-->	raw2590	-->	sdnx3
iordr2_0_168	-->	raw2474	-->	sdadu2		system_3	-->	raw2914	-->	sdnx6
temp_0_168	-->	raw2666	-->	sdadu3		stok_0_12	-->	raw13	-->	sdo1
stok_0_676	-->	raw677	-->	sdadx1		cust_0_395	-->	raw1165	-->	sdo10
cust_0_893	-->	raw1663	-->	sdadx10		stok_0_13	-->	raw14	-->	sdo2
stok_0_677	-->	raw678	-->	sdadx2		stok_0_14	-->	raw15	-->	sdo3
stok_0_678	-->	raw679	-->	sdadx3		stok_0_15	-->	raw16	-->	sdo5
stok_0_679	-->	raw680	-->	sdadx5		cust_0_6	-->	raw776	-->	sdo6
cust_0_338	-->	raw1108	-->	sdadx6		cust_0_7	-->	raw777	-->	sdo7
cust_0_339	-->	raw1109	-->	sdadx7		cust_0_393	-->	raw1163	-->	sdo8
cust_0_891	-->	raw1661	-->	sdadx8		cust_0_394	-->	raw1164	-->	sdo9
cust_0_892	-->	raw1662	-->	sdadx9		stok_0_372	-->	raw373	-->	sdoa1
icust2_0_169	-->	raw2283	-->	sdady1		cust_0_665	-->	raw1435	-->	sdoa10
iordr2_0_169	-->	raw2475	-->	sdady2		stok_0_373	-->	raw374	-->	sdoa2
temp_0_169	-->	raw2667	-->	sdady3		stok_0_374	-->	raw375	-->	sdoa3
stok_0_28	-->	raw29	-->	sdae1		stok_0_375	-->	raw376	-->	sdoa5
cust_0_407	-->	raw1177	-->	sdae10		cust_0_186	-->	raw956	-->	sdoa6
stok_0_29	-->	raw30	-->	sdae2		cust_0_187	-->	raw957	-->	sdoa7
stok_0_30	-->	raw31	-->	sdae3		cust_0_663	-->	raw1433	-->	sdoa8
stok_0_31	-->	raw32	-->	sdae5		cust_0_664	-->	raw1434	-->	sdoa9
cust_0_14	-->	raw784	-->	sdae6		icust2_0_93	-->	raw2207	-->	sdob1
cust_0_15	-->	raw785	-->	sdae7		iordr2_0_93	-->	raw2399	-->	sdob2
cust_0_405	-->	raw1175	-->	sdae8		temp_0_93	-->	raw2591	-->	sdob3
cust_0_406	-->	raw1176	-->	sdae9		system_4	-->	raw2915	-->	sdob6
stok_0_680	-->	raw681	-->	sdaeb1		stok_0_376	-->	raw377	-->	sdoe1
cust_0_896	-->	raw1666	-->	sdaeb10		cust_0_668	-->	raw1438	-->	sdoe10
stok_0_681	-->	raw682	-->	sdaeb2		stok_0_377	-->	raw378	-->	sdoe2
stok_0_682	-->	raw683	-->	sdaeb3		stok_0_378	-->	raw379	-->	sdoe3
stok_0_683	-->	raw684	-->	sdaeb5		stok_0_379	-->	raw380	-->	sdoe5
cust_0_340	-->	raw1110	-->	sdaeb6		cust_0_188	-->	raw958	-->	sdoe6
cust_0_341	-->	raw1111	-->	sdaeb7		cust_0_189	-->	raw959	-->	sdoe7
cust_0_894	-->	raw1664	-->	sdaeb8		cust_0_666	-->	raw1436	-->	sdoe8
cust_0_895	-->	raw1665	-->	sdaeb9		cust_0_667	-->	raw1437	-->	sdoe9
icust2_0_170	-->	raw2284	-->	sdaec1		icust2_0_94	-->	raw2208	-->	sdoe1
iordr2_0_170	-->	raw2476	-->	sdaec2		iordr2_0_94	-->	raw2400	-->	sdoe2
temp_0_170	-->	raw2668	-->	sdaec3		temp_0_94	-->	raw2592	-->	sdoe3
stok_0_684	-->	raw685	-->	sdaef1		system_5	-->	raw2916	-->	sdoe6
cust_0_899	-->	raw1669	-->	sdaef10		stok_0_380	-->	raw381	-->	sdoi1
stok_0_685	-->	raw686	-->	sdaef2		cust_0_671	-->	raw1441	-->	sdoi10
stok_0_686	-->	raw687	-->	sdaef3		stok_0_381	-->	raw382	-->	sdoi2
stok_0_687	-->	raw688	-->	sdaef5		stok_0_382	-->	raw383	-->	sdoi3
cust_0_342	-->	raw1112	-->	sdaef6		stok_0_383	-->	raw384	-->	sdoi5
cust_0_343	-->	raw1113	-->	sdaef7		cust_0_190	-->	raw960	-->	sdoi6
cust_0_897	-->	raw1667	-->	sdaef8		cust_0_191	-->	raw961	-->	sdoi7
cust_0_898	-->	raw1668	-->	sdaef9		cust_0_669	-->	raw1439	-->	sdoi8
icust2_0_171	-->	raw2285	-->	sdaeg1		cust_0_670	-->	raw1440	-->	sdoi9
iordr2_0_171	-->	raw2477	-->	sdaeg2		icust2_0_95	-->	raw2209	-->	sdoj1
temp_0_171	-->	raw2669	-->	sdaeg3		iordr2_0_95	-->	raw2401	-->	sdoj2
stok_0_688	-->	raw689	-->	sdaej1		temp_0_95	-->	raw2593	-->	sdoj3
cust_0_902	-->	raw1672	-->	sdaej10		def_0	-->	raw2917	-->	sdoj6
stok_0_689	-->	raw690	-->	sdaej2		stok_0_384	-->	raw385	-->	sdom1
stok_0_690	-->	raw691	-->	sdaej3		cust_0_674	-->	raw1444	-->	sdom10
stok_0_691	-->	raw692	-->	sdaej5		stok_0_385	-->	raw386	-->	sdom2
cust_0_344	-->	raw1114	-->	sdaej6		stok_0_386	-->	raw387	-->	sdom3
cust_0_345	-->	raw1115	-->	sdaej7		stok_0_387	-->	raw388	-->	sdom5
cust_0_900	-->	raw1670	-->	sdaej8		cust_0_192	-->	raw962	-->	sdom6
cust_0_901	-->	raw1671	-->	sdaej9		cust_0_193	-->	raw963	-->	sdom7

icust2_0_172 --> raw2286 --> sdaek1	cust_0_672 --> raw1442 --> sdom8
iordr2_0_172 --> raw2478 --> sdaek2	cust_0_673 --> raw1443 --> sdom9
temp_0_172 --> raw2670 --> sdaek3	icust2_0_96 --> raw2210 --> sdon1
stok_0_692 --> raw693 --> sdaen1	iordr2_0_96 --> raw2402 --> sdon2
cust_0_905 --> raw1675 --> sdaen10	temp_0_96 --> raw2594 --> sdon3
stok_0_693 --> raw694 --> sdaen2	def_1 --> raw2918 --> sdon6
stok_0_694 --> raw695 --> sdaen3	stok_0_388 --> raw389 --> sdoq1
stok_0_695 --> raw696 --> sdaen5	cust_0_677 --> raw1447 --> sdoq10
cust_0_346 --> raw1116 --> sdaen6	stok_0_389 --> raw390 --> sdoq2
cust_0_347 --> raw1117 --> sdaen7	stok_0_390 --> raw391 --> sdoq3
cust_0_903 --> raw1673 --> sdaen8	stok_0_391 --> raw392 --> sdoq5
cust_0_904 --> raw1674 --> sdaen9	cust_0_194 --> raw964 --> sdoq6
icust2_0_173 --> raw2287 --> sdaeo1	cust_0_195 --> raw965 --> sdoq7
iordr2_0_173 --> raw2479 --> sdaeo2	cust_0_675 --> raw1445 --> sdoq8
temp_0_173 --> raw2671 --> sdaeo3	cust_0_676 --> raw1446 --> sdoq9
stok_0_696 --> raw697 --> sdaer1	icust2_0_97 --> raw2211 --> sdor1
cust_0_908 --> raw1678 --> sdaer10	iordr2_0_97 --> raw2403 --> sdor2
stok_0_697 --> raw698 --> sdaer2	temp_0_97 --> raw2595 --> sdor3
stok_0_698 --> raw699 --> sdaer3	def_2 --> raw2919 --> sdor6
stok_0_699 --> raw700 --> sdaer5	stok_0_392 --> raw393 --> sdou1
cust_0_348 --> raw1118 --> sdaer6	cust_0_680 --> raw1450 --> sdou10
cust_0_349 --> raw1119 --> sdaer7	stok_0_393 --> raw394 --> sdou2
cust_0_906 --> raw1676 --> sdaer8	stok_0_394 --> raw395 --> sdou3
cust_0_907 --> raw1677 --> sdaer9	stok_0_395 --> raw396 --> sdou5
icust2_0_174 --> raw2288 --> sdaes1	cust_0_196 --> raw966 --> sdou6
iordr2_0_174 --> raw2480 --> sdaes2	cust_0_197 --> raw967 --> sdou7
temp_0_174 --> raw2672 --> sdaes3	cust_0_678 --> raw1448 --> sdou8
stok_0_700 --> raw701 --> sdaev1	cust_0_679 --> raw1449 --> sdou9
cust_0_911 --> raw1681 --> sdaev10	icust2_0_98 --> raw2212 --> sdov1
stok_0_701 --> raw702 --> sdaev2	iordr2_0_98 --> raw2404 --> sdov2
stok_0_702 --> raw703 --> sdaev3	temp_0_98 --> raw2596 --> sdov3
stok_0_703 --> raw704 --> sdaev5	def_3 --> raw2920 --> sdov6
cust_0_350 --> raw1120 --> sdaev6	stok_0_396 --> raw397 --> sdoy1
cust_0_351 --> raw1121 --> sdaev7	cust_0_683 --> raw1453 --> sdoy10
cust_0_909 --> raw1679 --> sdaev8	stok_0_397 --> raw398 --> sdoy2
cust_0_910 --> raw1680 --> sdaev9	stok_0_398 --> raw399 --> sdoy3
icust2_0_175 --> raw2289 --> sdaew1	stok_0_399 --> raw400 --> sdoy5
iordr2_0_175 --> raw2481 --> sdaew2	cust_0_198 --> raw968 --> sdoy6
temp_0_175 --> raw2673 --> sdaew3	cust_0_199 --> raw969 --> sdoy7
stok_0_704 --> raw705 --> sdaez1	cust_0_681 --> raw1451 --> sdoy8
cust_0_914 --> raw1684 --> sdaez10	cust_0_682 --> raw1452 --> sdoy9
stok_0_705 --> raw706 --> sdaez2	icust2_0_99 --> raw2213 --> sdoz1
stok_0_706 --> raw707 --> sdaez3	iordr2_0_99 --> raw2405 --> sdoz2
stok_0_707 --> raw708 --> sdaez5	temp_0_99 --> raw2597 --> sdoz3
cust_0_352 --> raw1122 --> sdaez6	def_4 --> raw2921 --> sdoz6
cust_0_353 --> raw1123 --> sdaez7	icust2_0_3 --> raw2117 --> sdp1
cust_0_912 --> raw1682 --> sdaez8	iordr2_0_3 --> raw2309 --> sdp2
cust_0_913 --> raw1683 --> sdaez9	temp_0_3 --> raw2501 --> sdp3
icust2_0_7 --> raw2121 --> sdaf1	istok_0_0 --> raw2693 --> sdp5
iordr2_0_7 --> raw2313 --> sdaf2	stok_0_400 --> raw401 --> sdpc1
temp_0_7 --> raw2505 --> sdaf3	cust_0_686 --> raw1456 --> sdpc10
istok_0_4 --> raw2697 --> sdaf5	stok_0_401 --> raw402 --> sdpc2
icust2_0_176 --> raw2290 --> sdafa1	stok_0_402 --> raw403 --> sdpc3
iordr2_0_176 --> raw2482 --> sdafa2	stok_0_403 --> raw404 --> sdpc5
temp_0_176 --> raw2674 --> sdafa3	cust_0_200 --> raw970 --> sdpc6
stok_0_708 --> raw709 --> sdafd1	cust_0_201 --> raw971 --> sdpc7
cust_0_917 --> raw1687 --> sdafd10	cust_0_684 --> raw1454 --> sdpc8
stok_0_709 --> raw710 --> sdafd2	cust_0_685 --> raw1455 --> sdpc9
stok_0_710 --> raw711 --> sdafd3	icust2_0_100 --> raw2214 --> sdpd1
stok_0_711 --> raw712 --> sdafd5	iordr2_0_100 --> raw2406 --> sdpd2
cust_0_354 --> raw1124 --> sdafd6	temp_0_100 --> raw2598 --> sdpd3
cust_0_355 --> raw1125 --> sdafd7	log_1_1 --> raw2901 --> sdpd6
cust_0_915 --> raw1685 --> sdafd8	stok_0_404 --> raw405 --> sdpg1
cust_0_916 --> raw1686 --> sdafd9	cust_0_689 --> raw1459 --> sdpg10
icust2_0_177 --> raw2291 --> sdafe1	stok_0_405 --> raw406 --> sdpg2

iordr2_0_177	-->	raw2483	-->	sdafe2		stok_0_406	-->	raw407	-->	sdpg3
temp_0_177	-->	raw2675	-->	sdafe3		stok_0_407	-->	raw408	-->	sdpg5
stok_0_712	-->	raw713	-->	sdafh1		cust_0_202	-->	raw972	-->	sdpg6
cust_0_920	-->	raw1690	-->	sdafh10		cust_0_203	-->	raw973	-->	sdpg7
stok_0_713	-->	raw714	-->	sdafh2		cust_0_687	-->	raw1457	-->	sdpg8
stok_0_714	-->	raw715	-->	sdafh3		cust_0_688	-->	raw1458	-->	sdpg9
stok_0_715	-->	raw716	-->	sdafh5		icust2_0_101	-->	raw2215	-->	sdph1
cust_0_356	-->	raw1126	-->	sdafh6		iordr2_0_101	-->	raw2407	-->	sdph2
cust_0_357	-->	raw1127	-->	sdafh7		temp_0_101	-->	raw2599	-->	sdph3
cust_0_918	-->	raw1688	-->	sdafh8		stok_0_408	-->	raw409	-->	sdpk1
cust_0_919	-->	raw1689	-->	sdafh9		cust_0_692	-->	raw1462	-->	sdpk10
icust2_0_178	-->	raw2292	-->	sdafi1		stok_0_409	-->	raw410	-->	sdpk2
iordr2_0_178	-->	raw2484	-->	sdafi2		stok_0_410	-->	raw411	-->	sdpk3
temp_0_178	-->	raw2676	-->	sdafi3		stok_0_411	-->	raw412	-->	sdpk5
stok_0_716	-->	raw717	-->	sdafl1		cust_0_204	-->	raw974	-->	sdpk6
cust_0_923	-->	raw1693	-->	sdafl10		cust_0_205	-->	raw975	-->	sdpk7
stok_0_717	-->	raw718	-->	sdafl2		cust_0_690	-->	raw1460	-->	sdpk8
stok_0_718	-->	raw719	-->	sdafl3		cust_0_691	-->	raw1461	-->	sdpk9
stok_0_719	-->	raw720	-->	sdafl5		icust2_0_102	-->	raw2216	-->	sdpl1
cust_0_358	-->	raw1128	-->	sdafl6		iordr2_0_102	-->	raw2408	-->	sdpl2
cust_0_359	-->	raw1129	-->	sdafl7		temp_0_102	-->	raw2600	-->	sdpl3
cust_0_921	-->	raw1691	-->	sdafl8		stok_0_412	-->	raw413	-->	sdpo1
cust_0_922	-->	raw1692	-->	sdafl9		cust_0_695	-->	raw1465	-->	sdpo10
icust2_0_179	-->	raw2293	-->	sdafm1		stok_0_413	-->	raw414	-->	sdpo2
iordr2_0_179	-->	raw2485	-->	sdafm2		stok_0_414	-->	raw415	-->	sdpo3
temp_0_179	-->	raw2677	-->	sdafm3		stok_0_415	-->	raw416	-->	sdpo5
stok_0_720	-->	raw721	-->	sdafp1		cust_0_206	-->	raw976	-->	sdpo6
cust_0_926	-->	raw1696	-->	sdafp10		cust_0_207	-->	raw977	-->	sdpo7
stok_0_721	-->	raw722	-->	sdafp2		cust_0_693	-->	raw1463	-->	sdpo8
stok_0_722	-->	raw723	-->	sdafp3		cust_0_694	-->	raw1464	-->	sdpo9
stok_0_723	-->	raw724	-->	sdafp5		icust2_0_103	-->	raw2217	-->	sdpp1
cust_0_360	-->	raw1130	-->	sdafp6		iordr2_0_103	-->	raw2409	-->	sdpp2
cust_0_361	-->	raw1131	-->	sdafp7		temp_0_103	-->	raw2601	-->	sdpp3
cust_0_924	-->	raw1694	-->	sdafp8		stok_0_416	-->	raw417	-->	sdps1
cust_0_925	-->	raw1695	-->	sdafp9		cust_0_698	-->	raw1468	-->	sdps10
icust2_0_180	-->	raw2294	-->	sdafq1		stok_0_417	-->	raw418	-->	sdps2
iordr2_0_180	-->	raw2486	-->	sdafq2		stok_0_418	-->	raw419	-->	sdps3
temp_0_180	-->	raw2678	-->	sdafq3		stok_0_419	-->	raw420	-->	sdps5
stok_0_724	-->	raw725	-->	sdaft1		cust_0_208	-->	raw978	-->	sdps6
cust_0_929	-->	raw1699	-->	sdaft10		cust_0_209	-->	raw979	-->	sdps7
stok_0_725	-->	raw726	-->	sdaft2		cust_0_696	-->	raw1466	-->	sdps8
stok_0_726	-->	raw727	-->	sdaft3		cust_0_697	-->	raw1467	-->	sdps9
stok_0_727	-->	raw728	-->	sdaft5		icust2_0_104	-->	raw2218	-->	sdpt1
cust_0_362	-->	raw1132	-->	sdaft6		iordr2_0_104	-->	raw2410	-->	sdpt2
cust_0_363	-->	raw1133	-->	sdaft7		temp_0_104	-->	raw2602	-->	sdpt3
cust_0_927	-->	raw1697	-->	sdaft8		stok_0_420	-->	raw421	-->	sdpw1
cust_0_928	-->	raw1698	-->	sdaft9		cust_0_701	-->	raw1471	-->	sdpw10
icust2_0_181	-->	raw2295	-->	sdafu1		stok_0_421	-->	raw422	-->	sdpw2
iordr2_0_181	-->	raw2487	-->	sdafu2		stok_0_422	-->	raw423	-->	sdpw3
temp_0_181	-->	raw2679	-->	sdafu3		stok_0_423	-->	raw424	-->	sdpw5
stok_0_728	-->	raw729	-->	sdafx1		cust_0_210	-->	raw980	-->	sdpw6
cust_0_932	-->	raw1702	-->	sdafx10		cust_0_211	-->	raw981	-->	sdpw7
stok_0_729	-->	raw730	-->	sdafx2		cust_0_699	-->	raw1469	-->	sdpw8
stok_0_730	-->	raw731	-->	sdafx3		cust_0_700	-->	raw1470	-->	sdpw9
stok_0_731	-->	raw732	-->	sdafx5		icust2_0_105	-->	raw2219	-->	sdpx1
cust_0_364	-->	raw1134	-->	sdafx6		iordr2_0_105	-->	raw2411	-->	sdpx2
cust_0_365	-->	raw1135	-->	sdafx7		temp_0_105	-->	raw2603	-->	sdpx3
cust_0_930	-->	raw1700	-->	sdafx8		stok_0_424	-->	raw425	-->	sdqa1
cust_0_931	-->	raw1701	-->	sdafx9		cust_0_704	-->	raw1474	-->	sdqa10
icust2_0_182	-->	raw2296	-->	sdafy1		stok_0_425	-->	raw426	-->	sdqa2
iordr2_0_182	-->	raw2488	-->	sdafy2		stok_0_426	-->	raw427	-->	sdqa3
temp_0_182	-->	raw2680	-->	sdafy3		stok_0_427	-->	raw428	-->	sdqa5
stok_0_732	-->	raw733	-->	sdagb1		cust_0_212	-->	raw982	-->	sdqa6
cust_0_935	-->	raw1705	-->	sdagb10		cust_0_213	-->	raw983	-->	sdqa7
stok_0_733	-->	raw734	-->	sdagb2		cust_0_702	-->	raw1472	-->	sdqa8

stok_0_734	-->	raw735	-->	sdagb3		cust_0_703	-->	raw1473	-->	sdqa9
stok_0_735	-->	raw736	-->	sdagb5		icust2_0_106	-->	raw2220	-->	sdqb1
cust_0_366	-->	raw1136	-->	sdagb6		iordr2_0_106	-->	raw2412	-->	sdqb2
cust_0_367	-->	raw1137	-->	sdagb7		temp_0_106	-->	raw2604	-->	sdqb3
cust_0_933	-->	raw1703	-->	sdagb8		stok_0_428	-->	raw429	-->	sdqe1
cust_0_934	-->	raw1704	-->	sdagb9		cust_0_707	-->	raw1477	-->	sdqe10
icust2_0_183	-->	raw2297	-->	sdagc1		stok_0_429	-->	raw430	-->	sdqe2
iordr2_0_183	-->	raw2489	-->	sdagc2		stok_0_430	-->	raw431	-->	sdqe3
temp_0_183	-->	raw2681	-->	sdagc3		stok_0_431	-->	raw432	-->	sdqe5
stok_0_736	-->	raw737	-->	sdagf1		cust_0_214	-->	raw984	-->	sdqe6
cust_0_938	-->	raw1708	-->	sdagf10		cust_0_215	-->	raw985	-->	sdqe7
stok_0_737	-->	raw738	-->	sdagf2		cust_0_705	-->	raw1475	-->	sdqe8
stok_0_738	-->	raw739	-->	sdagf3		cust_0_706	-->	raw1476	-->	sdqe9
stok_0_739	-->	raw740	-->	sdagf5		icust2_0_107	-->	raw2221	-->	sdqf1
cust_0_368	-->	raw1138	-->	sdagf6		iordr2_0_107	-->	raw2413	-->	sdqf2
cust_0_369	-->	raw1139	-->	sdagf7		temp_0_107	-->	raw2605	-->	sdqf3
cust_0_936	-->	raw1706	-->	sdagf8		stok_0_432	-->	raw433	-->	sdqi1
cust_0_937	-->	raw1707	-->	sdagf9		cust_0_710	-->	raw1480	-->	sdqi10
icust2_0_184	-->	raw2298	-->	sdagg1		stok_0_433	-->	raw434	-->	sdqi2
iordr2_0_184	-->	raw2490	-->	sdagg2		stok_0_434	-->	raw435	-->	sdqi3
temp_0_184	-->	raw2682	-->	sdagg3		stok_0_435	-->	raw436	-->	sdqi5
stok_0_740	-->	raw741	-->	sdagj1		cust_0_216	-->	raw986	-->	sdqi6
cust_0_941	-->	raw1711	-->	sdagj10		cust_0_217	-->	raw987	-->	sdqi7
stok_0_741	-->	raw742	-->	sdagj2		cust_0_708	-->	raw1478	-->	sdqi8
stok_0_742	-->	raw743	-->	sdagj3		cust_0_709	-->	raw1479	-->	sdqi9
stok_0_743	-->	raw744	-->	sdagj5		icust2_0_108	-->	raw2222	-->	sdqj1
cust_0_370	-->	raw1140	-->	sdagj6		iordr2_0_108	-->	raw2414	-->	sdqj2
cust_0_371	-->	raw1141	-->	sdagj7		temp_0_108	-->	raw2606	-->	sdqj3
cust_0_939	-->	raw1709	-->	sdagj8		stok_0_436	-->	raw437	-->	sdqm1
cust_0_940	-->	raw1710	-->	sdagj9		cust_0_713	-->	raw1483	-->	sdqm10
icust2_0_185	-->	raw2299	-->	sdagk1		stok_0_437	-->	raw438	-->	sdqm2
iordr2_0_185	-->	raw2491	-->	sdagk2		stok_0_438	-->	raw439	-->	sdqm3
temp_0_185	-->	raw2683	-->	sdagk3		stok_0_439	-->	raw440	-->	sdqm5
stok_0_744	-->	raw745	-->	sdagn1		cust_0_218	-->	raw988	-->	sdqm6
cust_0_944	-->	raw1714	-->	sdagn10		cust_0_219	-->	raw989	-->	sdqm7
stok_0_745	-->	raw746	-->	sdagn2		cust_0_711	-->	raw1481	-->	sdqm8
stok_0_746	-->	raw747	-->	sdagn3		cust_0_712	-->	raw1482	-->	sdqm9
stok_0_747	-->	raw748	-->	sdagn5		icust2_0_109	-->	raw2223	-->	sdqn1
cust_0_372	-->	raw1142	-->	sdagn6		iordr2_0_109	-->	raw2415	-->	sdqn2
cust_0_373	-->	raw1143	-->	sdagn7		temp_0_109	-->	raw2607	-->	sdqn3
cust_0_942	-->	raw1712	-->	sdagn8		stok_0_440	-->	raw441	-->	sdqq1
cust_0_943	-->	raw1713	-->	sdagn9		cust_0_716	-->	raw1486	-->	sdqq10
icust2_0_186	-->	raw2300	-->	sdago1		stok_0_441	-->	raw442	-->	sdqq2
iordr2_0_186	-->	raw2492	-->	sdago2		stok_0_442	-->	raw443	-->	sdqq3
temp_0_186	-->	raw2684	-->	sdago3		stok_0_443	-->	raw444	-->	sdqq5
stok_0_748	-->	raw749	-->	sdagr1		cust_0_220	-->	raw990	-->	sdqq6
cust_0_947	-->	raw1717	-->	sdagr10		cust_0_221	-->	raw991	-->	sdqq7
stok_0_749	-->	raw750	-->	sdagr2		cust_0_714	-->	raw1484	-->	sdqq8
stok_0_750	-->	raw751	-->	sdagr3		cust_0_715	-->	raw1485	-->	sdqq9
stok_0_751	-->	raw752	-->	sdagr5		icust2_0_110	-->	raw2224	-->	sdqr1
cust_0_374	-->	raw1144	-->	sdagr6		iordr2_0_110	-->	raw2416	-->	sdqr2
cust_0_375	-->	raw1145	-->	sdagr7		temp_0_110	-->	raw2608	-->	sdqr3
cust_0_945	-->	raw1715	-->	sdagr8		stok_0_444	-->	raw445	-->	sdqu1
cust_0_946	-->	raw1716	-->	sdagr9		cust_0_719	-->	raw1489	-->	sdqu10
icust2_0_187	-->	raw2301	-->	sdags1		stok_0_445	-->	raw446	-->	sdqu2
iordr2_0_187	-->	raw2493	-->	sdags2		stok_0_446	-->	raw447	-->	sdqu3
temp_0_187	-->	raw2685	-->	sdags3		stok_0_447	-->	raw448	-->	sdqu5
stok_0_752	-->	raw753	-->	sdagv1		cust_0_222	-->	raw992	-->	sdqu6
cust_0_950	-->	raw1720	-->	sdagv10		cust_0_223	-->	raw993	-->	sdqu7
stok_0_753	-->	raw754	-->	sdagv2		cust_0_717	-->	raw1487	-->	sdqu8
stok_0_754	-->	raw755	-->	sdagv3		cust_0_718	-->	raw1488	-->	sdqu9
stok_0_755	-->	raw756	-->	sdagv5		icust2_0_111	-->	raw2225	-->	sdqv1
cust_0_376	-->	raw1146	-->	sdagv6		iordr2_0_111	-->	raw2417	-->	sdqv2
cust_0_377	-->	raw1147	-->	sdagv7		temp_0_111	-->	raw2609	-->	sdqv3
cust_0_948	-->	raw1718	-->	sdagv8		stok_0_448	-->	raw449	-->	sdqv1

cust_0_949	-->	raw1719	-->	sdagv9		cust_0_722	-->	raw1492	-->	sdqy10
icust2_0_188	-->	raw2302	-->	sdagw1		stok_0_449	-->	raw450	-->	sdqy2
iordr2_0_188	-->	raw2494	-->	sdagw2		stok_0_450	-->	raw451	-->	sdqy3
temp_0_188	-->	raw2686	-->	sdagw3		stok_0_451	-->	raw452	-->	sdqy5
stok_0_756	-->	raw757	-->	sdagz1		cust_0_224	-->	raw994	-->	sdqy6
cust_0_953	-->	raw1723	-->	sdagz10		cust_0_225	-->	raw995	-->	sdqy7
stok_0_757	-->	raw758	-->	sdagz2		cust_0_720	-->	raw1490	-->	sdqy8
stok_0_758	-->	raw759	-->	sdagz3		cust_0_721	-->	raw1491	-->	sdqy9
stok_0_759	-->	raw760	-->	sdagz5		icust2_0_112	-->	raw2226	-->	sdqz1
cust_0_378	-->	raw1148	-->	sdagz6		iordr2_0_112	-->	raw2418	-->	sdqz2
cust_0_379	-->	raw1149	-->	sdagz7		temp_0_112	-->	raw2610	-->	sdqz3
cust_0_951	-->	raw1721	-->	sdagz8		stok_0_452	-->	raw453	-->	sdrcl
cust_0_952	-->	raw1722	-->	sdagz9		cust_0_725	-->	raw1495	-->	sdrcl0
icust2_0_189	-->	raw2303	-->	sdaha1		stok_0_453	-->	raw454	-->	sdrcl2
iordr2_0_189	-->	raw2495	-->	sdaha2		stok_0_454	-->	raw455	-->	sdrcl3
temp_0_189	-->	raw2687	-->	sdaha3		stok_0_455	-->	raw456	-->	sdrcl5
stok_0_760	-->	raw761	-->	sdahd1		cust_0_226	-->	raw996	-->	sdrcl6
cust_0_956	-->	raw1726	-->	sdahd10		cust_0_227	-->	raw997	-->	sdrcl7
stok_0_761	-->	raw762	-->	sdahd2		cust_0_723	-->	raw1493	-->	sdrcl8
stok_0_762	-->	raw763	-->	sdahd3		cust_0_724	-->	raw1494	-->	sdrcl9
stok_0_763	-->	raw764	-->	sdahd5		icust2_0_113	-->	raw2227	-->	sdrdl
cust_0_380	-->	raw1150	-->	sdahd6		iordr2_0_113	-->	raw2419	-->	sdrdl2
cust_0_381	-->	raw1151	-->	sdahd7		temp_0_113	-->	raw2611	-->	sdrdl3
cust_0_954	-->	raw1724	-->	sdahd8		stok_0_456	-->	raw457	-->	sdrdl
cust_0_955	-->	raw1725	-->	sdahd9		cust_0_728	-->	raw1498	-->	sdrdl0
icust2_0_190	-->	raw2304	-->	sdahe1		stok_0_457	-->	raw458	-->	sdrdl2
iordr2_0_190	-->	raw2496	-->	sdahe2		stok_0_458	-->	raw459	-->	sdrdl3
temp_0_190	-->	raw2688	-->	sdahe3		stok_0_459	-->	raw460	-->	sdrdl5
stok_0_764	-->	raw765	-->	sdahh1		cust_0_228	-->	raw998	-->	sdrdl6
cust_0_959	-->	raw1729	-->	sdahh10		cust_0_229	-->	raw999	-->	sdrdl7
stok_0_765	-->	raw766	-->	sdahh2		cust_0_726	-->	raw1496	-->	sdrdl8
stok_0_766	-->	raw767	-->	sdahh3		cust_0_727	-->	raw1497	-->	sdrdl9
stok_0_767	-->	raw768	-->	sdahh5		icust2_0_114	-->	raw2228	-->	sdrhl
cust_0_382	-->	raw1152	-->	sdahh6		iordr2_0_114	-->	raw2420	-->	sdrhl2
cust_0_383	-->	raw1153	-->	sdahh7		temp_0_114	-->	raw2612	-->	sdrhl3
cust_0_957	-->	raw1727	-->	sdahh8		stok_0_460	-->	raw461	-->	sdrkl
cust_0_958	-->	raw1728	-->	sdahh9		cust_0_731	-->	raw1501	-->	sdrkl0
icust2_0_191	-->	raw2305	-->	sdahi1		stok_0_461	-->	raw462	-->	sdrkl2
iordr2_0_191	-->	raw2497	-->	sdahi2		stok_0_462	-->	raw463	-->	sdrkl3
temp_0_191	-->	raw2689	-->	sdahi3		stok_0_463	-->	raw464	-->	sdrkl5
stok_0_768	-->	raw769	-->	sdahi6		cust_0_230	-->	raw1000	-->	sdrkl6
stok_0_32	-->	raw33	-->	sdai1		cust_0_231	-->	raw1001	-->	sdrkl7
cust_0_410	-->	raw1180	-->	sdai10		cust_0_729	-->	raw1499	-->	sdrkl8
stok_0_33	-->	raw34	-->	sdai2		cust_0_730	-->	raw1500	-->	sdrkl9
stok_0_34	-->	raw35	-->	sdai3		icust2_0_115	-->	raw2229	-->	sdrll
stok_0_35	-->	raw36	-->	sdai5		iordr2_0_115	-->	raw2421	-->	sdrll2
cust_0_16	-->	raw786	-->	sdai6		temp_0_115	-->	raw2613	-->	sdrll3
cust_0_17	-->	raw787	-->	sdai7		stok_0_464	-->	raw465	-->	sdro1
cust_0_408	-->	raw1178	-->	sdai8		cust_0_734	-->	raw1504	-->	sdro10
cust_0_409	-->	raw1179	-->	sdai9		stok_0_465	-->	raw466	-->	sdro2
icust2_0_8	-->	raw2122	-->	sdaj1		stok_0_466	-->	raw467	-->	sdro3
iordr2_0_8	-->	raw2314	-->	sdaj2		stok_0_467	-->	raw468	-->	sdro5
temp_0_8	-->	raw2506	-->	sdaj3		cust_0_232	-->	raw1002	-->	sdro6
istok_0_5	-->	raw2698	-->	sdaj5		cust_0_233	-->	raw1003	-->	sdro7
stok_0_36	-->	raw37	-->	sdam1		cust_0_732	-->	raw1502	-->	sdro8
cust_0_413	-->	raw1183	-->	sdam10		cust_0_733	-->	raw1503	-->	sdro9
stok_0_37	-->	raw38	-->	sdam2		icust2_0_116	-->	raw2230	-->	sdrpl
stok_0_38	-->	raw39	-->	sdam3		iordr2_0_116	-->	raw2422	-->	sdrpl2
stok_0_39	-->	raw40	-->	sdam5		temp_0_116	-->	raw2614	-->	sdrpl3
cust_0_18	-->	raw788	-->	sdam6		stok_0_468	-->	raw469	-->	sdrs1
cust_0_19	-->	raw789	-->	sdam7		cust_0_737	-->	raw1507	-->	sdrs10
cust_0_411	-->	raw1181	-->	sdam8		stok_0_469	-->	raw470	-->	sdrs2
cust_0_412	-->	raw1182	-->	sdam9		stok_0_470	-->	raw471	-->	sdrs3
icust2_0_9	-->	raw2123	-->	sdan1		stok_0_471	-->	raw472	-->	sdrs5
iordr2_0_9	-->	raw2315	-->	sdan2		cust_0_234	-->	raw1004	-->	sdrs6

temp_0_9	-->	raw2507	-->	sdan3		cust_0_235	-->	raw1005	-->	sdrs7
istok_0_6	-->	raw2699	-->	sdan5		cust_0_735	-->	raw1505	-->	sdrs8
stok_0_40	-->	raw41	-->	sdaq1		cust_0_736	-->	raw1506	-->	sdrs9
cust_0_416	-->	raw1186	-->	sdaq10		icust2_0_117	-->	raw2231	-->	sdrt1
stok_0_41	-->	raw42	-->	sdaq2		iordr2_0_117	-->	raw2423	-->	sdrt2
stok_0_42	-->	raw43	-->	sdaq3		temp_0_117	-->	raw2615	-->	sdrt3
stok_0_43	-->	raw44	-->	sdaq5		stok_0_472	-->	raw473	-->	sdrw1
cust_0_20	-->	raw790	-->	sdaq6		cust_0_740	-->	raw1510	-->	sdrw10
cust_0_21	-->	raw791	-->	sdaq7		stok_0_473	-->	raw474	-->	sdrw2
cust_0_414	-->	raw1184	-->	sdaq8		stok_0_474	-->	raw475	-->	sdrw3
cust_0_415	-->	raw1185	-->	sdaq9		stok_0_475	-->	raw476	-->	sdrw5
icust2_0_10	-->	raw2124	-->	sdar1		cust_0_236	-->	raw1006	-->	sdrw6
iordr2_0_10	-->	raw2316	-->	sdar2		cust_0_237	-->	raw1007	-->	sdrw7
temp_0_10	-->	raw2508	-->	sdar3		cust_0_738	-->	raw1508	-->	sdrw8
stok_0_44	-->	raw45	-->	sdau1		cust_0_739	-->	raw1509	-->	sdrw9
cust_0_419	-->	raw1189	-->	sdau10		icust2_0_118	-->	raw2232	-->	sdrx1
stok_0_45	-->	raw46	-->	sdau2		iordr2_0_118	-->	raw2424	-->	sdrx2
stok_0_46	-->	raw47	-->	sdau3		temp_0_118	-->	raw2616	-->	sdrx3
stok_0_47	-->	raw48	-->	sdau5		cust_0_398	-->	raw1168	-->	sds10
cust_0_22	-->	raw792	-->	sdau6		stok_0_16	-->	raw17	-->	sds14
cust_0_23	-->	raw793	-->	sdau7		stok_0_17	-->	raw18	-->	sds2
cust_0_417	-->	raw1187	-->	sdau8		stok_0_18	-->	raw19	-->	sds3
cust_0_418	-->	raw1188	-->	sdau9		stok_0_19	-->	raw20	-->	sds5
icust2_0_11	-->	raw2125	-->	sdav1		cust_0_8	-->	raw778	-->	sds6
iordr2_0_11	-->	raw2317	-->	sdav2		cust_0_9	-->	raw779	-->	sds7
temp_0_11	-->	raw2509	-->	sdav3		cust_0_396	-->	raw1166	-->	sds8
stok_0_48	-->	raw49	-->	sday1		cust_0_397	-->	raw1167	-->	sds9
cust_0_422	-->	raw1192	-->	sday10		cust_0_743	-->	raw1513	-->	sdsa10
stok_0_49	-->	raw50	-->	sday2		stok_0_476	-->	raw477	-->	sdsa14
stok_0_50	-->	raw51	-->	sday3		stok_0_477	-->	raw478	-->	sdsa2
stok_0_51	-->	raw52	-->	sday5		stok_0_478	-->	raw479	-->	sdsa3
cust_0_24	-->	raw794	-->	sday6		stok_0_479	-->	raw480	-->	sdsa5
cust_0_25	-->	raw795	-->	sday7		cust_0_238	-->	raw1008	-->	sdsa6
cust_0_420	-->	raw1190	-->	sday8		cust_0_239	-->	raw1009	-->	sdsa7
cust_0_421	-->	raw1191	-->	sday9		cust_0_741	-->	raw1511	-->	sdsa8
icust2_0_12	-->	raw2126	-->	sdaz1		cust_0_742	-->	raw1512	-->	sdsa9
iordr2_0_12	-->	raw2318	-->	sdaz2		icust2_0_119	-->	raw2233	-->	sdsb1
temp_0_12	-->	raw2510	-->	sdaz3		iordr2_0_119	-->	raw2425	-->	sdsb2
stok_0_52	-->	raw53	-->	sdbc1		temp_0_119	-->	raw2617	-->	sdsb3
cust_0_425	-->	raw1195	-->	sdbc10		icust2_0_4	-->	raw2118	-->	sdt1
stok_0_53	-->	raw54	-->	sdbc2		iordr2_0_4	-->	raw2310	-->	sdt2
stok_0_54	-->	raw55	-->	sdbc3		temp_0_4	-->	raw2502	-->	sdt3
stok_0_55	-->	raw56	-->	sdbc5		istok_0_1	-->	raw2694	-->	sdt5
cust_0_26	-->	raw796	-->	sdbc6		stok_0_20	-->	raw21	-->	sdw1
cust_0_27	-->	raw797	-->	sdbc7		cust_0_401	-->	raw1171	-->	sdw10
cust_0_423	-->	raw1193	-->	sdbc8		stok_0_21	-->	raw22	-->	sdw2
cust_0_424	-->	raw1194	-->	sdbc9		stok_0_22	-->	raw23	-->	sdw3
icust2_0_13	-->	raw2127	-->	sdbd1		stok_0_23	-->	raw24	-->	sdw5
iordr2_0_13	-->	raw2319	-->	sdbd2		cust_0_10	-->	raw780	-->	sdw6
temp_0_13	-->	raw2511	-->	sdbd3		cust_0_11	-->	raw781	-->	sdw7
stok_0_56	-->	raw57	-->	sdbg1		cust_0_399	-->	raw1169	-->	sdw8
cust_0_428	-->	raw1198	-->	sdbg10		cust_0_400	-->	raw1170	-->	sdw9
stok_0_57	-->	raw58	-->	sdbg2		stok_0_480	-->	raw481	-->	sdwj1
stok_0_58	-->	raw59	-->	sdbg3		cust_0_746	-->	raw1516	-->	sdwj10
stok_0_59	-->	raw60	-->	sdbg5		stok_0_481	-->	raw482	-->	sdwj2
cust_0_28	-->	raw798	-->	sdbg6		stok_0_482	-->	raw483	-->	sdwj3
cust_0_29	-->	raw799	-->	sdbg7		stok_0_483	-->	raw484	-->	sdwj5
cust_0_426	-->	raw1196	-->	sdbg8		cust_0_240	-->	raw1010	-->	sdwj6
cust_0_427	-->	raw1197	-->	sdbg9		cust_0_241	-->	raw1011	-->	sdwj7
icust2_0_14	-->	raw2128	-->	sdbh1		cust_0_744	-->	raw1514	-->	sdwj8
iordr2_0_14	-->	raw2320	-->	sdbh2		cust_0_745	-->	raw1515	-->	sdwj9
temp_0_14	-->	raw2512	-->	sdbh3		icust2_0_120	-->	raw2234	-->	sdwk1
stok_0_60	-->	raw61	-->	sdbk1		iordr2_0_120	-->	raw2426	-->	sdwk2
cust_0_431	-->	raw1201	-->	sdbk10		temp_0_120	-->	raw2618	-->	sdwk3
stok_0_61	-->	raw62	-->	sdbk2		stok_0_484	-->	raw485	-->	sdwn1

stok_0_62	-->	raw63	-->	sdbk3		cust_0_749	-->	raw1519	-->	sdwn10
stok_0_63	-->	raw64	-->	sdbk5		stok_0_485	-->	raw486	-->	sdwn2
cust_0_30	-->	raw800	-->	sdbk6		stok_0_486	-->	raw487	-->	sdwn3
cust_0_31	-->	raw801	-->	sdbk7		stok_0_487	-->	raw488	-->	sdwn5
cust_0_429	-->	raw1199	-->	sdbk8		cust_0_242	-->	raw1012	-->	sdwn6
cust_0_430	-->	raw1200	-->	sdbk9		cust_0_243	-->	raw1013	-->	sdwn7
icust2_0_15	-->	raw2129	-->	sdbl1		cust_0_747	-->	raw1517	-->	sdwn8
iordr2_0_15	-->	raw2321	-->	sdbl2		cust_0_748	-->	raw1518	-->	sdwn9
temp_0_15	-->	raw2513	-->	sdbl3		icust2_0_121	-->	raw2235	-->	sdwo1
stok_0_64	-->	raw65	-->	sdbo1		iordr2_0_121	-->	raw2427	-->	sdwo2
cust_0_434	-->	raw1204	-->	sdbo10		temp_0_121	-->	raw2619	-->	sdwo3
stok_0_65	-->	raw66	-->	sdbo2		stok_0_488	-->	raw489	-->	sdwr1
stok_0_66	-->	raw67	-->	sdbo3		cust_0_752	-->	raw1522	-->	sdwr10
stok_0_67	-->	raw68	-->	sdbo5		stok_0_489	-->	raw490	-->	sdwr2
cust_0_32	-->	raw802	-->	sdbo6		stok_0_490	-->	raw491	-->	sdwr3
cust_0_33	-->	raw803	-->	sdbo7		stok_0_491	-->	raw492	-->	sdwr5
cust_0_432	-->	raw1202	-->	sdbo8		cust_0_244	-->	raw1014	-->	sdwr6
cust_0_433	-->	raw1203	-->	sdbo9		cust_0_245	-->	raw1015	-->	sdwr7
icust2_0_16	-->	raw2130	-->	sdbp1		cust_0_750	-->	raw1520	-->	sdwr8
iordr2_0_16	-->	raw2322	-->	sdbp2		cust_0_751	-->	raw1521	-->	sdwr9
temp_0_16	-->	raw2514	-->	sdbp3		icust2_0_122	-->	raw2236	-->	sdws1
stok_0_68	-->	raw69	-->	sdfs1		iordr2_0_122	-->	raw2428	-->	sdws2
cust_0_437	-->	raw1207	-->	sdfs10		temp_0_122	-->	raw2620	-->	sdws3
stok_0_69	-->	raw70	-->	sdfs2		stok_0_492	-->	raw493	-->	sdwv1
stok_0_70	-->	raw71	-->	sdfs3		cust_0_755	-->	raw1525	-->	sdwv10
stok_0_71	-->	raw72	-->	sdfs5		stok_0_493	-->	raw494	-->	sdwv2
cust_0_34	-->	raw804	-->	sdfs6		stok_0_494	-->	raw495	-->	sdwv3
cust_0_35	-->	raw805	-->	sdfs7		stok_0_495	-->	raw496	-->	sdwv5
cust_0_435	-->	raw1205	-->	sdfs8		cust_0_246	-->	raw1016	-->	sdwv6
cust_0_436	-->	raw1206	-->	sdfs9		cust_0_247	-->	raw1017	-->	sdwv7
icust2_0_17	-->	raw2131	-->	sdbt1		cust_0_753	-->	raw1523	-->	sdwv8
iordr2_0_17	-->	raw2323	-->	sdbt2		cust_0_754	-->	raw1524	-->	sdwv9
temp_0_17	-->	raw2515	-->	sdbt3		icust2_0_123	-->	raw2237	-->	sdww1
stok_0_72	-->	raw73	-->	sdbw1		iordr2_0_123	-->	raw2429	-->	sdww2
cust_0_440	-->	raw1210	-->	sdbw10		temp_0_123	-->	raw2621	-->	sdww3
stok_0_73	-->	raw74	-->	sdbw2		stok_0_496	-->	raw497	-->	sdwz1
stok_0_74	-->	raw75	-->	sdbw3		cust_0_758	-->	raw1528	-->	sdwz10
stok_0_75	-->	raw76	-->	sdbw5		stok_0_497	-->	raw498	-->	sdwz2
cust_0_36	-->	raw806	-->	sdbw6		stok_0_498	-->	raw499	-->	sdwz3
cust_0_37	-->	raw807	-->	sdbw7		stok_0_499	-->	raw500	-->	sdwz5
cust_0_438	-->	raw1208	-->	sdbw8		cust_0_248	-->	raw1018	-->	sdwz6
cust_0_439	-->	raw1209	-->	sdbw9		cust_0_249	-->	raw1019	-->	sdwz7
icust2_0_18	-->	raw2132	-->	sdbx1		cust_0_756	-->	raw1526	-->	sdwz8
iordr2_0_18	-->	raw2324	-->	sdbx2		cust_0_757	-->	raw1527	-->	sdwz9
temp_0_18	-->	raw2516	-->	sdbx3		icust2_0_5	-->	raw2119	-->	sdx1
stok_0_0	-->	raw1	-->	sdcl1		iordr2_0_5	-->	raw2311	-->	sdx2
cust_0_386	-->	raw1156	-->	sdcl10		temp_0_5	-->	raw2503	-->	sdx3
stok_0_1	-->	raw2	-->	sdcl2		istok_0_2	-->	raw2695	-->	sdx5
stok_0_2	-->	raw3	-->	sdcl3		icust2_0_124	-->	raw2238	-->	sdxa1
stok_0_3	-->	raw4	-->	sdcl5		iordr2_0_124	-->	raw2430	-->	sdxa2
cust_0_0	-->	raw770	-->	sdcl6		temp_0_124	-->	raw2622	-->	sdxa3
cust_0_1	-->	raw771	-->	sdcl7		stok_0_500	-->	raw501	-->	sdxd1
cust_0_384	-->	raw1154	-->	sdcl8		cust_0_761	-->	raw1531	-->	sdxd10
cust_0_385	-->	raw1155	-->	sdcl9		stok_0_501	-->	raw502	-->	sdxd2
stok_0_76	-->	raw77	-->	sdca1		stok_0_502	-->	raw503	-->	sdxd3
cust_0_443	-->	raw1213	-->	sdca10		stok_0_503	-->	raw504	-->	sdxd5
stok_0_77	-->	raw78	-->	sdca2		cust_0_250	-->	raw1020	-->	sdxd6
stok_0_78	-->	raw79	-->	sdca3		cust_0_251	-->	raw1021	-->	sdxd7
stok_0_79	-->	raw80	-->	sdca5		cust_0_759	-->	raw1529	-->	sdxd8
cust_0_38	-->	raw808	-->	sdca6		cust_0_760	-->	raw1530	-->	sdxd9
cust_0_39	-->	raw809	-->	sdca7		icust2_0_125	-->	raw2239	-->	sdxe1
cust_0_441	-->	raw1211	-->	sdca8		iordr2_0_125	-->	raw2431	-->	sdxe2
cust_0_442	-->	raw1212	-->	sdca9		temp_0_125	-->	raw2623	-->	sdxe3
icust2_0_19	-->	raw2133	-->	sdcb1		stok_0_504	-->	raw505	-->	sdxh1
iordr2_0_19	-->	raw2325	-->	sdcb2		cust_0_764	-->	raw1534	-->	sdxh10

temp_0_19	-->	raw2517	-->	sdcx3		stok_0_505	-->	raw506	-->	sdxc2
stok_0_80	-->	raw81	-->	sdce1		stok_0_506	-->	raw507	-->	sdxc3
cust_0_446	-->	raw1216	-->	sdce10		stok_0_507	-->	raw508	-->	sdxc5
stok_0_81	-->	raw82	-->	sdce2		cust_0_252	-->	raw1022	-->	sdxc6
stok_0_82	-->	raw83	-->	sdce3		cust_0_253	-->	raw1023	-->	sdxc7
stok_0_83	-->	raw84	-->	sdce5		cust_0_762	-->	raw1532	-->	sdxc8
cust_0_40	-->	raw810	-->	sdce6		cust_0_763	-->	raw1533	-->	sdxc9
cust_0_41	-->	raw811	-->	sdce7		icust2_0_126	-->	raw2240	-->	sdxi1
cust_0_444	-->	raw1214	-->	sdce8		iordr2_0_126	-->	raw2432	-->	sdxi2
cust_0_445	-->	raw1215	-->	sdce9		temp_0_126	-->	raw2624	-->	sdxi3
icust2_0_20	-->	raw2134	-->	sdcf1		stok_0_508	-->	raw509	-->	sdxi11
iordr2_0_20	-->	raw2326	-->	sdcf2		cust_0_767	-->	raw1537	-->	sdxi10
temp_0_20	-->	raw2518	-->	sdcf3		stok_0_509	-->	raw510	-->	sdxi12
stok_0_84	-->	raw85	-->	sdci1		stok_0_510	-->	raw511	-->	sdxi13
cust_0_449	-->	raw1219	-->	sdci10		stok_0_511	-->	raw512	-->	sdxi15
stok_0_85	-->	raw86	-->	sdci2		cust_0_254	-->	raw1024	-->	sdxi16
stok_0_86	-->	raw87	-->	sdci3		cust_0_255	-->	raw1025	-->	sdxi17
stok_0_87	-->	raw88	-->	sdci5		cust_0_765	-->	raw1535	-->	sdxi18
cust_0_42	-->	raw812	-->	sdci6		cust_0_766	-->	raw1536	-->	sdxi19
cust_0_43	-->	raw813	-->	sdci7		icust2_0_127	-->	raw2241	-->	sdxi1
cust_0_447	-->	raw1217	-->	sdci8		iordr2_0_127	-->	raw2433	-->	sdxi2
cust_0_448	-->	raw1218	-->	sdci9		temp_0_127	-->	raw2625	-->	sdxi3
icust2_0_21	-->	raw2135	-->	sdci1		stok_0_512	-->	raw513	-->	sdxi11
iordr2_0_21	-->	raw2327	-->	sdci2		cust_0_770	-->	raw1540	-->	sdxi10
temp_0_21	-->	raw2519	-->	sdci3		stok_0_513	-->	raw514	-->	sdxi12
stok_0_88	-->	raw89	-->	sdcm1		stok_0_514	-->	raw515	-->	sdxi13
cust_0_452	-->	raw1222	-->	sdcm10		stok_0_515	-->	raw516	-->	sdxi15
stok_0_89	-->	raw90	-->	sdcm2		cust_0_256	-->	raw1026	-->	sdxi16
stok_0_90	-->	raw91	-->	sdcm3		cust_0_257	-->	raw1027	-->	sdxi17
stok_0_91	-->	raw92	-->	sdcm5		cust_0_768	-->	raw1538	-->	sdxi18
cust_0_44	-->	raw814	-->	sdcm6		cust_0_769	-->	raw1539	-->	sdxi19
cust_0_45	-->	raw815	-->	sdcm7		icust2_0_128	-->	raw2242	-->	sdxi1
cust_0_450	-->	raw1220	-->	sdcm8		iordr2_0_128	-->	raw2434	-->	sdxi2
cust_0_451	-->	raw1221	-->	sdcm9		temp_0_128	-->	raw2626	-->	sdxi3
icust2_0_22	-->	raw2136	-->	sdcm1		stok_0_516	-->	raw517	-->	sdxi11
iordr2_0_22	-->	raw2328	-->	sdcm2		cust_0_773	-->	raw1543	-->	sdxi10
temp_0_22	-->	raw2520	-->	sdcm3		stok_0_517	-->	raw518	-->	sdxi12
stok_0_92	-->	raw93	-->	sdcm5		stok_0_518	-->	raw519	-->	sdxi13
cust_0_455	-->	raw1225	-->	sdcm10		stok_0_519	-->	raw520	-->	sdxi15
stok_0_93	-->	raw94	-->	sdcm2		cust_0_258	-->	raw1028	-->	sdxi16
stok_0_94	-->	raw95	-->	sdcm3		cust_0_259	-->	raw1029	-->	sdxi17
stok_0_95	-->	raw96	-->	sdcm5		cust_0_771	-->	raw1541	-->	sdxi18
cust_0_46	-->	raw816	-->	sdcm6		cust_0_772	-->	raw1542	-->	sdxi19
cust_0_47	-->	raw817	-->	sdcm7		icust2_0_129	-->	raw2243	-->	sdxi1
cust_0_453	-->	raw1223	-->	sdcm8		iordr2_0_129	-->	raw2435	-->	sdxi2
cust_0_454	-->	raw1224	-->	sdcm9		temp_0_129	-->	raw2627	-->	sdxi3
icust2_0_23	-->	raw2137	-->	sdcm1		stok_0_520	-->	raw521	-->	sdxi11
iordr2_0_23	-->	raw2329	-->	sdcm2		cust_0_776	-->	raw1546	-->	sdxi10
temp_0_23	-->	raw2521	-->	sdcm3		stok_0_521	-->	raw522	-->	sdxi12
stok_0_96	-->	raw97	-->	sdcm5		stok_0_522	-->	raw523	-->	sdxi13
cust_0_458	-->	raw1228	-->	sdcm10		stok_0_523	-->	raw524	-->	sdxi15
stok_0_97	-->	raw98	-->	sdcm2		cust_0_260	-->	raw1030	-->	sdxi16
stok_0_98	-->	raw99	-->	sdcm3		cust_0_261	-->	raw1031	-->	sdxi17
stok_0_99	-->	raw100	-->	sdcm5		cust_0_774	-->	raw1544	-->	sdxi18
cust_0_48	-->	raw818	-->	sdcm6		cust_0_775	-->	raw1545	-->	sdxi19
cust_0_49	-->	raw819	-->	sdcm7		icust2_0_130	-->	raw2244	-->	sdxi1
cust_0_456	-->	raw1226	-->	sdcm8		iordr2_0_130	-->	raw2436	-->	sdxi2
cust_0_457	-->	raw1227	-->	sdcm9		temp_0_130	-->	raw2628	-->	sdxi3
icust2_0_24	-->	raw2138	-->	sdcm1		stok_0_524	-->	raw525	-->	sdxi11
iordr2_0_24	-->	raw2330	-->	sdcm2		cust_0_779	-->	raw1549	-->	sdxi10
temp_0_24	-->	raw2522	-->	sdcm3		stok_0_525	-->	raw526	-->	sdxi12
stok_0_100	-->	raw101	-->	sdcm5		stok_0_526	-->	raw527	-->	sdxi13
cust_0_461	-->	raw1231	-->	sdcm10		stok_0_527	-->	raw528	-->	sdxi15
stok_0_101	-->	raw102	-->	sdcm2		cust_0_262	-->	raw1032	-->	sdxi16
stok_0_102	-->	raw103	-->	sdcm3		cust_0_263	-->	raw1033	-->	sdxi17

stok_0_103	-->	raw104	-->	sdcy5		cust_0_777	-->	raw1547	-->	sdyc8
cust_0_50	-->	raw820	-->	sdcy6		cust_0_778	-->	raw1548	-->	sdyc9
cust_0_51	-->	raw821	-->	sdcy7		icust2_0_131	-->	raw2245	-->	sdyc1
cust_0_459	-->	raw1229	-->	sdcy8		iordr2_0_131	-->	raw2437	-->	sdyc2
cust_0_460	-->	raw1230	-->	sdcy9		temp_0_131	-->	raw2629	-->	sdyc3
icust2_0_25	-->	raw2139	-->	sdcz1		stok_0_528	-->	raw529	-->	sdyc4
iordr2_0_25	-->	raw2331	-->	sdcz2		cust_0_782	-->	raw1552	-->	sdyc5
temp_0_25	-->	raw2523	-->	sdcz3		stok_0_529	-->	raw530	-->	sdyc6
icust2_0_0	-->	raw2114	-->	sdd1		stok_0_530	-->	raw531	-->	sdyc7
iordr2_0_0	-->	raw2306	-->	sdd2		stok_0_531	-->	raw532	-->	sdyc8
temp_0_0	-->	raw2498	-->	sdd3		cust_0_264	-->	raw1034	-->	sdyc9
icust1_0_0	-->	raw2690	-->	sdd5		cust_0_265	-->	raw1035	-->	sdyc10
stok_0_104	-->	raw105	-->	sddc1		cust_0_780	-->	raw1550	-->	sdyc11
cust_0_464	-->	raw1234	-->	sddc10		cust_0_781	-->	raw1551	-->	sdyc12
stok_0_105	-->	raw106	-->	sddc2		icust2_0_132	-->	raw2246	-->	sdyc13
stok_0_106	-->	raw107	-->	sddc3		iordr2_0_132	-->	raw2438	-->	sdyc14
stok_0_107	-->	raw108	-->	sddc5		temp_0_132	-->	raw2630	-->	sdyc15
cust_0_52	-->	raw822	-->	sddc6		stok_0_532	-->	raw533	-->	sdyc16
cust_0_53	-->	raw823	-->	sddc7		cust_0_785	-->	raw1555	-->	sdyc17
cust_0_462	-->	raw1232	-->	sddc8		stok_0_533	-->	raw534	-->	sdyc18
cust_0_463	-->	raw1233	-->	sddc9		stok_0_534	-->	raw535	-->	sdyc19
icust2_0_26	-->	raw2140	-->	sddd1		stok_0_535	-->	raw536	-->	sdyc20
iordr2_0_26	-->	raw2332	-->	sddd2		cust_0_266	-->	raw1036	-->	sdyc21
temp_0_26	-->	raw2524	-->	sddd3		cust_0_267	-->	raw1037	-->	sdyc22
stok_0_108	-->	raw109	-->	sddg1		cust_0_783	-->	raw1553	-->	sdyc23
cust_0_467	-->	raw1237	-->	sddg10		cust_0_784	-->	raw1554	-->	sdyc24
stok_0_109	-->	raw110	-->	sddg2		icust2_0_133	-->	raw2247	-->	sdyc25
stok_0_110	-->	raw111	-->	sddg3		iordr2_0_133	-->	raw2439	-->	sdyc26
stok_0_111	-->	raw112	-->	sddg5		temp_0_133	-->	raw2631	-->	sdyc27
cust_0_54	-->	raw824	-->	sddg6		stok_0_536	-->	raw537	-->	sdyc28
cust_0_55	-->	raw825	-->	sddg7		cust_0_788	-->	raw1558	-->	sdyc29
cust_0_465	-->	raw1235	-->	sddg8		stok_0_537	-->	raw538	-->	sdyc30
cust_0_466	-->	raw1236	-->	sddg9		stok_0_538	-->	raw539	-->	sdyc31
icust2_0_27	-->	raw2141	-->	sddh1		stok_0_539	-->	raw540	-->	sdyc32
iordr2_0_27	-->	raw2333	-->	sddh2		cust_0_268	-->	raw1038	-->	sdyc33
temp_0_27	-->	raw2525	-->	sddh3		cust_0_269	-->	raw1039	-->	sdyc34
stok_0_112	-->	raw113	-->	sddk1		cust_0_786	-->	raw1556	-->	sdyc35
cust_0_470	-->	raw1240	-->	sddk10		cust_0_787	-->	raw1557	-->	sdyc36
stok_0_113	-->	raw114	-->	sddk2		icust2_0_134	-->	raw2248	-->	sdyc37
stok_0_114	-->	raw115	-->	sddk3		iordr2_0_134	-->	raw2440	-->	sdyc38
stok_0_115	-->	raw116	-->	sddk5		temp_0_134	-->	raw2632	-->	sdyc39
cust_0_56	-->	raw826	-->	sddk6		stok_0_540	-->	raw541	-->	sdyc40
cust_0_57	-->	raw827	-->	sddk7		cust_0_791	-->	raw1561	-->	sdyc41
cust_0_468	-->	raw1238	-->	sddk8		stok_0_541	-->	raw542	-->	sdyc42
cust_0_469	-->	raw1239	-->	sddk9		stok_0_542	-->	raw543	-->	sdyc43
icust2_0_28	-->	raw2142	-->	sddl1		stok_0_543	-->	raw544	-->	sdyc44
iordr2_0_28	-->	raw2334	-->	sddl2		cust_0_270	-->	raw1040	-->	sdyc45
temp_0_28	-->	raw2526	-->	sddl3		cust_0_271	-->	raw1041	-->	sdyc46
stok_0_116	-->	raw117	-->	sddo1		cust_0_789	-->	raw1559	-->	sdyc47
cust_0_473	-->	raw1243	-->	sddo10		cust_0_790	-->	raw1560	-->	sdyc48
stok_0_117	-->	raw118	-->	sddo2		icust2_0_135	-->	raw2249	-->	sdyc49
stok_0_118	-->	raw119	-->	sddo3		iordr2_0_135	-->	raw2441	-->	sdyc50
stok_0_119	-->	raw120	-->	sddo5		temp_0_135	-->	raw2633	-->	sdyc51
cust_0_58	-->	raw828	-->	sddo6		stok_0_544	-->	raw545	-->	sdyc52
cust_0_59	-->	raw829	-->	sddo7		cust_0_794	-->	raw1564	-->	sdyc53
cust_0_471	-->	raw1241	-->	sddo8		stok_0_545	-->	raw546	-->	sdyc54
cust_0_472	-->	raw1242	-->	sddo9		stok_0_546	-->	raw547	-->	sdyc55
icust2_0_29	-->	raw2143	-->	sddp1		stok_0_547	-->	raw548	-->	sdyc56
iordr2_0_29	-->	raw2335	-->	sddp2		cust_0_272	-->	raw1042	-->	sdyc57
temp_0_29	-->	raw2527	-->	sddp3		cust_0_273	-->	raw1043	-->	sdyc58
stok_0_120	-->	raw121	-->	sdds1		cust_0_792	-->	raw1562	-->	sdyc59
cust_0_476	-->	raw1246	-->	sdds10		cust_0_793	-->	raw1563	-->	sdyc60
stok_0_121	-->	raw122	-->	sdds2		icust2_0_136	-->	raw2250	-->	sdyc61
stok_0_122	-->	raw123	-->	sdds3		iordr2_0_136	-->	raw2442	-->	sdyc62
stok_0_123	-->	raw124	-->	sdds5		temp_0_136	-->	raw2634	-->	sdyc63

cust_0_60	-->	raw830	-->	sdds6		stok_0_548	-->	raw549	-->	sdyz1
cust_0_61	-->	raw831	-->	sdds7		cust_0_797	-->	raw1567	-->	sdyz10
cust_0_474	-->	raw1244	-->	sdds8		stok_0_549	-->	raw550	-->	sdyz2
cust_0_475	-->	raw1245	-->	sdds9		stok_0_550	-->	raw551	-->	sdyz3
icust2_0_30	-->	raw2144	-->	sddt1		stok_0_551	-->	raw552	-->	sdyz5
iordr2_0_30	-->	raw2336	-->	sddt2		cust_0_274	-->	raw1044	-->	sdyz6
temp_0_30	-->	raw2528	-->	sddt3		cust_0_275	-->	raw1045	-->	sdyz7
stok_0_124	-->	raw125	-->	sddw1		cust_0_795	-->	raw1565	-->	sdyz8
cust_0_479	-->	raw1249	-->	sddw10		cust_0_796	-->	raw1566	-->	sdyz9
stok_0_125	-->	raw126	-->	sddw2		icust2_0_137	-->	raw2251	-->	sdza1
stok_0_126	-->	raw127	-->	sddw3		iordr2_0_137	-->	raw2443	-->	sdza2
stok_0_127	-->	raw128	-->	sddw5		temp_0_137	-->	raw2635	-->	sdza3
cust_0_62	-->	raw832	-->	sddw6		stok_0_552	-->	raw553	-->	sdzd1
cust_0_63	-->	raw833	-->	sddw7		cust_0_800	-->	raw1570	-->	sdzd10
cust_0_477	-->	raw1247	-->	sddw8		stok_0_553	-->	raw554	-->	sdzd2
cust_0_478	-->	raw1248	-->	sddw9		stok_0_554	-->	raw555	-->	sdzd3
icust2_0_31	-->	raw2145	-->	sddx1		stok_0_555	-->	raw556	-->	sdzd5
iordr2_0_31	-->	raw2337	-->	sddx2		cust_0_276	-->	raw1046	-->	sdzd6
temp_0_31	-->	raw2529	-->	sddx3		cust_0_277	-->	raw1047	-->	sdzd7
stok_0_128	-->	raw129	-->	sdea1		cust_0_798	-->	raw1568	-->	sdzd8
cust_0_482	-->	raw1252	-->	sdea10		cust_0_799	-->	raw1569	-->	sdzd9
stok_0_129	-->	raw130	-->	sdea2		icust2_0_138	-->	raw2252	-->	sdze1
stok_0_130	-->	raw131	-->	sdea3		iordr2_0_138	-->	raw2444	-->	sdze2
stok_0_131	-->	raw132	-->	sdea5		temp_0_138	-->	raw2636	-->	sdze3
cust_0_64	-->	raw834	-->	sdea6		stok_0_556	-->	raw557	-->	sdzh1
cust_0_65	-->	raw835	-->	sdea7		cust_0_803	-->	raw1573	-->	sdzh10
cust_0_480	-->	raw1250	-->	sdea8		stok_0_557	-->	raw558	-->	sdzh2
cust_0_481	-->	raw1251	-->	sdea9		stok_0_558	-->	raw559	-->	sdzh3
icust2_0_32	-->	raw2146	-->	sdeb1		stok_0_559	-->	raw560	-->	sdzh5
iordr2_0_32	-->	raw2338	-->	sdeb2		cust_0_278	-->	raw1048	-->	sdzh6
temp_0_32	-->	raw2530	-->	sdeb3		cust_0_279	-->	raw1049	-->	sdzh7
stok_0_132	-->	raw133	-->	sdee1		cust_0_801	-->	raw1571	-->	sdzh8
cust_0_485	-->	raw1255	-->	sdee10		cust_0_802	-->	raw1572	-->	sdzh9
stok_0_133	-->	raw134	-->	sdee2		icust2_0_139	-->	raw2253	-->	sdzi1
stok_0_134	-->	raw135	-->	sdee3		iordr2_0_139	-->	raw2445	-->	sdzi2
stok_0_135	-->	raw136	-->	sdee5		temp_0_139	-->	raw2637	-->	sdzi3
cust_0_66	-->	raw836	-->	sdee6		stok_0_560	-->	raw561	-->	sdzl1
cust_0_67	-->	raw837	-->	sdee7		cust_0_806	-->	raw1576	-->	sdzl10
cust_0_483	-->	raw1253	-->	sdee8		stok_0_561	-->	raw562	-->	sdzl2
cust_0_484	-->	raw1254	-->	sdee9		stok_0_562	-->	raw563	-->	sdzl3
icust2_0_33	-->	raw2147	-->	sdef1		stok_0_563	-->	raw564	-->	sdzl5
iordr2_0_33	-->	raw2339	-->	sdef2		stok_0_564	-->	raw1050	-->	sdzl6
temp_0_33	-->	raw2531	-->	sdef3		cust_0_280	-->	raw1051	-->	sdzl7
stok_0_136	-->	raw137	-->	sdei1		cust_0_281	-->	raw1051	-->	sdzl7
cust_0_488	-->	raw1258	-->	sdei10		cust_0_804	-->	raw1574	-->	sdzl8
stok_0_137	-->	raw138	-->	sdei2		cust_0_805	-->	raw1575	-->	sdzl9
stok_0_138	-->	raw139	-->	sdei3		icust2_0_140	-->	raw2254	-->	sdzm1
stok_0_139	-->	raw140	-->	sdei5		iordr2_0_140	-->	raw2446	-->	sdzm2
cust_0_68	-->	raw838	-->	sdei6		temp_0_140	-->	raw2638	-->	sdzm3
cust_0_69	-->	raw839	-->	sdei7		stok_0_564	-->	raw565	-->	sdzp1
cust_0_486	-->	raw1256	-->	sdei8		cust_0_809	-->	raw1579	-->	sdzp10
cust_0_487	-->	raw1257	-->	sdei9		stok_0_565	-->	raw566	-->	sdzp2
icust2_0_34	-->	raw2148	-->	sdej1		stok_0_566	-->	raw567	-->	sdzp3
iordr2_0_34	-->	raw2340	-->	sdej2		stok_0_567	-->	raw568	-->	sdzp5
temp_0_34	-->	raw2532	-->	sdej3		cust_0_282	-->	raw1052	-->	sdzp6
stok_0_140	-->	raw141	-->	sdem1		cust_0_283	-->	raw1053	-->	sdzp7
cust_0_491	-->	raw1261	-->	sdem10		cust_0_807	-->	raw1577	-->	sdzp8
stok_0_141	-->	raw142	-->	sdem2		cust_0_808	-->	raw1578	-->	sdzp9
stok_0_142	-->	raw143	-->	sdem3		icust2_0_141	-->	raw2255	-->	sdzq1
stok_0_143	-->	raw144	-->	sdem5		iordr2_0_141	-->	raw2447	-->	sdzq2
cust_0_70	-->	raw840	-->	sdem6		temp_0_141	-->	raw2639	-->	sdzq3
cust_0_71	-->	raw841	-->	sdem7		stok_0_568	-->	raw569	-->	sdzt1
cust_0_489	-->	raw1259	-->	sdem8		cust_0_812	-->	raw1582	-->	sdzt10
cust_0_490	-->	raw1260	-->	sdem9		stok_0_569	-->	raw570	-->	sdzt2
icust2_0_35	-->	raw2149	-->	sdem1		stok_0_570	-->	raw571	-->	sdzt3
						stok_0_571	-->	raw572	-->	sdzt5

iordr2_0_35	-->	raw2341	-->	sden2		cust_0_284	-->	raw1054	-->	sdzt6
temp_0_35	-->	raw2533	-->	sden3		cust_0_285	-->	raw1055	-->	sdzt7
stok_0_144	-->	raw145	-->	sdeu1		cust_0_810	-->	raw1580	-->	sdzt8
cust_0_494	-->	raw1264	-->	sdeu10		cust_0_811	-->	raw1581	-->	sdzt9
stok_0_145	-->	raw146	-->	sdeu2		icust2_0_142	-->	raw2256	-->	sdzu1
stok_0_146	-->	raw147	-->	sdeu3		iordr2_0_142	-->	raw2448	-->	sdzu2
stok_0_147	-->	raw148	-->	sdeu5		temp_0_142	-->	raw2640	-->	sdzu3
cust_0_72	-->	raw842	-->	sdeu6		stok_0_572	-->	raw573	-->	sdzx1
cust_0_73	-->	raw843	-->	sdeu7		cust_0_815	-->	raw1585	-->	sdzx10
cust_0_492	-->	raw1262	-->	sdeu8		stok_0_573	-->	raw574	-->	sdzx2
cust_0_493	-->	raw1263	-->	sdeu9		stok_0_574	-->	raw575	-->	sdzx3
icust2_0_36	-->	raw2150	-->	sdev1		stok_0_575	-->	raw576	-->	sdzx5
iordr2_0_36	-->	raw2342	-->	sdev2		cust_0_286	-->	raw1056	-->	sdzx6
temp_0_36	-->	raw2534	-->	sdev3		cust_0_287	-->	raw1057	-->	sdzx7
stok_0_148	-->	raw149	-->	sdey1		cust_0_813	-->	raw1583	-->	sdzx8
cust_0_497	-->	raw1267	-->	sdey10		cust_0_814	-->	raw1584	-->	sdzx9
stok_0_149	-->	raw150	-->	sdey2		icust2_0_143	-->	raw2257	-->	sdzy1
stok_0_150	-->	raw151	-->	sdey3		iordr2_0_143	-->	raw2449	-->	sdzy2
stok_0_151	-->	raw152	-->	sdey5		temp_0_143	-->	raw2641	-->	sdzy3

sdc	-->	scsi 16	lun 0		sdom	-->	scsi 56	lun 0
sdd	-->	scsi 16	lun 1		sdon	-->	scsi 56	lun 1
sdg	-->	scsi 16	lun 4		sdoq	-->	scsi 56	lun 4
sdh	-->	scsi 16	lun 5		sdor	-->	scsi 56	lun 5
sdk	-->	scsi 16	lun 8		sdou	-->	scsi 56	lun 8
sdll	-->	scsi 16	lun 9		sdov	-->	scsi 56	lun 9
sdo	-->	scsi 17	lun 0		sdox	-->	scsi 57	lun 0
sdp	-->	scsi 17	lun 1		sdoz	-->	scsi 57	lun 1
sds	-->	scsi 17	lun 4		sdpc	-->	scsi 57	lun 4
sdt	-->	scsi 17	lun 5		sdpd	-->	scsi 57	lun 5
sdw	-->	scsi 17	lun 8		sdpg	-->	scsi 57	lun 8
sdx	-->	scsi 17	lun 9		sdph	-->	scsi 57	lun 9
sdaa	-->	scsi 18	lun 0		sdpk	-->	scsi 58	lun 0
sdab	-->	scsi 18	lun 1		sdpl	-->	scsi 58	lun 1
sdae	-->	scsi 18	lun 4		sdpo	-->	scsi 58	lun 4
sdaf	-->	scsi 18	lun 5		sdpp	-->	scsi 58	lun 5
sdai	-->	scsi 18	lun 8		sdps	-->	scsi 58	lun 8
sdaj	-->	scsi 18	lun 9		sdpt	-->	scsi 58	lun 9
sdam	-->	scsi 19	lun 0		sdpw	-->	scsi 59	lun 0
sdan	-->	scsi 19	lun 1		sdpx	-->	scsi 59	lun 1
sdaq	-->	scsi 19	lun 4		sdqa	-->	scsi 59	lun 4
sdar	-->	scsi 19	lun 5		sdqb	-->	scsi 59	lun 5
sdau	-->	scsi 19	lun 8		sdqe	-->	scsi 59	lun 8
sdav	-->	scsi 19	lun 9		sdqf	-->	scsi 59	lun 9
sday	-->	scsi 20	lun 0		sdqi	-->	scsi 60	lun 0
sdaz	-->	scsi 20	lun 1		sdqj	-->	scsi 60	lun 1
sdbc	-->	scsi 20	lun 4		sdqm	-->	scsi 60	lun 4
sdbd	-->	scsi 20	lun 5		sdqn	-->	scsi 60	lun 5
sdbg	-->	scsi 20	lun 8		sdqq	-->	scsi 60	lun 8
sdbh	-->	scsi 20	lun 9		sdqr	-->	scsi 60	lun 9
sdbk	-->	scsi 21	lun 0		sdqu	-->	scsi 61	lun 0
sdbl	-->	scsi 21	lun 1		sdqv	-->	scsi 61	lun 1
sdbo	-->	scsi 21	lun 4		sdqy	-->	scsi 61	lun 4
sdbp	-->	scsi 21	lun 5		sdqz	-->	scsi 61	lun 5
sdbx	-->	scsi 21	lun 8		sdrc	-->	scsi 61	lun 8
sdbc	-->	scsi 21	lun 9		sdrd	-->	scsi 61	lun 9
sdbw	-->	scsi 22	lun 0		sdrh	-->	scsi 62	lun 0
sdbx	-->	scsi 22	lun 1		sdrh	-->	scsi 62	lun 1
sdca	-->	scsi 22	lun 4		sdrk	-->	scsi 62	lun 4
sdcb	-->	scsi 22	lun 5		sdrk	-->	scsi 62	lun 4
sdc	-->	scsi 22	lun 8		sdrl	-->	scsi 62	lun 5
sdcf	-->	scsi 22	lun 9		sdro	-->	scsi 62	lun 8
sdcf	-->	scsi 22	lun 9		sdrp	-->	scsi 62	lun 9
sdc	-->	scsi 23	lun 0		sdrs	-->	scsi 63	lun 0
sdcj	-->	scsi 23	lun 1		sdrs	-->	scsi 63	lun 0
sdc	-->	scsi 23	lun 1		sdr	-->	scsi 63	lun 1
sdc	-->	scsi 23	lun 4		sdr	-->	scsi 63	lun 1
sdc	-->	scsi 23	lun 4		sdrw	-->	scsi 63	lun 4

sdcn	-->	scsi 23	lun 5		sdrx	-->	scsi 63	lun 5
sdcq	-->	scsi 23	lun 8		sdsa	-->	scsi 63	lun 8
sdcr	-->	scsi 23	lun 9		sdsb	-->	scsi 63	lun 9
sdcu	-->	scsi 24	lun 0		sdwj	-->	scsi 72	lun 0
sdcv	-->	scsi 24	lun 1		sdwk	-->	scsi 72	lun 1
sdcy	-->	scsi 24	lun 4		sdwn	-->	scsi 72	lun 4
sdcz	-->	scsi 24	lun 5		sdwo	-->	scsi 72	lun 5
sddc	-->	scsi 24	lun 8		sdwr	-->	scsi 72	lun 8
sddd	-->	scsi 24	lun 9		sdws	-->	scsi 72	lun 9
sddg	-->	scsi 25	lun 0		sdwv	-->	scsi 73	lun 0
sddh	-->	scsi 25	lun 1		sdww	-->	scsi 73	lun 1
sddk	-->	scsi 25	lun 4		sdwz	-->	scsi 73	lun 4
sddl	-->	scsi 25	lun 5		sdx	-->	scsi 73	lun 5
sddo	-->	scsi 25	lun 8		sdx	-->	scsi 73	lun 8
sddp	-->	scsi 25	lun 9		sdx	-->	scsi 73	lun 9
sdds	-->	scsi 26	lun 0		sdxh	-->	scsi 74	lun 0
sddt	-->	scsi 26	lun 1		sdx	-->	scsi 74	lun 1
sddw	-->	scsi 26	lun 4		sdxl	-->	scsi 74	lun 4
sddx	-->	scsi 26	lun 5		sdxm	-->	scsi 74	lun 5
sdea	-->	scsi 26	lun 8		sdxp	-->	scsi 74	lun 8
sdeb	-->	scsi 26	lun 9		sdxq	-->	scsi 74	lun 9
sdee	-->	scsi 27	lun 0		sdx	-->	scsi 75	lun 0
sdef	-->	scsi 27	lun 1		sdxu	-->	scsi 75	lun 1
sdei	-->	scsi 27	lun 4		sdx	-->	scsi 75	lun 4
sdej	-->	scsi 27	lun 5		sdx	-->	scsi 75	lun 5
sdem	-->	scsi 27	lun 8		sdyb	-->	scsi 75	lun 8
sden	-->	scsi 27	lun 9		sdyc	-->	scsi 75	lun 9
sdeu	-->	scsi 30	lun 0		sdyf	-->	scsi 76	lun 0
sdev	-->	scsi 30	lun 1		sdyg	-->	scsi 76	lun 1
sdey	-->	scsi 30	lun 4		sdyj	-->	scsi 76	lun 4
sdez	-->	scsi 30	lun 5		sdyk	-->	scsi 76	lun 5
sdfc	-->	scsi 30	lun 8		sdyn	-->	scsi 76	lun 8
sdfd	-->	scsi 30	lun 9		sdyo	-->	scsi 76	lun 9
sdfg	-->	scsi 31	lun 0		sdyr	-->	scsi 77	lun 0
sdfh	-->	scsi 31	lun 1		sdys	-->	scsi 77	lun 1
sdfk	-->	scsi 31	lun 4		sdyv	-->	scsi 77	lun 4
sdf	-->	scsi 31	lun 5		sdyw	-->	scsi 77	lun 5
sdf	-->	scsi 31	lun 8		sdyz	-->	scsi 77	lun 8
sdfp	-->	scsi 31	lun 9		sdza	-->	scsi 77	lun 9
sdfs	-->	scsi 32	lun 0		sdzd	-->	scsi 78	lun 0
sdf	-->	scsi 32	lun 1		sdze	-->	scsi 78	lun 1
sdfw	-->	scsi 32	lun 4		sdzh	-->	scsi 78	lun 4
sdfx	-->	scsi 32	lun 5		sdzi	-->	scsi 78	lun 5
sdga	-->	scsi 32	lun 8		sdzl	-->	scsi 78	lun 8
sdgb	-->	scsi 32	lun 9		sdzm	-->	scsi 78	lun 9
sdge	-->	scsi 33	lun 0		sdzp	-->	scsi 79	lun 0
sdgf	-->	scsi 33	lun 1		sdzq	-->	scsi 79	lun 1
sdgi	-->	scsi 33	lun 4		sdzt	-->	scsi 79	lun 4
sdgj	-->	scsi 33	lun 5		sdzu	-->	scsi 79	lun 5
sdgm	-->	scsi 33	lun 8		sdzx	-->	scsi 79	lun 8
sdgn	-->	scsi 33	lun 9		sdzy	-->	scsi 79	lun 9
sdhc	-->	scsi 40	lun 0		sdaab	-->	scsi 80	lun 0
sdhd	-->	scsi 40	lun 1		sdaac	-->	scsi 80	lun 1
sdhg	-->	scsi 40	lun 4		sdaaf	-->	scsi 80	lun 4
sdhh	-->	scsi 40	lun 5		sdaag	-->	scsi 80	lun 5
sdhk	-->	scsi 40	lun 8		sdaaj	-->	scsi 80	lun 8
sdhl	-->	scsi 40	lun 9		sdaak	-->	scsi 80	lun 9
sdho	-->	scsi 41	lun 0		sdaan	-->	scsi 81	lun 0
sdhp	-->	scsi 41	lun 1		sdaao	-->	scsi 81	lun 1
sdhs	-->	scsi 41	lun 4		sdaar	-->	scsi 81	lun 4
sdht	-->	scsi 41	lun 5		sdaas	-->	scsi 81	lun 5
sdhw	-->	scsi 41	lun 8		sdaav	-->	scsi 81	lun 8
sdhx	-->	scsi 41	lun 9		sdaaw	-->	scsi 81	lun 9
sdia	-->	scsi 42	lun 0		sdaaz	-->	scsi 82	lun 0
sdib	-->	scsi 42	lun 1		sdaba	-->	scsi 82	lun 1

side	-->	scsi	42	lun	4		sdabd	-->	scsi	82	lun	4
sdif	-->	scsi	42	lun	5		sdabe	-->	scsi	82	lun	5
sdii	-->	scsi	42	lun	8		sdabh	-->	scsi	82	lun	8
sdij	-->	scsi	42	lun	9		sdabi	-->	scsi	82	lun	9
sdim	-->	scsi	43	lun	0		sdabl	-->	scsi	83	lun	0
sdin	-->	scsi	43	lun	1		sdabm	-->	scsi	83	lun	1
sdiqu	-->	scsi	43	lun	4		sdabp	-->	scsi	83	lun	4
sdir	-->	scsi	43	lun	5		sdabq	-->	scsi	83	lun	5
sdiu	-->	scsi	43	lun	8		sdabt	-->	scsi	83	lun	8
sdiv	-->	scsi	43	lun	9		sdabu	-->	scsi	83	lun	9
sdiy	-->	scsi	44	lun	0		sdabx	-->	scsi	84	lun	0
sdiz	-->	scsi	44	lun	1		sdaby	-->	scsi	84	lun	1
sdjc	-->	scsi	44	lun	4		sdacb	-->	scsi	84	lun	4
sdjd	-->	scsi	44	lun	5		sdacc	-->	scsi	84	lun	5
sdjg	-->	scsi	44	lun	8		sdacf	-->	scsi	84	lun	8
sdjh	-->	scsi	44	lun	9		sdacg	-->	scsi	84	lun	9
sdjk	-->	scsi	45	lun	0		sdacj	-->	scsi	85	lun	0
sdjl	-->	scsi	45	lun	1		sdack	-->	scsi	85	lun	1
sdjo	-->	scsi	45	lun	4		sdacn	-->	scsi	85	lun	4
sdjp	-->	scsi	45	lun	5		sdaco	-->	scsi	85	lun	5
sdjs	-->	scsi	45	lun	8		sdacr	-->	scsi	85	lun	8
sdjt	-->	scsi	45	lun	9		sdacs	-->	scsi	85	lun	9
sdjw	-->	scsi	46	lun	0		sdacv	-->	scsi	86	lun	0
sdjx	-->	scsi	46	lun	1		sdacw	-->	scsi	86	lun	1
sdka	-->	scsi	46	lun	4		sdacz	-->	scsi	86	lun	4
sdkb	-->	scsi	46	lun	5		sdada	-->	scsi	86	lun	5
sdke	-->	scsi	46	lun	8		sdadd	-->	scsi	86	lun	8
sdkf	-->	scsi	46	lun	9		sdade	-->	scsi	86	lun	9
sdki	-->	scsi	47	lun	0		sdadh	-->	scsi	87	lun	0
sdkj	-->	scsi	47	lun	1		sdadi	-->	scsi	87	lun	1
sdkm	-->	scsi	47	lun	4		sdadl	-->	scsi	87	lun	4
sdkn	-->	scsi	47	lun	5		sdadm	-->	scsi	87	lun	5
sdkq	-->	scsi	47	lun	8		sdadp	-->	scsi	87	lun	8
sdkr	-->	scsi	47	lun	9		sdadq	-->	scsi	87	lun	9
sdku	-->	scsi	48	lun	0		sdadt	-->	scsi	88	lun	0
sdkv	-->	scsi	48	lun	1		sdadu	-->	scsi	88	lun	1
sdky	-->	scsi	48	lun	4		sdadx	-->	scsi	88	lun	4
sdkz	-->	scsi	48	lun	5		sdady	-->	scsi	88	lun	5
sdlc	-->	scsi	48	lun	8		sdaeb	-->	scsi	88	lun	8
sdlc	-->	scsi	48	lun	9		sdaec	-->	scsi	88	lun	9
sdlg	-->	scsi	49	lun	0		sdaef	-->	scsi	89	lun	0
sdlh	-->	scsi	49	lun	1		sdaeg	-->	scsi	89	lun	1
sdlk	-->	scsi	49	lun	4		sdaej	-->	scsi	89	lun	4
sdlm	-->	scsi	49	lun	5		sdaek	-->	scsi	89	lun	5
sdlo	-->	scsi	49	lun	8		sdaen	-->	scsi	89	lun	8
sdlp	-->	scsi	49	lun	9		sdaeo	-->	scsi	89	lun	9
sdlr	-->	scsi	50	lun	0		sdaer	-->	scsi	90	lun	0
sdlw	-->	scsi	50	lun	1		sdaes	-->	scsi	90	lun	1
sdlx	-->	scsi	50	lun	4		sdaev	-->	scsi	90	lun	4
sdma	-->	scsi	50	lun	5		sdaew	-->	scsi	90	lun	5
sdmb	-->	scsi	50	lun	8		sdaez	-->	scsi	90	lun	8
sdmb	-->	scsi	50	lun	9		sdafa	-->	scsi	90	lun	9
sdme	-->	scsi	51	lun	0		sdafd	-->	scsi	91	lun	0
sdmf	-->	scsi	51	lun	1		sdafe	-->	scsi	91	lun	1
sdmi	-->	scsi	51	lun	4		sdafh	-->	scsi	91	lun	4
sdmj	-->	scsi	51	lun	5		sdafi	-->	scsi	91	lun	5
sdmm	-->	scsi	51	lun	8		sdafj	-->	scsi	91	lun	8
sdmn	-->	scsi	51	lun	9		sdafk	-->	scsi	91	lun	9
sdmq	-->	scsi	52	lun	0		sdafm	-->	scsi	91	lun	0
sdmr	-->	scsi	52	lun	0		sdafp	-->	scsi	92	lun	0
sdmr	-->	scsi	52	lun	1		sdafq	-->	scsi	92	lun	1
sdmu	-->	scsi	52	lun	4		sdafq	-->	scsi	92	lun	1
sdmv	-->	scsi	52	lun	4		sdافت	-->	scsi	92	lun	4
sdmv	-->	scsi	52	lun	5		sdafu	-->	scsi	92	lun	5
sdmy	-->	scsi	52	lun	8		sdafx	-->	scsi	92	lun	8
sdmz	-->	scsi	52	lun	9		sdafy	-->	scsi	92	lun	9
sdnc	-->	scsi	53	lun	0		sdagb	-->	scsi	93	lun	0

sdnd	-->	scsi 53	lun 1		sdagc	-->	scsi 93	lun 1
sdng	-->	scsi 53	lun 4		sdagf	-->	scsi 93	lun 4
sdnh	-->	scsi 53	lun 5		sdagg	-->	scsi 93	lun 5
sdnk	-->	scsi 53	lun 8		sdagj	-->	scsi 93	lun 8
sdnl	-->	scsi 53	lun 9		sdagk	-->	scsi 93	lun 9
sdno	-->	scsi 54	lun 0		sdagn	-->	scsi 94	lun 0
sdnp	-->	scsi 54	lun 1		sdago	-->	scsi 94	lun 1
sdns	-->	scsi 54	lun 4		sdagr	-->	scsi 94	lun 4
sdnt	-->	scsi 54	lun 5		sdags	-->	scsi 94	lun 5
sdnw	-->	scsi 54	lun 8		sdagv	-->	scsi 94	lun 8
sdnx	-->	scsi 54	lun 9		sdagw	-->	scsi 94	lun 9
sdoa	-->	scsi 55	lun 0		sdagz	-->	scsi 95	lun 0
sdob	-->	scsi 55	lun 1		sdaha	-->	scsi 95	lun 1
sdoe	-->	scsi 55	lun 4		sdahd	-->	scsi 95	lun 4
sdof	-->	scsi 55	lun 5		sdahe	-->	scsi 95	lun 5
sdoi	-->	scsi 55	lun 8		sdahh	-->	scsi 95	lun 8
sdoj	-->	scsi 55	lun 9		sdahi	-->	scsi 95	lun 9

4.3 Type of Database

A statement must be provided that describes:

1. *The data model implemented by DBMS used (e.g. relational, network, hierarchical).*
2. *The database interface (e.g. embedded, call level) and access language (e.g. SQL, DL/I, COBOL read/write used to implement the TPC-C transaction. 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 is a relational DBMS.

The interface used was Oracle stored procedures accessed using the Oracle Call Interface (OCI) embedded in C code.

4.4 Database Mapping

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

Horizontal partitioning was used for one (hist) of the tables and one (iordr2) of the indices. The detail of this partitioning can be understood by examining the table and index definition statements in Appendix E. Vertical partitioning and replications were not used in this implementation.

4.5 60 Day Space

Details of the 60 days 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).

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

- The size of the redo log was queried from the Oracle catalog.
- A full performance run was executed.
- The increase in size to the redo logs was divided by the number of transactions, giving bytes used per new order.
- This amount was multiplied by the reported tpm rate times 480 minutes, giving total space needed for 8 hours.

For the dynamic tables the following steps were followed:

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

The 60 day space requirement is shown in Appendix F.

Clause 5 Related Items

5.1 Throughput

Measured tpmC must be reported.

Measured tpmC: 2,196,268 tpmC
 Price per tpmC: \$4.70 USD per tpmC

5.2 Response Times

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

An emulation delay of 0.1 second is included in response time and menu time to compensate for browser delay.

Table 5.1 Response Times in Seconds

Type	Average	90th %	Maximum
New-Order	0.352	0.344	38.972
Payment	0.337	0.333	38.966
Order-Status	0.344	0.341	38.946
Interactive Delivery	0.103	0.104	0.325
Deferred Delivery	0.243	0.242	38.849
Stock-Level	0.330	0.325	38.922
Menu	0.103	0.104	0.363

5.3 Keying and Think Times

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

Table 5.2 Keying Times

Type	Minimum	Average	Maximum
New-Order	18.002	18.012	18.035
Payment	3.002	3.012	3.032
Order-Status	2.003	2.012	2.031
Interactive Delivery	2.002	2.012	2.031
Stock-Level	2.003	2.012	2.029

Table 5.3 Think Times

Type	Minimum	Average	Maximum
New-Order	0.000	12.016	120.201
Payment	0.000	12.016	120.201
Order-Status	0.000	10.019	100.198
Interactive Delivery	0.000	5.019	50.199
Stock-Level	0.000	5.019	50.198

5.4 Response Time Frequency Distribution Curves and Other Graphs

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

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

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

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

Figure 5.1: New Order Response Time Distribution

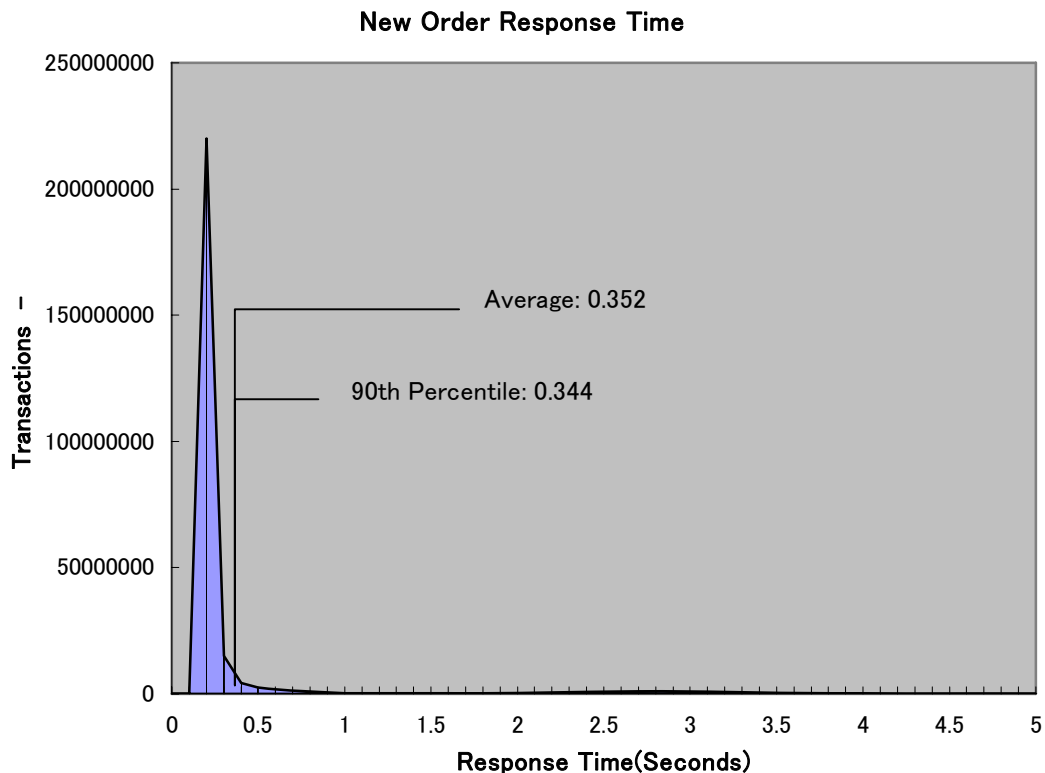


Figure 5.2: Payment Response Time Distribution

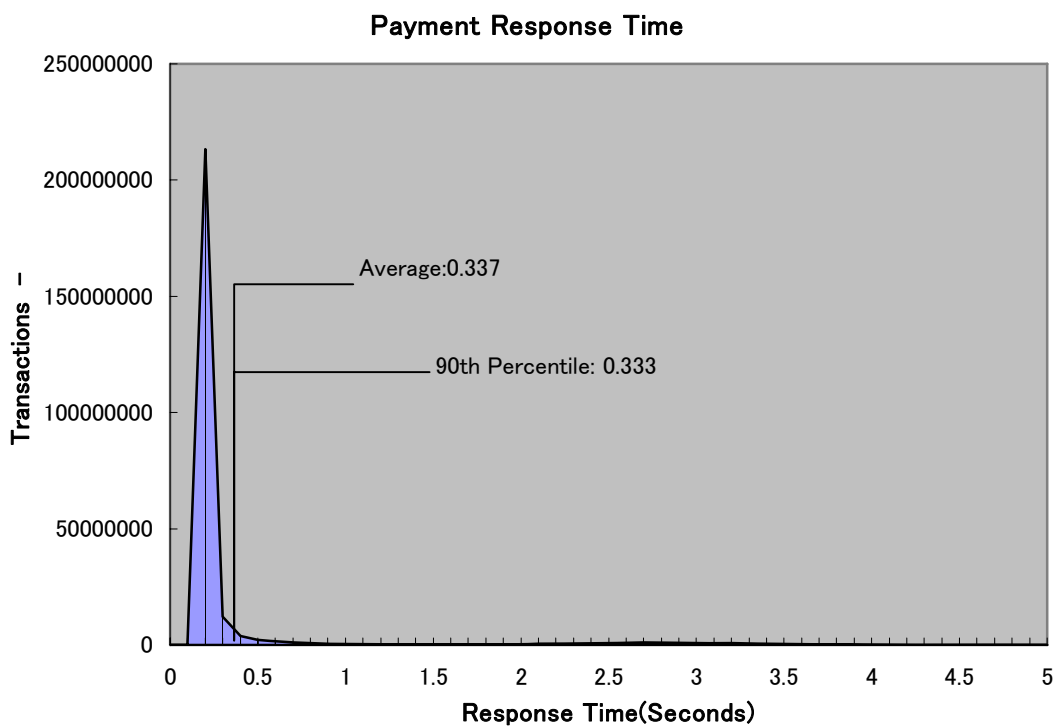


Figure 5.3: Order Status Response Time Distribution

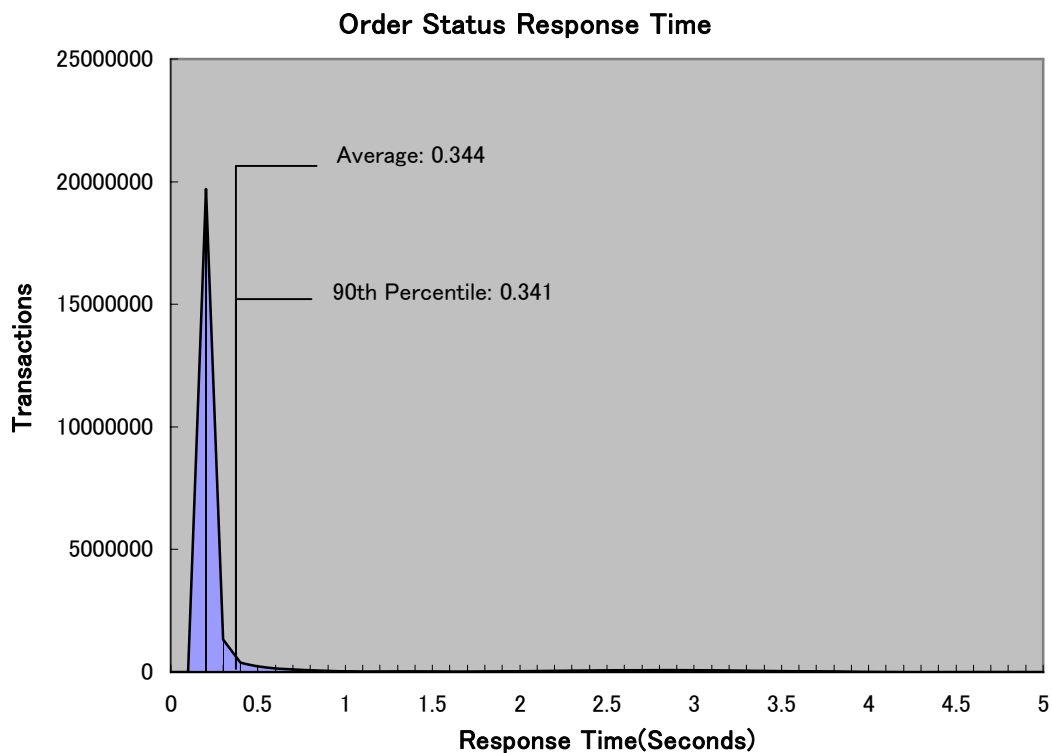


Figure 5.4: Delivery Response Time Distribution

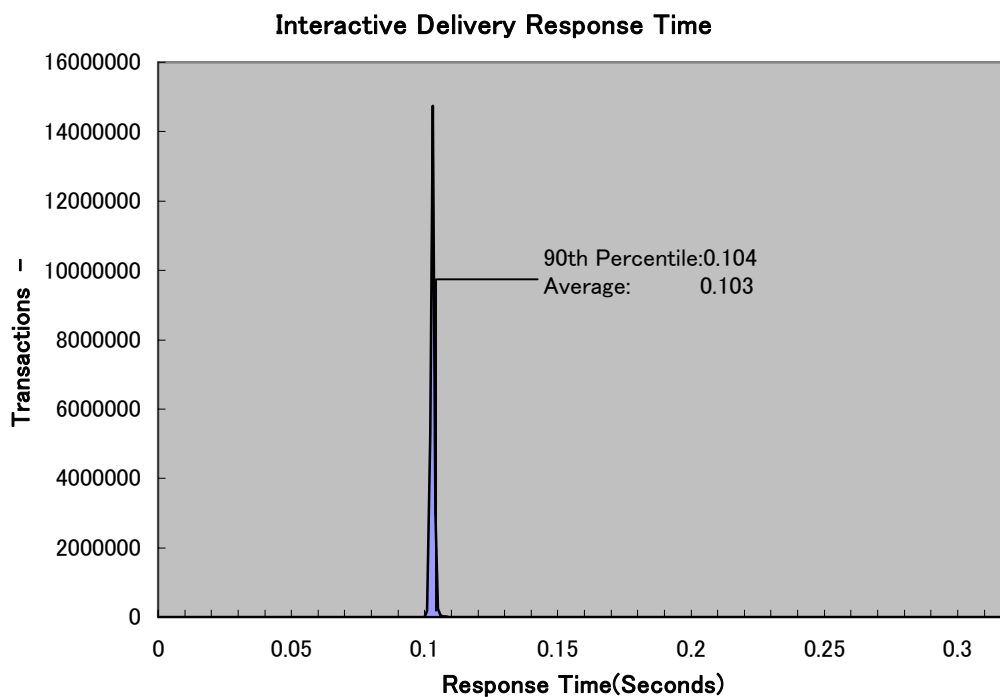


Figure 5.5: Stock Level Response Time Distribution

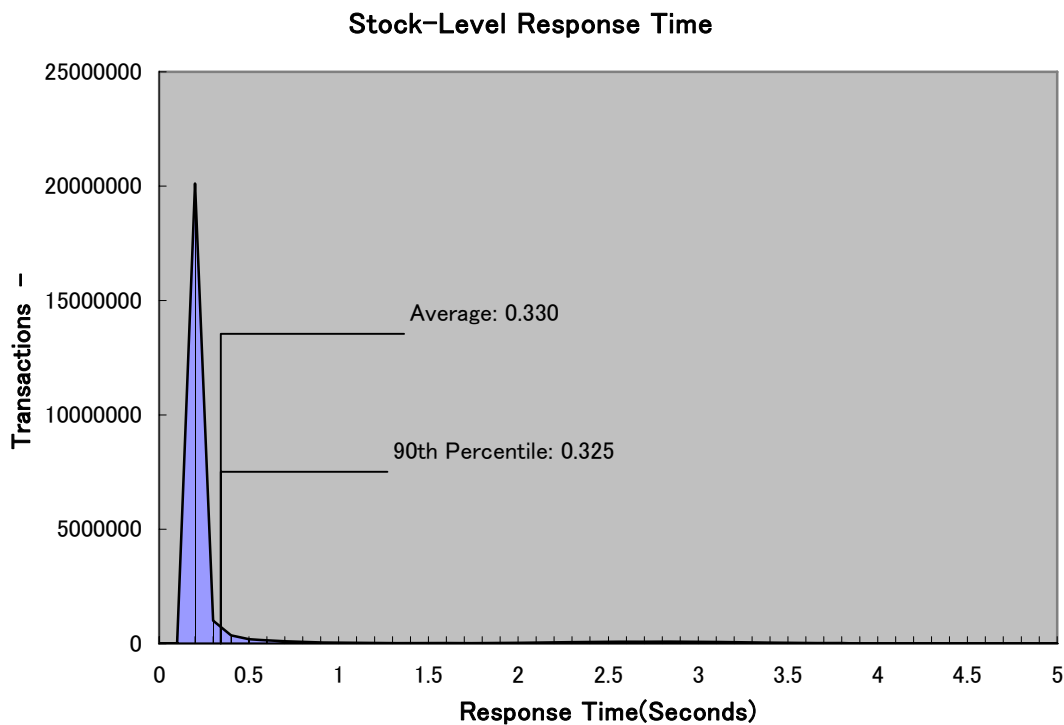


Figure 5.6: New Order Think Time Frequency Distribution

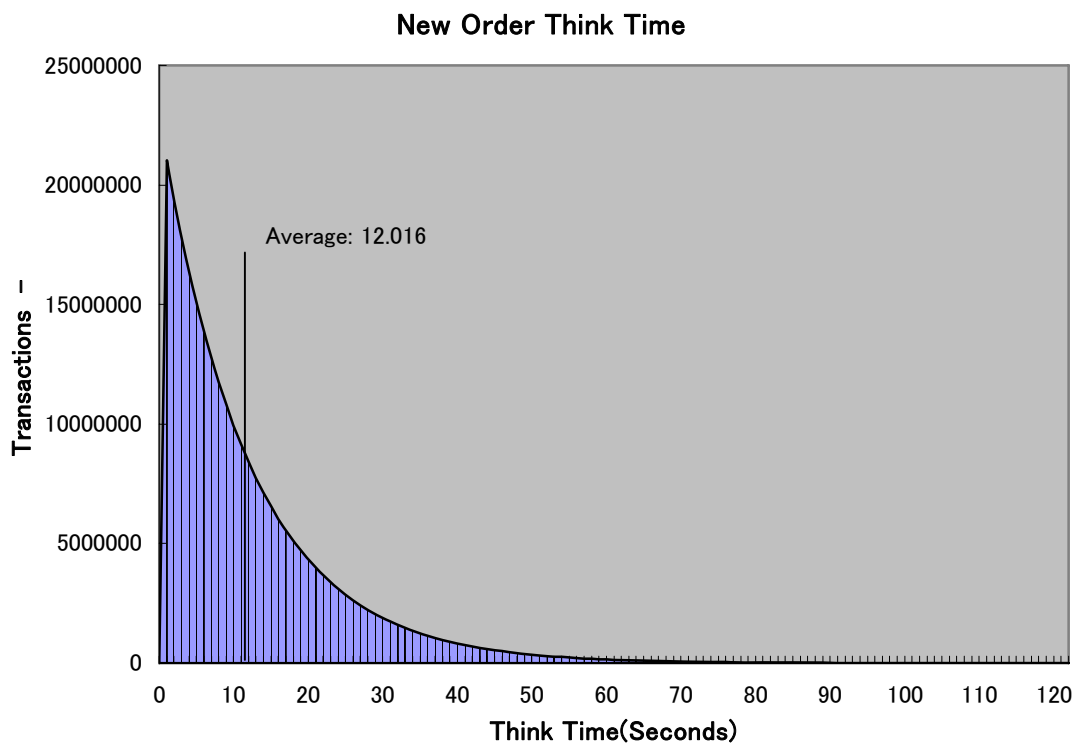


Figure 5.7: New-Order Response time vs. Throughput

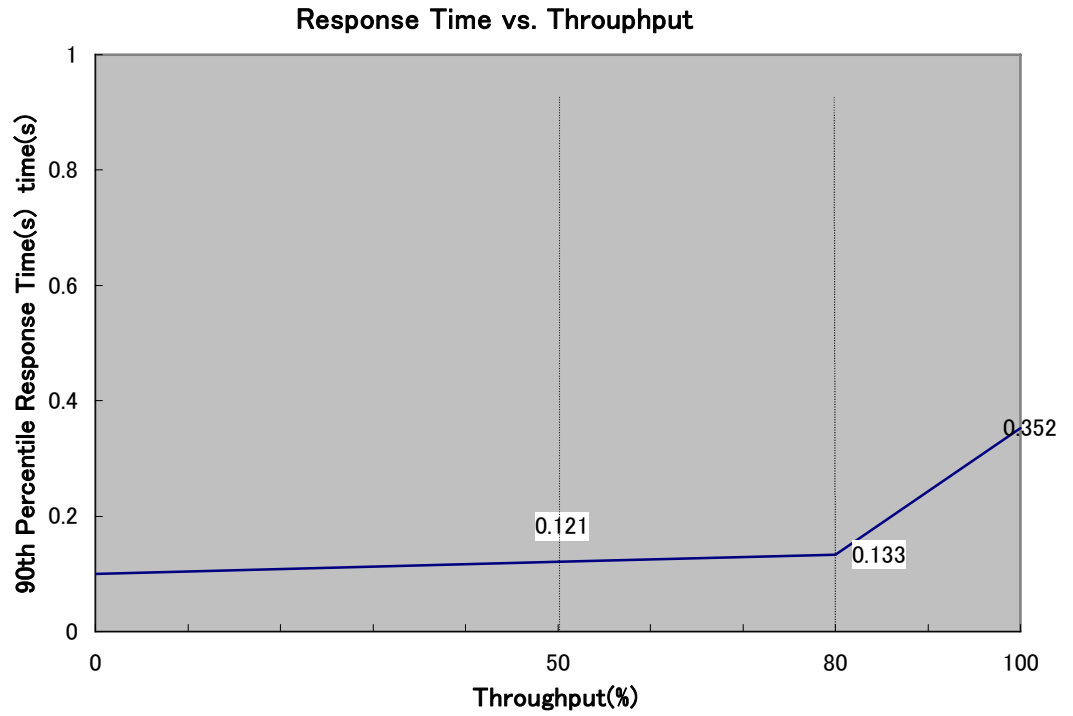
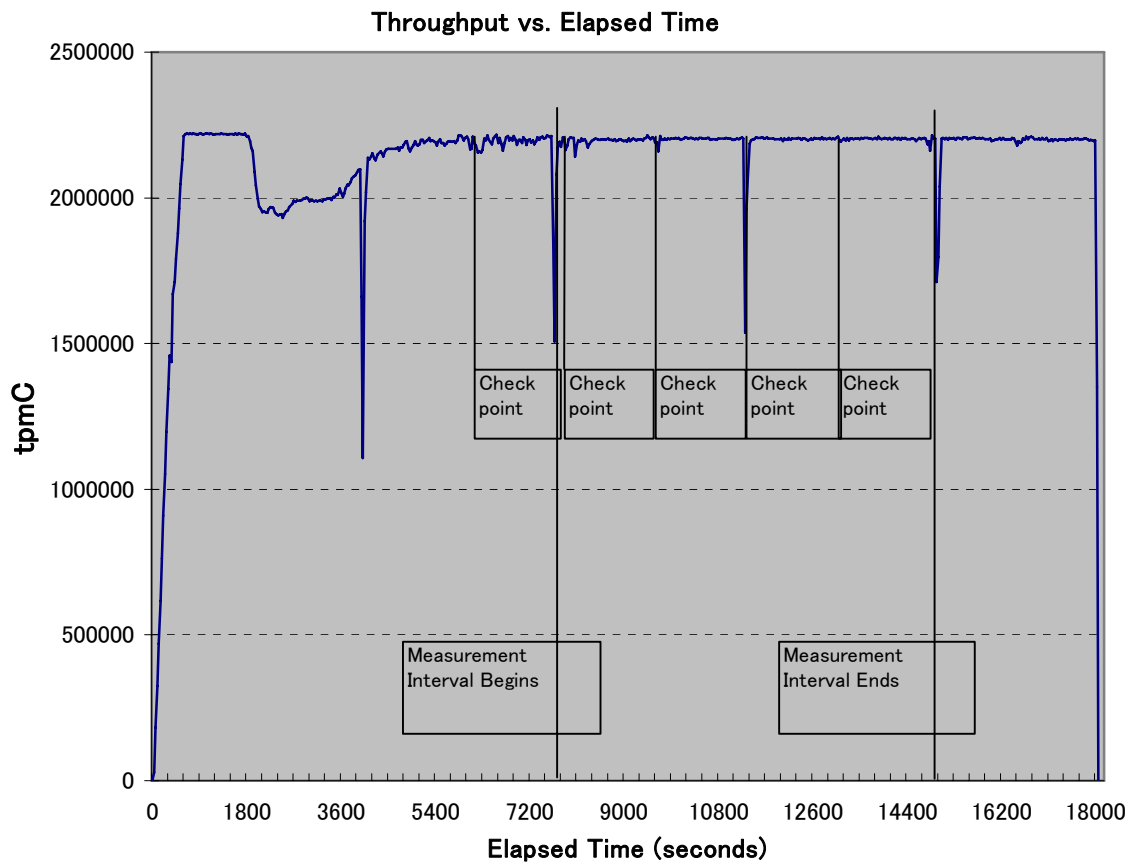


Figure 5.8: Throughput versus Elapsed Time



5.5 Steady State Determination

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 disclosed.

Steady state was determined by examining data reported for each 30-second interval over the duration of the measured run. Steady state was further confirmed by the throughput data collected during the run and graphed in Figure 5.8.

5.6 Work Performed During Steady State

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

The Oracle logical log is on a RAID0+1 array. When one log file becomes full or a time specified by parameters comes, Oracle Database 10g starts a checkpoint process. Oracle automatically logs all checkpoints to an alert file on the server. We configured log files and parameters so that checkpoints would occur in 30 minutes interval. Oracle Database 10g performed 4 times of Log file Switches during MI. At each checkpoint, Oracle wrote to disk all buffer pages that had been updated but not yet physically written to disk.

For the priced system, the logical log space for an 8-hour period is priced.

Serializable Transactions:

Oracle supports serializable transaction isolation in full compliance with the SQL92 and TPC-C requirements. This is implemented by extending multiple concurrency control mechanisms long supported by Oracle.

Oracle queries take no read locks and see only data committed as of the beginning of the query's execution. This means that the readers and writers coexist without blocking one another, providing a high degree of concurrency and consistency. While this mode does prevent reading dirty data, Oracle's default isolation level also permits a transaction that issues a query twice to see non-repeatable reads and phantoms, as defined in SQL92 and TPC-C.

Beginning with Oracle7 release 7.3, a transaction may request a higher degree of isolation with the command `SET TRANSACTION ISOLATION LEVEL SERIALIZABLE` as defined in SQL92. This command will prevent read/write and write/write conflicts that would cause serializability failures.

A session can establish this mode as its default mode, so the `SET TRANSACTION` command need not be issued in each transaction.

Oracle implements `SERIALIZABLE` mode by extending the scope of read consistency from individual query to the entire transaction itself. ALL reads by serializable transactions are therefore repeatable, as the transaction will access prior versions of data changed (or deleted) by other transactions after the start of serializable transactions.

Thus, a serializable transaction sees a fixed snapshot of the database, established at the beginning of the transaction.

To ensure proper isolation, a serializable transaction cannot modify the rows that were changed by other transactions after the beginning of a serializable transaction, or an update (or delete) statement will fail with error `ORA_08177: "cannot serialize access"` and the statement will rollback.

When a serializable transaction fails with this error, the application may either commit the work executed to that point, execute additional statements, or rollback the entire transaction. Repeated attempts to execute the same statement will always fail with the error "can't serialize access" unless the other transaction has rolled back and released its lock. This error and these recovery options are similar to the treatment of deadlocks in systems that use read locks to ensure serializable execution.

In both cases, conflicts between transactions rollback and restarts or commits without re-executing the statement receiving the error.

5.7 Reproducibility

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

No reproducibility run is needed in this revision of the benchmark.

5.8 Measurement Period Duration

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

The reported measured interval was exactly 120 minutes long.

5.9 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.

5.10 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 per New-Order transaction must be disclosed. The percentage of remote Payment transactions must be disclosed. The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed. The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed.

Table 5.4: Transaction Statistics

Statistics		Value
Transaction Mix	New Order	44.94%
	Payment	43.02%
	Order status	4.02%
	Delivery	4.01%
	Stock level	4.01%
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse	85.00%
	Remote warehouse	15.00%
	Accessed by last name	60.00%
Order Status	Accessed by last name	59.99%
Delivery	Skipped transactions	None

5.11 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.

One checkpoint was recorded before the measured window opened and four checkpoints were started inside the measured window.

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)).

	start	end	duration
measurement	17:29:25	19:29:25	120 minutes
	start	End	duration
checkpoint 0	17:02:20	17:32:06	29:46
checkpoint 1	17:31:34	18:01:21	29:47
checkpoint 2	18:00:30	18:30:09	29:39
checkpoint 3	18:29:35	18:59:12	29:37
checkpoint 4	18:58:11	19:27:48	29:37

Clause 6 Related Items

6.1 RTE Descriptions

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

The RTE used is proprietary to Fujitsu. Appendix C contains the profile used as input to this RTE.

6.2 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.

6.3 Emulated Components

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

There were no emulated components in the benchmark configuration other than the emulated users' workstations.

6.4 Functional Diagrams

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

The driver system performed the data generation and input functions of the display device. It also captured the input and output data and timestamps for post-processing of the reported metrics. No other functionality was included on the driver system

The abstract at the beginning of this report contains detailed diagrams of both the benchmark configuration and the priced configuration, including the driver system.

6.5 Networks

The network configuration of both the tested services and 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).

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

Five 1Gbps ethernet LAN connections were used between the server and five switches, to which clients were connected by 96(ninety-six) 100Mbps ethernet LAN connections. Another 96(ninety-six) 100Mbps ethernet LAN connections were used between the clients and the switches connected to the emulated users.

6.6 Operator Intervention

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

This configuration does not require any operator intervention to sustain eight hours of the reported throughput, other than beginning the checkpointing process.

Clause 7 Related Items

7.1 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 data. 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 basis of all discounts used must be disclosed.

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

7.2 Availability

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

The total solution as priced will be available by April 30, 2008.

7.3 Throughput, and Price Performance

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

Maximum Qualified Throughput :	2,196,268 tpmC
Price per tpmC :	\$4.70 per tpmC
Three-year cost of ownership :	\$10,306,995 USD

7.4 Country Specific Pricing

Additional Clause 7 related items may be included in the Full Disclosure Report for each country specific priced configuration. Country specific pricing is subject to Clause 7.1.7

This system is being priced for the United States of America.

7.5 Usage Pricing

For any usage pricing, the sponsor must disclose:

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

The component pricing based on usage is shown below:

- 32 Oracle Database 10g Enterprise Edition, Per Processor, Unlimited Users, 3 years
- 1 Red Hat Enterprise Linux AS (Standard)
- 96 Red Hat Enterprise Linux ES (Standard)
- 96 BEA Tuxedo Core Functionality Services(CFS-R)

7.6 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 H at the end of this document.

Clause 9 Related Items

9.1 Auditor's Report

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

This implementation of the TPC-C benchmark was audited by Francois Raab of InfoSizing, Inc. The auditor's attestation letter is provided in this section provided in Appendix I.

9.2 Availability of the Full Disclosure Report

The Full Disclosure Report must be readily available to the public at a reasonable charge, similar to the charges for similar documents by the test sponsor. The report must be made available when results are made public. In order to use the phrase "TPC Benchmark™ C", the Full Disclosure Report must have been submitted to the TPC Administrator as well as written permission obtained to distribute same.

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

Transaction Processing Performance Council
Presidio of San Francisco
Building 572B Ruger St. (surface)
P.O.Box 29920 (mail)
San Francisco, CA 94129-0920
Voice: 415-561-6272
Fax: 415-561-6120
Email: info@tpc.org

Appendix A: Client Source Code

```

.....
common/GetPrivateProfileString.c
.....

/*****
*****
*
*          *
*   TPC-C Client Application Program Source
*
*          *
*   Entry Functions
*   (1) GetPrivateProfileString
*
*          *
*   CREATE by TSL 2003.12.18
*
*          *
*   All Right Reserved, Copyright Co. FUJITSU
*   LIMITED 2003-2004 *
*****
*****/

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

/*****
*****
*   Get data string corresponded key in
*   cogfiguration file.
*   *
*   Return Value
*   *
*   Get string length
*   *
*****
*****/

int GetPrivateProfileString(char*
section_name, /* Section name
*/
name          char* key_name, /* Key
name          char* default_str, /* Default
string, if kye nothing */
data          char* key_data, /* Key
size of key data int buf_size, /* Buffer
name          char* file_name){ /* File

FILE* prof_file;
char read_buf[256];
char search[32];
char* get_str;
char* key_pos=0;
int get_cnt;
int i;

/* Open profile file */
if ((prof_file = fopen(file_name, "r")) == NULL)
{
goto DEFAULT_STRING;
}

/* Make searching section name "[section
name]" */

```

```

search[0] = '[';
strcpy(&search[1], section_name);
strcat(search, "]");

/* Search section name */
while((get_str = fgets(read_buf,
sizeof(read_buf), prof_file)) != NULL) {

/* Search section name form to be read one
line */
if ((char*)strstr(read_buf, search) == NULL)
{
/* No match section name, next line read
*/
continue;
}
break;
}
if (get_str == NULL) {
/* Found EOF or read error */
goto DEFAULT_STRING_FCLOSE;
}

/* Make searching key name "key_name=" */
strcpy(search, key_name);
strcat(search, "=");

/* Search key name in this section */
while((get_str = fgets(read_buf,
sizeof(read_buf), prof_file)) != NULL) {
for (i = 0; read_buf[i] == ' ' || read_buf[i] ==
'\t'; i++);
if (read_buf[i] == '[') {
/* Other section started, undefined key
name */
goto DEFAULT_STRING_FCLOSE;
}
if ((key_pos = (char*)strstr(read_buf,
search)) == NULL) {
/* No match key name */
continue;
}
break;
}
if (get_str == NULL) {
/* Found EOF or read error */
goto DEFAULT_STRING_FCLOSE;
}

fclose(prof_file);

/* Get key_value, fixed format "key value" */
for (; *key_pos != ""; key_pos++);
key_pos++;
for (get_cnt = 0; *key_pos != ""; key_pos++) {
/* Get & set key value */
*key_data = *key_pos;
key_data++;
get_cnt++;
if (get_cnt >= (buf_size - 1)) {
/* Key data buffer full */
break;
}
}
*key_data = '\0';
return(get_cnt);

DEFAULT_STRING_FCLOSE:
fclose(prof_file);

DEFAULT_STRING:
strncpy(key_data, default_str, buf_size-1);
return(strlen(key_data));
}

```

```

.....
common/MakeShell
.....

#!/bin/sh
cd /home/tpc/client_apl/common
make > make_result.txt 2>&1

.....
common/Makefile
.....

#-----
-----
# Makefile : Makefile for common of TPAPL and
SVRAPL.
#
# Created by TSL 2003.12.17
#
# All Right Rserverd, Copyright Co, FUJITSU
LIMITED 2003-2004.
#-----
-----

# GCC compile configurations
AR = ar
ARFLAGS = rv
CFLAGS = -Wall -O2
CC = gcc

# MACRO definition
DMACRO = -DTSL -DPLSQLFLAG=1 -DTUX

# home directory
ORADIR = /usr/local/oracle
TUXDIR = /usr/local/BEA/tuxedo8.1
SVRDIR = /home/tpc/client_apl/svrapl

# include directory
ORA_INC = -I$(ORADIR)/rdbms/demo -
I$(ORADIR)/rdbms/public
COM_INC = -I$(SVRDIR)/common
SVR_INC = -I$(SVRDIR)
TUX_INC = -I$(TUXDIR)/include
INCLUDE = $(COM_INC) $(SVR_INC)
$(ORA_INC) $(TUX_INC)

# target object
COMOBS = log.o sema.o
GetPrivateProfileString.o shmем.o
COMLIB = libcom.a

INCFILES = log.h sema.h forlinux.h shmем.h

$(COMLIB) : $(COMOBS)
$(AR) $(ARFLAGS) $(COMLIB) $(COMOBS)

.SUFFIXES : .o .c
.c.o:
$(CC) -o $@ -c $(CFLAGS) $(INCLUDE)
$(DMACRO) $<

$(COMOBS) : $(INCFILES)

clean:
rm $(COMLIB) $(COMOBS)

```

```

.....
common/forlinux.h
.....

/*****
*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* definition for converting Linux.
*
*
* CREATE by TSL 2003.05.16
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *
*****
*****/
/* forlinux.h */

#include <limits.h>
#define MAX_PATH PATH_MAX /*
Windows:MAX_PATH , Linux:PATH_MAX */
#define Sleep(x) poll(0, 0, x); /* sleep unit is
a msec. */

.....
common/log.c
.....

/*****
*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Log is outputted to a file.
*
*
* CREATE by TSL 2002.11.29
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002-2004 *
*****
*****/
#include "forlinux.h"
#include <stdio.h>
#include <string.h>
#include <time.h>
#include <sys/types.h>
#include <stdarg.h>
#include <unistd.h>
#include <pthread.h>
#include <sys/types.h>
#include <sys/stat.h>
#include "sema.h"

#define LOG_MODULE
#include "log.h"

void TpcUserLog(char* file_name, int line_no,
char* type_name, char* ftmp, ...)
{
FILE* fp;
pid_t pid;
pthread_t tld;

```

```

char* fname;
int stat;

/* -- BEGIN -- Modified by Hayashi for thread-
safe. 2006/02/13 */
#if 0
! struct tm *nowtime;
#else
struct tm tt;
struct tm *nowtime=&tt;
#endif
/* -- END -- Modified by Hayashi for thread-safe.
2006/02/13 */

time_t long_time;
va_list va;

if (strcmp(type_name, "LCK") != 0) {
/* Lock semaphore */
stat = LockSem(GLB_LogSemId);
}
/* Get current time. */

time( &long_time );

/* -- BEGIN -- Modified by Hayashi for thread-
safe. 2006/02/13 */
#if 0
! nowtime = localtime( &long_time );
#else
localtime_r( &long_time, nowtime );
#endif
/* -- END -- Modified by Hayashi for thread-safe.
2006/02/13 */

/* Get process Id. */
pid = getpid();

/* Get thread Id. */
tld = pthread_self();

/* Get just file name from a path. */
fname = (char*)strchr(file_name, (int)'/');
if (fname == NULL) {
fname = file_name;
} else {
fname = fname + 1;
}

va_start(va, ftmp);

fp = fopen(GLB_LogFilePath, "a");
fprintf(fp, "%02d:%02d:%02d [%6d:%08x] %-
32s(%4d) :%s: ",
nowtime->tm_hour, nowtime->tm_min,
nowtime->tm_sec, pid, (int)tld, fname, line_no,
type_name);
vfprintf(fp, ftmp, va);

if (*(ftmp + strlen(ftmp) - 1) != '\n')
fprintf(fp, "\n");

va_end(va);

fclose(fp);

/* change mode which all users can read and
write. */
chmod(GLB_LogFilePath, S_IRUSR
|S_IWUSR|S_IRGRP|S_IWGRP|S_IROTH|
S_IWOTH);

if (strcmp(type_name, "LCK") != 0) {
// Unlock semaphore
stat = UnlockSem(GLB_LogSemId);

```

```

}

return;
}

.....
common/log.h
.....

/*****
*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Log is outputted to a file.
*
*
* CREATE by TSL 2002.11.29
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *
*****
*****/

void TpcUserLog(char *file_name, int line_no,
char* type_name, char* ftmp, ...);

extern char GLB_LogFilePath[MAX_PATH];
extern int GLB_LogSemId;

#define DEFAULT_SVRAPL_LOG_PATH
"/home/tpc/log/DBDepend_Userlog.log"
#define DEFAULT_TPAPL_LOG_PATH
"/home/tpc/log/userlog.log"

#define LOG_ERR __FILE__, __LINE__, "ERR"
#define LOG_INF __FILE__, __LINE__, "INF"
#define LOG_WRN __FILE__, __LINE__,
"WRN"
#define LOG_LCK __FILE__, __LINE__, "LCK"

#define LOG_FILE_INF __FILE__,
__LINE__, "INF"
#define LOG_FILE_LINE __FILE__,
__LINE__

.....
common/sema.c
.....

/*****
*****
*
* TPC-C Client Application Program Source
*
*
* Filename :
* sema.c
*
* Entry Functions :
* There are functions to control semaphore.
*
*
* CREATE by TSL 2003.12.18
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *

```

```

*****
*****/

#include "forlinux.h"
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#include <errno.h>
#include "log.h"
#include "sema.h"

/*****
*****/
* Initialize semaphore.
* Return Value
* > 0 semaphore Id. (always over 0)
*
* < 0 fail.
*

*****
*****/
int InitSem(char *path, int projectId)
{
    int sid;
    union semun{
        int val;
        struct semid_ds *buf;
        ushort *array;
    } c_arg;

    TpcUserLog(LOG_LCK, "InitSem: start
path<%s> projectId=%d\n",
    path, projectId);

    if ((sid = GetSem(path, projectId)) == -1) {
        TpcUserLog(LOG_LCK, "GetSem() fail,
path<%s> projectId=%d\n",
        path, projectId);
        return(-1);
    }
    c_arg.val=1;
    if (semctl(sid,0,SETVAL,c_arg)==-1) {
        TpcUserLog(LOG_LCK, "semctl fail,
sid=%d\n",sid);
        return(-1);
    }
    TpcUserLog(LOG_LCK, "InitSem: Get
semid =%d\n",sid);

    return(sid);
}
/*****
*****/
* Get semaphore.
* Return Value
* > 0 semaphore Id. (always over 0)
*
* < 0 fail.
*

*****
*****/
int GetSem(char *path, int projectId)
{
    int sid;
    int key;

    if ((key = ftok(path,projectId)) == -1) {
        TpcUserLog(LOG_LCK, "ftok() fail,
path<%s> projectId=%d errno=%d\n",
        path, projectId, errno);
        return(-1);
    }

```

```

    if ((sid=semget(key,1,0666|IPC_CREAT))== -
1){
        TpcUserLog(LOG_LCK, "semget() fail,
key=%d errno=%d\n",key, errno);
        return(-1);
    }

    return(sid);
}
/*****
*****/
* Reuire to lock semaphore.
*
* Return Value
* 1 success.
* -1 fail.
*

*****
*****/
int LockSem(int sid)
{
    struct sembuf sb;

    sb.sem_num=0;
    sb.sem_op=-1;
    sb.sem_flg=0;
    if(semop(sid,&sb,1)== -1) {
        TpcUserLog(LOG_LCK, "semop() fail,
sid=%d\n",sid);
        return(-1);
    }
    return(1);
}
/*****
*****/
* Reuire to unlock semaphore.
*
* Return Value
* 1 success.
* -1 fail.
*

*****
*****/
int UnlockSem(int sid)
{
    struct sembuf sb;

    sb.sem_num=0;
    sb.sem_op=1;
    sb.sem_flg=0;
    if(semop(sid,&sb,1)== -1){
        TpcUserLog(LOG_LCK, "semop() fail,
sid=%d\n",sid);
        return(-1);
    }
    return(1);
}

.....
common/sema.h
.....

/*****
*****/
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Semaphore control.
*
*
* CREATE by TSL 2003.12.19
*

```

```

*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *

*****/

/*****
*****/
/*== project Id =====*/
#define SEM_SVRAPL_PROJID
(int)'S'
#define SEM_TPAPL_PROJID (int)'T'
#define SEM_SAMPLING_PERFORMANCE
(int)'P'

/*****
*****/
/*=====*/
/*==*/
/* prototype definition */
/*=====*/
/*=====*/
int InitSem(char *path, int projectId);
int GetSem(char *path, int projectId);
int LockSem(int sid);
int UnlockSem(int sid);

.....
common/Shmem.c
.....

/*****
*****/
*
* TPC-C Client Application Program Source
*
*
* Filename :
* sema.c
*
* Entry Functions :
* There are functions to control shared
memory.
*
* CREATE by TSL 2004.01.15
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *

*****/

#include "forlinux.h"
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <errno.h>
#include "log.h"

/*****
*****/
* Initialize shared memory.
* Return Value
* > 0 shared memory address. (always over
0)
* < 0 fail.
*

*****
*****/
char* InitShmem(char *path, int projectId, int
size)
{
    int shmId;
    int key;

```

```

char *shmaddr;

TpccUserLog(LOG_LCK, "InitShmem: start
path<%s> projectId=%d\n",
    path, projectId);

if ((key = ftok(path,projectId)) == -1) {
    TpccUserLog(LOG_LCK, "ftok() fail,
path<%s> projectId=%d errno=%d\n",
    path, projectId, errno);
    return((char *)-1);
}
if
((shmidx=shmget(key,size,IPC_CREAT|0666))==
-1){
    TpccUserLog(LOG_LCK, "shmget() fail,
key=%d errno=%d",key, errno);
    return((char *)-1);
}
if( (shmaddr = (char *)shmat(shmid, NULL, 0))
== (char *)-1 ) {
    TpccUserLog(LOG_LCK, "shmat() fail,
shmid=%d path<%s> projectId=%d errno=%d\n",
    shmid, path, projectId, errno);
    return ((char *)-1);
}

TpccUserLog(LOG_LCK, "InitShmem: Get
shmid =%d shmaddr = %08x\n",shmid,
shmaddr);

return(shmaddr);
}
/*****
* Get shared memory.
* Return Value
* > 0 shared memory address. (always over
0)
* < 0 fail.
*****/
char* GetShmem(char *path, int projectId, int
size)
{
    int shmid;
    int key;
    char *shmaddr;

    if ((key = ftok(path,projectId)) == -1) {
        TpccUserLog(LOG_LCK, "ftok() fail,
path<%s> projectId=%d errno=%d\n",
        path, projectId,errno);
        return((char *)-1);
    }
    if ((shmidx=shmget(key,size, 0))== -1){
        TpccUserLog(LOG_LCK, "shmget() fail,
key=%d errno=%d\n",key,errno);
        return((char *)-1);
    }
    if ((shmaddr = (char *)shmat(shmid, NULL, 0))
== (char *)-1 ) {
        TpccUserLog(LOG_LCK, "shmat() fail,
shmid=%d path<%s> projectId=%d errno=%d\n",
        shmid, path, projectId, errno);
        return ((char *)-1);
    }

    return(shmaddr);
}
.....
common/shmem.h

```

```

.....
/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* Shared memory control.
*
* CREATE by TSL 2004.01.15
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *
*****/

/!== project Id =====*/
#define
SHMEM_SAMPLING_PERFORMANCE
(int)P'

/!=====
====*/
/! prototype definition */
/!=====
====*/
char* InitShmem(char *path, int projectId, int
size);
char* GetShmem(char *path, int projectId, int
size);

.....
tpapl/ClientMonitor.c
.....
/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* (1) ClientMonitor
* (2) ClientLogCheck
* (3) CleantShutdown
* (4) ClientInfSample
* (5) ClientSampleInit
* (6) ClientSampleSelfCsv
*
* CREATE by TSL 2004.01.18
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2004 *
*****/
#include "forlinux.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <unistd.h>
#include "sema.h"
#include "shmem.h"
#include "SampleInfo.h"
#include "log.h"

```

```

/* Global area */
extern char GLB_TpAplLogPath[];
extern char GLB_SvrAplLogPath[];
MAC_SampleGlobalArea;

/*****
* Client monitoring function.
* Return Value
* 0 : Normal end
* !0: Illegal function no.
* Return Information
* HTML document
*****/

int ClientMonitor(int func_no, char* html_buf) {

/* Dispatch function by function no. */
switch(func_no) {

/* Client startup function */
case -1:
    ClientLogCheck(html_buf);
    break;

/* Client shutdown */
case -2:
    ClientSetSample(html_buf);
    break;

/* Client monitor */
case -3:
    ClientInfSample(html_buf);
    break;

default:
/* Error return */
return -1;
break;
}

return 0;
}

/*****
* Check client's log files.
* Check files are ...
* use:log.log : TpApl log
* DBDepend_Userlog.log : SvrApl log
*
* Return Value
* NONE
* Return Information
* HTML document
*****/

void ClientLogCheck(char* html_buf) {

int CheckLogFile(char* file_path, char*
key_word);

#define NO_ERROR_LOG "No error found."
#define CLIENT_LOG_CHECK "\
<HTML><HEAD><TITLE>Client Log
Check</TITLE></HEAD><BODY>\r\n\
<P> \
The %s check log files.\r\n\
Result : %s \r\n\
</P></BODY></HTML>\r\n"

```

```

char host_name[32];

/* Get host name */
host_name[0] = '\0';
gethostname(host_name, sizeof(host_name));

/* Check TpApl log file */
if (CheckLogFile(GLB_TpAplLogPath,
":ERR:") == 0) {
    /* No error */
    sprintf(html_buf, CLIENT_LOG_CHECK,
host_name, NO_ERROR_LOG);
} else {
    /* Error found */
    sprintf(html_buf, CLIENT_LOG_CHECK,
host_name, "Error in userlog.log");
    return;
}

/* Check SvrApl log file */
if (CheckLogFile(GLB_SvrAplLogPath,
":ERR:") == 0) {
    /* No error */
    sprintf(html_buf, CLIENT_LOG_CHECK,
host_name, NO_ERROR_LOG);
} else {
    /* Error found */
    sprintf(html_buf, CLIENT_LOG_CHECK,
host_name, "Error in DBDepend_Userlog.log");
}

/*****
* Check log files has error key word.
*
* Return Value
* >0 : found number of keywords
*
* -1 : file open error (maybe no exist
*
*****/
int CheckLogFile(char* file_path, char*
key_word) {

    FILE* log_file;
    char rd_buff[256];
    int find_words = 0;

    if ((log_file = fopen(file_path, "r")) == NULL) {
        /* Open error */
        return -1;
    }

    while (fgets(rd_buff, sizeof(rd_buff),
log_file) != NULL) {

        if (strstr(rd_buff, key_word) != NULL) {
            find_words++;
        }
    }
    fclose(log_file);
    return find_words;
}

/*****
* Set sampling disable.
* Return Value
* NONE
* Return Information

```

```

* HTML document
*
*****/
void ClientSetSample(char* html_buf) {

#define CLIENT_DIRECT "\
<HTML><HEAD><TITLE>Client sampling
disable</TITLE></HEAD><BODY>\
<P>\
The %s set sampling disable.\r\n\
Result : No error found.\r\n\
</P></BODY></HTML>\r\n"

    char host_name[32];

    GLBSMP_shared_mem->DataSampling =
DATASAMPLE_DISABLE;

    host_name[0] = '\0';
    gethostname(host_name, sizeof(host_name));
    sprintf(html_buf, CLIENT_DIRECT,
host_name);
}

/*****
* Client performance information Sampling
*
* Return Value
* NONE
* Return Information
* HTML document
*
*****/
void ClientInfSample(char* html_buf) {

#define CLIENT_SAMPLE "\
<HTML><HEAD><TITLE>Client Sampling
information</TITLE></HEAD><BODY>\
<PRE>\
Information of %s\r\n\
\r\n\
TpApl performance \r\n\
Sto\r\n\
    Num of waiting process %-7d %-7d %-
7d %-7d\r\n\
    Answer to RTE (ms) %-7d %-7d %-
7d %-7d\r\n\
\r\n\
SvrApl performance \r\n\
    SMAN MAX AVR
TRX\r\n\
    New Order Response %-7d %-7d %-
7d %-7d\r\n\
    Payment Response %-7d %-7d %-
7d %-7d\r\n\
    Order Status Response %-7d %-7d %-
7d %-7d\r\n\
    Derivery Response %-7d %-7d %-7d %-
7d\r\n\
    Stock Level Response %-7d %-7d %-
7d %-7d\r\n\
</PRE></BODY></HTML>\r\n"

#define EXT_FUNC3_ERROR "\
<HTML><HEAD><TITLE>Error
information</TITLE></HEAD><BODY>\
<PRE>\
Failure create SvrAPL object(Extended function
= -3)\
</PRE></BODY></HTML>\r\n"

#define EXT_FUNC3_ERROR "\
<HTML><HEAD><TITLE>Error
information</TITLE></HEAD><BODY>\
<PRE>\
Failure create SvrAPL object(Extended function
= -3)\
</PRE></BODY></HTML>\r\n"

```

```

#if 0
#define CLIENT_SAMPLE "\
<HTML><HEAD><TITLE>Client Sampling
information</TITLE></HEAD><BODY>\
<PRE>\
Information of CL001\r\n\
\r\n\
TpApl performance \r\n\
    New Pay Odr Del
Sto\r\n\
    Num of waiting process 10 20 30
40 50\r\n\
    Answer to RTE (ms) 110 220 330
440 550\r\n\
\r\n\
SvrApl performance \r\n\
    SMAN MAX AVR
TRX\r\n\
    New Order Response 10 11 12
13\r\n\
    Payment Response 110 111 112
113\r\n\
    Order Status Response 210 211 212
213\r\n\
    Derivery Response 310 311 312
313\r\n\
    Stock Level Response 410 411 412
413\r\n\
</PRE></BODY></HTML>\r\n"
#endif

char host_name[32];
unsigned int ans_new_avr, ans_pay_avr,
ans_odr_avr, ans_del_avr, ans_sto_avr;
unsigned int rsp_new_avr, rsp_pay_avr,
rsp_odr_avr, rsp_del_avr, rsp_sto_avr;

    SAMPLING_DATA sampling_data;

/* Get host name, inserting to HTML */
host_name[0] = '\0';
gethostname(host_name, sizeof(host_name));

/* copy sampling information into own area */
LockSem(GLBSMP_semid);
memcpy((void*)&sampling_data,
(void*)GLBSMP_shared_mem,
(size_t)sizeof(SAMPLING_DATA));

/* Clear sampling information for next
sampling interval */
memset((void*)GLBSMP_shared_mem, 0x00,
(unsigned int)&(SAMPLING_DATA*0)-
>MaxRspTimeNewOrder);

    UnlockSem(GLBSMP_semid);

/* Compute average data */
ans_new_avr =
sampling_data.NumReqNewOrder != 0?
sampling_data.AnsNewOrder /
sampling_data.NumReqNewOrder : 0;
ans_pay_avr =
sampling_data.NumReqPayment != 0?
sampling_data.AnsPayment /
sampling_data.NumReqPayment : 0;
ans_odr_avr =
sampling_data.NumReqOrderStatus != 0?
sampling_data.AnsOrderStatus /
sampling_data.NumReqOrderStatus : 0;
ans_del_avr =
sampling_data.NumReqDelivery != 0?
sampling_data.AnsDelivery /
sampling_data.NumReqDelivery : 0;

```

```

ans_sto_avr =
sampling_data.NumReqStockLevel != 0?
    sampling_data.AnsStockLevel /
sampling_data.NumReqStockLevel : 0;

rsp_new_avr =
sampling_data.NumNewOrder != 0?
    sampling_data.RspTimeNewOrder /
sampling_data.NumNewOrder : 0;
rsp_pay_avr =
sampling_data.NumPayment != 0?
    sampling_data.RspTimePayment /
sampling_data.NumPayment : 0;
rsp_odr_avr =
sampling_data.NumOrderStatus != 0?

sampling_data.RspTimeOrderStatus /
sampling_data.NumOrderStatus : 0;
rsp_del_avr = sampling_data.NumDelivery !=
0?
    sampling_data.RspTimeDelivery /
sampling_data.NumDelivery : 0;
rsp_sto_avr =
sampling_data.NumStockLevel != 0?
    sampling_data.RspTimeStockLevel
/ sampling_data.NumStockLevel : 0;

sprintf(html_buf, CLIENT_SAMPLE ,
    host_name,
    sampling_data.NumQueNewOrder,
    sampling_data.NumQuePayment,
    sampling_data.NumQueOrderStatus,
    sampling_data.NumQueDelivery,
    sampling_data.NumQueStockLevel,
    ans_new_avr, ans_pay_avr, ans_odr_avr,
    ans_del_avr, ans_sto_avr,

    sampling_data.SMaxRspTimeNewOrder,
    sampling_data.MaxRspTimeNewOrder,
    rsp_new_avr,
    sampling_data.NumNewOrder,
    sampling_data.SMaxRspTimePayment,
    sampling_data.MaxRspTimePayment,
    rsp_pay_avr,
    sampling_data.NumPayment,
    sampling_data.SMaxRspTimeOrderStatus,
    sampling_data.MaxRspTimeOrderStatus,
    rsp_odr_avr,
    sampling_data.NumOrderStatus,
    sampling_data.SMaxRspTimeDelivery,
    sampling_data.MaxRspTimeDelivery,
    rsp_del_avr,
    sampling_data.NumDelivery,
    sampling_data.SMaxRspTimeStockLevel,
    sampling_data.MaxRspTimeStockLevel,
    rsp_sto_avr,
    sampling_data.NumStockLevel);
}

/*****
* Initialize sampling
* Return Value
* NONE
*****/

void ClientSampleInit() {
#define SAMPLING_CONF_FILE
"/home/tpc/conf/sampling.conf"
#define DEFAULT_CSV_FILE
"/home/tpc/log/sampling.csv"
#define DEFAULT_SAMPLING_INTERVAL 5

FILE* conf_file;

```

```

char rd_buff[MAX_PATH];
int i;

/* Initialize shared memory */
MAC_SampleInitParent;

/* Setup sampling configuration */
if ((conf_file = fopen(SAMPLING_CONF_FILE,
"r")) == NULL) {
    GLBSMP_shared_mem-
>SelfSamplingOutput =
SELFOUTPUT_DISABLE;
    return;
}
GLBSMP_shared_mem->SelfSamplingOutput
= SELFOUTPUT_ENABLE;

/* CSV file path */
if (fgets(rd_buff, sizeof(rd_buff), conf_file) ==
NULL) {
    strcpy(GLBSMP_shared_mem-
>CsvFilePath, DEFAULT_CSV_FILE);
    GLBSMP_shared_mem->SamplingInterval
= DEFAULT_SAMPLING_INTERVAL;
    goto FILE_CLOSE;
}
for(i = 0; !(rd_buff[i] == '\n' || rd_buff[i] == '\0') ;
i++);
rd_buff[i] = '\0';
strcpy(GLBSMP_shared_mem->CsvFilePath,
rd_buff);

/* Sampling interval */
if (fgets(rd_buff, sizeof(rd_buff), conf_file) ==
NULL) {
    GLBSMP_shared_mem->SamplingInterval
= DEFAULT_SAMPLING_INTERVAL;
    goto FILE_CLOSE;
}
GLBSMP_shared_mem->SamplingInterval =
atoi(rd_buff);

FILE_CLOSE:
fclose(conf_file);
}

/*****
****
* Self CSV data output
* Return Value
* NONE
*****/

void ClientSampleSelfCsv(time_t cur_sec) {

    FILE* csv_file;

#define TITLE_LINE
"time,num_thread,stay_New,stay_Pay,stay_Odr,
stay_Del,stay_Sto,\n"

"resp_New,num_New,resp_Pay,num_Pay,resp_
Odr,num_Odr,resp_Del,num_Del,resp_Sto,num
_Sto,\n"

"imax_New,imax_Pay,imax_Odr,imax_Del,imax
_Sto,\n"

"max_New,max_Pay,max_Odr,max_Del,max_St
o,\n"

"ans_New,nas_Pay,ans_Odr,ans_Del,ans_Sto,c
onnect\n"

```

```

/* -- BEIGN -- Modified by Hayashi for thread-
safe. 2006/02/13 */
#if 0
! struct tm *nowtime;
#else
struct tm tt;
struct tm *nowtime= &tt;
#endif
/* -- END -- Modified by Hayashi for thread-safe.
2006/02/13 */

unsigned int ans_new_avr, ans_pay_avr,
ans_odr_avr, ans_del_avr, ans_sto_avr;
unsigned int rsp_new_avr, rsp_pay_avr,
rsp_odr_avr, rsp_del_avr, rsp_sto_avr;

SAMPLING_DATA sampling_data;

if (GLBSMP_shared_mem-
>SelfSamplingOutput ==
SELFOUTPUT_DISABLE) {
    /* Output disable */
    return;
}

LockSem(GLBSMP_semaphore);
if ((cur_sec - GLBSMP_shared_mem-
>CsvOutTime) < GLBSMP_shared_mem-
>SamplingInterval) {
    /* No output timing */
    goto UNLOCK_SEM;
}

/* Output CSV data */
if ((csv_file = fopen(GLBSMP_shared_mem-
>CsvFilePath, "a")) == NULL) {
    goto UNLOCK_SEM;
}

if (GLBSMP_shared_mem->CsvOutTime ==
0) {
    /* First time, output header data */
    fprintf(csv_file, TITLE_LINE);
    fclose(csv_file);
    GLBSMP_shared_mem->CsvOutTime =
cur_sec;
    goto UNLOCK_SEM;
}
GLBSMP_shared_mem->CsvOutTime =
cur_sec;

/* copy sampling information into own area */
memcpy((void*)&sampling_data,
(void*)GLBSMP_shared_mem,
(size_t)sizeof(SAMPLING_DATA));

/* Clear sampling information for next
sampling interval */
memset((void*)GLBSMP_shared_mem, 0x00,
(unsigned int)&((SAMPLING_DATA*)0)-
>MaxRspTimeNewOrder);

/* Compute average data */
ans_new_avr =
sampling_data.NumReqNewOrder != 0?
    sampling_data.AnsNewOrder /
sampling_data.NumReqNewOrder : 0;
ans_pay_avr =
sampling_data.NumReqPayment != 0?

```

```

        sampling_data.AnsPayment /
sampling_data.NumReqPayment : 0;
    ans_odr_avr =
sampling_data.NumReqOrderStatus != 0?
        sampling_data.AnsOrderStatus /
sampling_data.NumReqOrderStatus : 0;
    ans_del_avr =
sampling_data.NumReqDelivery != 0?
        sampling_data.AnsDelivery /
sampling_data.NumReqDelivery : 0;
    ans_sto_avr =
sampling_data.NumReqStockLevel != 0?
        sampling_data.AnsStockLevel /
sampling_data.NumReqStockLevel : 0;

    rsp_new_avr =
sampling_data.NumNewOrder != 0?
        sampling_data.RspTimeNewOrder /
sampling_data.NumNewOrder : 0;
    rsp_pay_avr =
sampling_data.NumPayment != 0?
        sampling_data.RspTimePayment /
sampling_data.NumPayment : 0;
    rsp_odr_avr =
sampling_data.NumOrderStatus != 0?
        sampling_data.RspTimeOrderStatus /
sampling_data.NumOrderStatus : 0;
    rsp_del_avr = sampling_data.NumDelivery !=
0?
        sampling_data.RspTimeDelivery /
sampling_data.NumDelivery : 0;
    rsp_sto_avr =
sampling_data.NumStockLevel != 0?
        sampling_data.RspTimeStockLevel /
sampling_data.NumStockLevel : 0;

    /* Output sampling data */
/* -- BEIGN -- Modified by Hayashi for thread-
safe. 2006/02/13 */
#if 0
! nowtime = localtime( &cur_sec );
#else
    localtime_r( &cur_sec, nowtime );
#endif
/* -- END -- Modified by Hayashi for thread-safe.
2006/02/13 */

    fprintf(csv_file,
        "%02d-%02d %02d:%02d:%02d,",
            nowtime->tm_mon+1, nowtime->tm_mday,
            nowtime->tm_hour, nowtime->tm_min, nowtime-
>tm_sec);

    /* Number of thread (no sampling information)
*/
    fprintf(csv_file, "%d,", 0);

    /* Waiting process queue */
    fprintf(csv_file, "%d,",
sampling_data.NumQueueNewOrder);
    fprintf(csv_file, "%d,",
sampling_data.NumQueuePayment);
    fprintf(csv_file, "%d,",
sampling_data.NumQueueOrderStatus);
    fprintf(csv_file, "%d,",
sampling_data.NumQueueDelivery);
    fprintf(csv_file, "%d,",
sampling_data.NumQueueStockLevel);

    /* Responce time & number of proesing
trasection */
    fprintf(csv_file, "%.3f,", (float)rsp_new_avr /
1000.0);

```

```

        fprintf(csv_file, "%d,",
sampling_data.NumNewOrder);
    fprintf(csv_file, "%.3f,", (float)rsp_pay_avr /
1000.0);
    fprintf(csv_file, "%d,",
sampling_data.NumPayment);
    fprintf(csv_file, "%.3f,", (float)rsp_odr_avr /
1000.0);
    fprintf(csv_file, "%d,",
sampling_data.NumOrderStatus);
    fprintf(csv_file, "%.3f,", (float)rsp_del_avr /
1000.0);
    fprintf(csv_file, "%d,",
sampling_data.NumDelivery);
    fprintf(csv_file, "%.3f,", (float)rsp_sto_avr /
1000.0);
    fprintf(csv_file, "%d,",
sampling_data.NumStockLevel);

    /* Max processing time in sampling interval */
    fprintf(csv_file, "%.3f,",
(float)sampling_data.SMaxRspTimeNewOrder /
1000.0);
    fprintf(csv_file, "%.3f,",
(float)sampling_data.SMaxRspTimePayment /
1000.0);
    fprintf(csv_file, "%.3f,",
(float)sampling_data.SMaxRspTimeOrderStatus /
1000.0);
    fprintf(csv_file, "%.3f,",
(float)sampling_data.SMaxRspTimeDelivery /
1000.0);
    fprintf(csv_file, "%.3f,",
(float)sampling_data.SMaxRspTimeStockLevel /
1000.0);

    /* Max processing time in all time */
    fprintf(csv_file, "%.3f,",
(float)sampling_data.MaxRspTimeNewOrder /
1000.0);
    fprintf(csv_file, "%.3f,",
(float)sampling_data.MaxRspTimePayment /
1000.0);
    fprintf(csv_file, "%.3f,",
(float)sampling_data.MaxRspTimeOrderStatus /
1000.0);
    fprintf(csv_file, "%.3f,",
(float)sampling_data.MaxRspTimeDelivery /
1000.0);
    fprintf(csv_file, "%.3f,",
(float)sampling_data.MaxRspTimeStockLevel /
1000.0);

    /* Ans time to RTE */
    fprintf(csv_file, "%.3f,", (float)ans_new_avr /
1000.0);
    fprintf(csv_file, "%.3f,", (float)ans_pay_avr /
1000.0);
    fprintf(csv_file, "%.3f,", (float)ans_odr_avr /
1000.0);
    fprintf(csv_file, "%.3f,", (float)ans_del_avr /
1000.0);
    fprintf(csv_file, "%.3f,", (float)ans_sto_avr /
1000.0);

    /* Number of connection (no sampling) */
    fprintf(csv_file, "%d", 0);

    fprintf(csv_file, "\n");

    fclose(csv_file);

UNLOCK_SEM:
    UnlockSem(GLBSPM_semaphore);
    return;

```

```

}

.....:
tpapl/ConvInt.c
.....:

/*****
****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* (1) str2int
* (2) str2short
* (3) str2dbl
*
* CREATE by TSL 2002.10.01
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *

*****
****/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define numcheck(num) ( 0x30 <= num && num
<= 0x39 ) /* 0 - 9 */
#define alpcheck(num) ( 0x41 <= num && num
<= 0x5a ) /* A - Z */

/*
str2int :
takes a string, makes sure it's not too long,
and ensures that it
represents an integer.
If it does, the corresponding int value is
returned.

-3: there is not string data.
-2: find not character data.
-1: string data is too many long
*/
int str2int(char *str, int field_len) {
    int x;

    //for warning
    // if(str == 0 || !(x = strlen(str))) return -3;
    if(str == 0 || (x = strlen(str)) == 0) return -3;

    if(x > field_len){
        if (strchr (str, '%') != 0) /* 98.8.3 :-----
----- */
            return -2;
        else
            return -1;
    }
    else{
        for( ; x > 0; x--){
            if (numcheck(str[x-1])) {
                return -2;
            }
        }
        return atoi(str);
    }
}

/*
str2short :
takes a string, makes sure it's not too long,
and ensures that it

```



```

represents an integer.
If it does, the corresponding short value is
returned.

-3: there is not string data.
-2: find not character data.
-1: string data is too many long
*/
short str2short(char *str, int field_len) {
    int x;

    //for warning
    // if(str == 0 || !(x = strlen(str))) return -3;
    if(str == 0 || (x = strlen(str)) == 0) return -3;

    if(x > field_len){
        if (strchr (str, '%') != 0) /* 98.8.3 :-----
    ----- */
            return -2;
        else
            return -1;
    }
    else {
        for( ; x; x--){
            if (!numcheck(str[x-1]))
                return -2;
        }
    }
    x = atoi(str);
    return (short)x;
}

/*
str2dbl :
takes a string, makes sure it's not too long,
and makes sure that it
represents a floating point number.
If so, delete the decimal point.
As a result, the value is increased hundredfold.
this function is returned integer value.

!! This function use Payment transaction only.

-3: there is not string data.
-2: find not character data.
-1: string data is too many long

*/
int str2dbl(char *str, int field_len) {
    int x, len, cnt;
    /* Replaced T.Kato 03.08.20 Bug Fix --over 5
    column integer is memory crush -- */
    /* total 5+2+1(NULL)bytes
    but editing area is 7bytes */
    /* char NUM[7];*/
    char NUM[16];
    /* Replaced end */

    char pointf = 0;
    int fcnt = 2; /* */

    //for warning
    // if(str == 0 || !(x = strlen(str))) return -3;
    if(str == 0 || (x = strlen(str)) == 0) return -3;

    len = x;

    if(x > field_len){
        if (strchr (str, '%') != 0) /* 98.8.3 :-----
    ----- */
            return -2;
        else
            return -1;
    }
}

```

```

else{
    /* check string data */
    for(;x;x--) {
        if(numcheck(str[x-1]));
        else if((str[x-1] == '.') && ((len - x) < 3));
        else if((str[x-1] == '-') && (x == 1));
        else if((str[x-1] == '+') && (x == 1));
        else return -2;
    }
}

/* delete the decimal point. As a result,do
hundredfold the value.*/
for (cnt = 0, x = 0; x < len; x++){

    if ( str[x] == '.' ){
        /* find the decimal point. set point flag.*/
        pointf = 1;
    } else {
        /* set character to work buffer.*/
        NUM[cnt] = str[x]; cnt++;

        /* The figure below the decimal point was
        detected */
        if ( pointf == 1 ) {fcnt--;}
    }

    if ( pointf == 1 && fcnt > 0 ){
        /*There was no figure below the decimal
        point or only one digit was
        found.: ----- */
        for ( ; fcnt > 0; fcnt-- ) {
            NUM[cnt++] = '0';
        }
    }
    else if ( pointf == 0 ) {
        /* There is no decimal point.: -----
    -- */
        NUM[cnt++] = '0'; NUM[cnt++] = '0';
    }

    NUM[cnt] = 0;

    return (atoi(NUM));
}

.....
tpapl/ConvOther.c
.....

/*****
****
*
*
* TPC-C Client Application Program Source
*
*
* Entry Functions *
* (1) para_split *
* (2) checkHTMLform *
* (3) convert_time *
* (4) convert_date *
*
* CREATE by TSL 2002.10.01
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****
****/
#include <stdio.h>
#include <stdlib.h>

```

```

#include <string.h>
#include <time.h>

/*
para_split :
----- (QueryString)-----
-----
-----: -----NULL-----
--
-----NULL-----

Split divides up a string based on the first
instance of a specified
delimiter ('sp'). The first instance of 'sp' is
converted to a NULL
and the address of the first character of the
second half is returned.
Thus the user has the first half (which he
passed in and still has) and
the second half (which was returned) with a
NULL between them. Yay.
(Yes, strtok does this, sort of, but I can't nest
strtok calls.)
*/
char *para_split(char *para, char delimita) {
    char *point = para;;

    /* The address of the delimitation character is
    calculated */
    /* ----- */
    // if ((point = strchr (para, delimita)) == NULL)
    // return (char *)0;

    for(; !(*point == '\0' || *point == delimita);
    point++);
    if (*point == '\0')
        return (char *)0;

    /* The delimitation character is replaced with
    NULL*/
    *point = '\0'; /* -----NULL----- */

    /* The first position of the analyzed variable is
    returned.*/
    return (point + 1); /* ----- */
}

/*
check HTML form

*/
int checkHTMLform( char *str, char *buffer)
{
    char* src = str;
    char* dst = buffer;

    while (*src != '\0'){
        if ( *(src) == '&' ){
            *(dst) = '&'; dst++;
            *(dst) = 'a'; dst++;
            *(dst) = 'm'; dst++;
            *(dst) = 'p'; dst++;
            *(dst) = ';'; dst++;
        }
        else if ( *(src) == '<' ) {
            *(dst) = '&'; dst++;
            *(dst) = 'l'; dst++;
            *(dst) = 't'; dst++;
        }
    }
}

```

```

        *(dst) = ':'; dst++;
    }
    else if ( *(src) == '>' ) {
        *(dst) = '&'; dst++;
        *(dst) = 'g'; dst++;
        *(dst) = 't'; dst++;
        *(dst) = ':'; dst++;
    }
    else if ( *(src) == "" ) {
        *(dst) = '&'; dst++;
        *(dst) = 'q'; dst++;
        *(dst) = 'u'; dst++;
        *(dst) = 'a'; dst++;
        *(dst) = 't'; dst++;
        *(dst) = ':'; dst++;
    }
    else {
        *(dst) = *(src);
        dst++;
    }
    src++;
}

*(dst) = 0;
return ( (unsigned long)dst - (unsigned
long)buffer );
}

//
// The date data is converted. (The time data is
not contained.)
// Numeric data is converted into character string
data.
//
void convert_time( char *save_p, double time )
{
/* Replaced T.Kato 20005.01.21 For thread safe
*/
#if 0
! struct tm* tim;
! time_t tt = ( time_t )time;
!
! tim = localtime( &tt );
#endif
struct tm tm_data;
struct tm* tim = &tm_data;
time_t tt = ( time_t )time;

localtime_r( &tt, tim );
/* Replaced end */

sprintf( save_p, "%02d-%02d-%02d-
%04d %02d:%02d:%02d",
tim->tm_mday, tim->tm_mon+1, tim-
>tm_year + 1900,
tim->tm_hour, tim->tm_min, tim->tm_sec );
}

//
// The date data is converted. (The time data is
contained.)
// Numeric data is converted into character string
data.
//
void convert_date( char *save_p, double time )
{
/* Replaced T.Kato 2005.01.21 For thread safe */
#if 0
! struct tm* tim;
! time_t tt = ( time_t )time;
!
! tim = localtime( &tt );
#endif

```

```

struct tm tm_data;
struct tm* tim = &tm_data;
time_t tt = ( time_t )time;

tim = localtime_r( &tt, tim );
/* Replaced end */

sprintf( save_p, "%02d-%02d-%04d",
tim->tm_mday, tim->tm_mon + 1, tim-
>tm_year + 1900 );
}

.....
tpapl/ConvString.c
.....

/******
****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* (1) int2str
* (2) int3str
* (3) dec2str
* (4) sigdec2str
* (5) str2str
* (6) alp2str
* (7) date2str
* (8) zip2str
* (9) phone2str
*
* CREATE by TSL 2002.10.01
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****
****/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/*
int2str : Converts an integer value to a string of
a specified length and
outputs the string to the memory buffer
supplied.

field = the destination field
field_size = number of characters to output
value = integer to be displayed
*/
void int2str(char *str, int len, int num)
{
int cnt;

for (cnt = len - 1; cnt >= 0; cnt--){

str[cnt] = (char)((num % 10) + '0');
num /= 10;
}

for (cnt = 0; cnt < len-1; cnt++){

if (str[cnt] == '0')
str[cnt] = ' ';
else
return;
}
}

```

```

/*
int3str : Converts an integer value to a string of
a specified length and
outputs the string to the memory buffer
supplied.

field = the destination field
field_size = number of characters to output
value = integer to be displayed
*/
void int3str(char *str, int len, int num)
{
int cnt;

for (cnt = len - 1; cnt >= 0; cnt--){

str[cnt] = (char)((num % 10) + '0');
num /= 10;
}
}

/*
dec2str:
Converts a double precision floating point
value to a string of
a specified length and outputs the string to the
memory buffer supplied.
This routine assumes the following restrictions
apply:
Precision is fixed at 2 places to the right of the
decimal point.
No string length will be less than 4.

field = the destination field
field_size = number of characters to output
value = floating point number to be displayed
*/
void dec2str(char *str, int len, double num)
{
int dec, sign, i, cnt;

/* Replaced T.Kato 2005.01.21 For thread safe */
#if 0
! char *string;
!
! string = ecvt(num, len-1, &dec, &sign);
#endif

char string_buf[17];
char *string = string_buf;

ecvt_r(num, len-1, &dec, &sign, string,
sizeof(string_buf)-1);
/* Replaced end */

/* dec = -----,sign = ---0,----1,string=-----
----- */

if ( dec > 0 ) {
/* if the integer part is not zero ..
Exsample :num data is 1234.56 */
cnt = (len - 3) - dec;

/* -----: "0012" -> " 12"
*/
/* If the high-order digit is zero , zero is
changed at the blank */
for (i = 0; i < cnt; i++){
/* pad with blank in the high part of the
number */
str[i] = ' ';
}
}
}

```

```

    /* The high-order digit set to the output
area: -----*/
    for (; i < (len - 3); i++){
        str[i] = *(string++);
    }
}
else {
    /* If the integer part is zero ... Example:
num data is 0.12 */
    cnt = len - 4;

    for (i = 0; i < cnt; i++){
        /* pad with blank in the high part of the
number */
        str[i] = ' ';
    }
    str[i++] = '0';
}

str[i++] = '.';

for (; dec < 0 && i < len; dec++, i++){
    /* pad with 0's in the high part of the fraction
*/
    str[i] = '0';
}

for (; i < len; i++){
    /* copy the decimal portion (2 places) */
    str[i] = *(string++);
}
}

/*
sigdec2str:
Converts a double precision floating point
value to a string of
a specified length and outputs the string to the
supplied buffer.
If the value is negative, the first character will
be a minus sign (-).

    field = the destination field
    field_size = number of characters to output
    value = floating point number to be displayed
*/
void sigdec2str(char *str, int len, double num)
{
    if (num >= 0.0) {
        str[0] = '.';
        dec2str (&str[1], len - 1, num);
    } else {
        str[0] = '-';
        dec2str (&str[1], len - 1, -num);
    }
}

/*
str2str :
makes sure the string exists and isn't too long.

-1: string data is too many long
-2: find not figure data.
0: there is not string data.
1: normal end
*/
int str2str(char *str, int field_len) {
    int x;

    //for warning
    // if (str == 0 || !(x = strlen (str))) return 0;
    if (str == 0 || (x = strlen (str)) == 0) return 0;

    if(x > field_len) {

```

```

        if ( strchr (str, '%') != 0) /* 98.8.3 :-----
----- */
            return -2;
        else
            return -1;
    }
}
/*
else {
    for(; x ; x--){
        if (!alpcheck(str[x-1]))
            return -2;
    }
}
*/
return 1;
}

/*
alp2str : Outputs a string into the memory
space supplied.

    field = the destination field
    field_size = number of characters to output
    string = alpha string to be displayed
*/
void alp2str(char *str, int len, char *alp)
{
    int cnt;

    cnt = strlen (alp);
    strncpy (str, alp, len); /* copy to destination
area */

    /* len-----*/
    /* If not coming up to the specified length then
set the blank. */
    if (len - cnt > 0)
        memset ( &str[cnt], ' ', len - cnt);
}

/*
date2str : Outputs a date in the supplied buffer
in the following format:
DD-MM-YYYY

    field = the destination field
    date = date to be converted and displayed
*/
void date2str(char *str, char *time)
{
    int year, month, day;

#ifdef DBPRT
    fprintf (test_fp, "date2: %s\n", time);
#endif
    sscanf( time, "%d-%d-%d", &day, &month,
&year );

    int3str (str, 2, day);
    str[2] = '-';
    int3str (&str[3], 2, month);
    str[5] = '-';
    int3str (&str[6], 4, year);
}

/*
zip2str:
Outputs a zipcode in the supplied buffer in the
following format:
XXXXX-XXXX

    str = the destination field
    zip = the zipcode to be output
*/
void zip2str (char *str, char *zip)

```

```

{
    alp2str (str, 5, zip);
    str[5] = '-';
    alp2str (&str[6], 4, &zip[5]);
}

/*
phone2str:
Outputs a phone number in the supplied buffer
in the following format:
XXXXXX-XXX-XXX-XXXX

    str = the destination field
    phone = the phone number to be output
*/
void phone2str(char *str, char *phone)
{
    alp2str (str, 6, phone);
    str[6] = '-';

    alp2str (&str[7], 3, &phone[6]);
    str[10] = '-';

    alp2str (&str[11], 3, &phone[9]);
    str[14] = '-';

    alp2str (&str[15], 4, &phone[12]);
}

.....
tpapl/ErrPage.c
.....

/*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* (1) set_errHTML
* (2) set_SvrApplErr
* (3) set_errpage
*
* CREATE by TSL 2002.10.01
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****/
#include "forlinux.h"

#include <stdio.h>
#include <string.h>
#include "tpcweb.h"

#include "tpapl.h"
#include <pthread.h>
#include <atmi.h>
#include "GlobalArea.h"

/*
set_errHTML :
this function make error message of
application program.
*/
int set_errHTML (char *page, char *err_inf, int
cookie, char *errname) {

```

```

    sprintf(page, errorpage, errname, err_inf,
SOPATH, cookie);

    return 0;
}

#if 0
!/* #ifdef symfo"-----Oracle--Symfo-----
------(set_errHTML)--
! set_or Kerr :
! this function make error message of the
Oracle application program.
!*/
lint set_or Kerr (char *page, char *err_inf, int
cookie) {
!
!#ifdef Symfo
! sprintf(page, symfoerr, err_inf, SOPATH,
cookie);
!#else
! sprintf(page, oraerr, err_inf, SOPATH,
cookie);
!#endif
!
! return 0;
!}
#endif

/*
set_tuxerr :
this function make error message of the TP-
application program.
*/
/* Replaced 03.01.15 */
#if 0
lint set_tuxerr (char *page, char *err_inf, int
cookie) {
#endif
int set_SvrAplErr (char *page, char *err_inf, int
cookie) {
/* Replaced end */

    sprintf(page, tuxerr, err_inf, SOPATH, cookie);

    return 0;
}

/* Error message list : these are notified from
CLINET to RTE */
/* 98.8.3 : ----- */
char errstrings[23][166] = {
"The function you selected doesn't exist.\r\n"
"Don't enter URLs manually!\r\n%s",
/* 0 */

"You seem to have responded to a form that
doesn't exist.\r\n"
"Don't enter URLs manually!\r\n%s",
/* 1 */

"The District ID you entered isn't valid.\r\n%s\r\n"
"It must be an integer in the range 1 to 10.\r\n",
/* 2 */

"The threshold value you entered isn't
valid.\r\n%s\r\n"
"It must be an integer in the range 10 to 20.\r\n",
/* 3 */

"The terminal number you entered isn't
valid.\r\n%s\r\n"
"It must be an integer in the range 1 to %d.\r\n",
/* 4 */

"The Carrier ID you entered isn't valid.\r\n%s\r\n"

```

```

"It must be an integer in the range 1 to 10.\r\n",
/* 5 */

"The Customer ID you entered isn't
valid.\r\n%s\r\n"
"It must be an integer of 4 or fewer digits.\r\n",
//It must be an integer in the 1 to 3000.\r\n",
/* 6 */

"The Customer Last Name you entered isn't
valid.\r\n%s\r\n"
"It must be a string shorter than 16
characters.\r\n", /* 7 */

"The Payment Amount you entered isn't
valid.\r\n%s\r\n"
"It must be a dollar amount, without the dollar
sign,"
" between $1.00 and $5000.00.\r\n",
/* 8 */

"The Customer Warehouse ID you entered isn't
valid.\r\n%s\r\n"
"It must be an integer in the range 1 to %d.\r\n",
/* 9 */

"The Customer District ID you entered isn't
valid.\r\n%s\r\n"
"It must be an integer in the 1 to 10.\r\n",
/* 10 */

"You must enter either a Customer ID or a
Customer Last Name.\r\n"
"You left both fields blank.\r\n%s",
/* 11 */

"The Warehouse ID you entered isn't
valid.\r\n%s\r\n"
"It must be an integer in the range 1 to %d.\r\n",
/* 12 */

"On entry line %d, the data you entered for
the %s field isn't valid.\r\n%s\r\n", /* 13 */

"Supply Warehouse ID",
/* 14 */

"Item ID", /* 15 */
*/

"Quantity", /*
16 */
*/
"Your entry was outside the range.",
/* 17 */
*/
"You didn't entry anything for the field.",
/* 18 */
*/
"Your entry contained too many characters.",
/* 19 */
*/
"The input data is wrong data type, must be
numeric.", /* 20 */
*/
"It must be an integer in the range 1 to %d.",
/* 21 */
*/
"The input data is wrong data type, must be
english capital letter.", /* 22 */
*/
};

/*
set_errpage:

RTE-----
-----

a generic error page generator. If the user
does anything screwy,

```

```

s/he gets here. The function generates an
error page based on the
two errlvl arguments and returns it for the user..

When err_no is 13 or more, Order Line Data is
Abnormal.
(err_no is the error data line number)

98.8.3 : -----
*/
int set_errpage (char *buf, int user, int err_no, int
err_inf, int sub_inf, int sub_inf2) {
char errmsg[1024];
int nchar;
int length;

//for warning
sub_inf;
nchar;

if(err_no >= 13) { /* OrderLine
Data(Neworder) is Abnormal */
switch(err_inf) {
case -5: /* S_W_ID data is abnormal
*/
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[14], errstrings[20]);
sub_inf2 = GLB_Numwh;
break;
case -8: /* S_W_ID data is uninput */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[14], errstrings[18]);
sub_inf2 = GLB_Name;
break;
case -15: /* S_W_ID data is outside
range */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[14], errstrings[17]);
sub_inf2 = GLB_Numwh;
break;

case -1: /* L_ID data is uninput */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[15], errstrings[18]);
sub_inf2 = 100000;
break;
case -6: /* L_ID data is abnormal */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[15], errstrings[20]);
sub_inf2 = 100000;
break;
case -16: /* L_ID data is outside
range */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[15], errstrings[17]);
sub_inf2 = 100000;
break;

case -7: /* Quantity data is abnormal
*/
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[16], errstrings[20]);
sub_inf2 = 10;
break;
case -2: /* Quantity data is uninput */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[16], errstrings[18]);
sub_inf2 = 10;
break;
case -17: /* Quantity data is outside
range */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[16], errstrings[17]);
sub_inf2 = 10;
break;

```

```

default:
    break;
}

length = strlen(errmsg);
sprintf(&errmsg[length], errstrings[21],
sub_inf2);
sprintf(buf, errhtml, errmsg, SOPATH,
user);
}
else if ( err_no == 4 || err_no == 9 || err_no
== 12 ) {

    switch(err_inf) {
        case -3: /* There is not Input data */
            sprintf(errmsg, errstrings[err_no],
errstrings[18], sub_inf2);
            break;

        case -1: /* too many characters */
            sprintf(errmsg, errstrings[err_no],
errstrings[19], sub_inf2);
            break;

        case -2: /* Not all digits */
            sprintf(errmsg, errstrings[err_no],
errstrings[20], sub_inf2 );
            break;

        case -4: /* nothing sub message */
            sprintf(errmsg, errstrings[err_no], " ",
sub_inf2);
            break;

        default: /* Other error */
            sprintf(errmsg, errstrings[err_no],
errstrings[17], sub_inf2);
            break;
    }

    sprintf(buf, errhtml, errmsg, SOPATH,
user);
    // printf("%s", buf);
}
else{
    switch(err_inf) {
        case -3: /* There is not Input data */
            sprintf(errmsg, errstrings[err_no],
errstrings[18]);
            break;

        case -1: /* too many characters */
            sprintf(errmsg, errstrings[err_no],
errstrings[19]);
            break;

        case -2: /* Not all digits */
            if (err_no == 7)
                sprintf(errmsg, errstrings[err_no],
errstrings[22]);
            else
                sprintf(errmsg, errstrings[err_no],
errstrings[20]);

            break;

        case -4: /* nothing sub message */
            sprintf(errmsg, errstrings[err_no], " ");
            break;

        default: /* Other error */
            sprintf(errmsg, errstrings[err_no],
errstrings[17]);
            break;
    }

    // DBGR(sprintf(test_fp, "This Transaction is
parameter ERROR\n"));
    return 0;
}

.....
tpapl/GetTerminalInfo.c
.....

/*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* (1) GetTerminalInfo
* (2) GetConfigFileInfo
*
* CREATE by TSL 2002.12.27
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****/
#include "forlinux.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <atmi.h>

#include "GlobalArea.h"
#include "log.h"
#include "log_level.h"

int GetPrivateProfileString(char* section_name,
char* key_name,
char* default_str, char*
key_data,
int buf_size, char* file_name);
int GetConfFileInfo_GetInt(char* section_name,
char* key_name);
int GetConfFileInfo_GetStr(char* section_name,
char* key_name, char* str);

/*****
* Get configuration file information.
*
* Return Value
* None
*****/

void GetConfFileInfo() {

    if (access(GLB_ConfigFilePath, 0x00) != 0) {
        /* INI file no exist, using default value */
        TpcUserLog(LOG_LCK, "INI file nothing,
using default value");
        GLB_TermBase =
DEFAULT_TERMBASE;
        GLB_Numwh =
DEFAULT_MAXWH;
        GLB_Maxconnect =
DEFAULT_MAXCONNECT;
        GLB_Maxterm =
DEFAULT_MAXTERM;
        GLB_C_FLAG =
DEFAULT_CFLAG;
        strcpy(GLB_TpAplLogPath,
DEFAULT_TPAPL_LOG_PATH);
        strcpy(GLB_SvrAplLogPath,
DEFAULT_SVRAPL_LOG_PATH);
        strcpy(GLB_LogFilePath,
DEFAULT_SVRAPL_LOG_PATH);
        return;
    }
    TpcUserLog(LOG_LCK, "INI file exist, using
spacified parameter\n");

    /* Get execution informations
*/
    /* If undefined key and illigal value, using
default value */
    if ((GLB_TermBase =
GetConfFileInfo_GetInt("TPAPL_INFO",
"Term_Base")) <= 0) {
        GLB_TermBase = DEFAULT_TERMBASE;
    }
    if ((GLB_Numwh =
GetConfFileInfo_GetInt("TPAPL_INFO",
"NumWarehouses")) <= 0) {
        GLB_Numwh = DEFAULT_MAXWH;
    }
    if ((GLB_Maxconnect =
GetConfFileInfo_GetInt("TPAPL_INFO",
"MaxUsers")) <= 0) {
        GLB_Maxconnect =
DEFAULT_MAXCONNECT;
    }
    if ((GLB_Maxterm =
GetConfFileInfo_GetInt("TPAPL_INFO",
"MaxTerm of Client")) <= 0) {
        GLB_Maxterm = DEFAULT_MAXTERM;
    }
    if ((GLB_C_FLAG =
GetConfFileInfo_GetInt("TPAPL_INFO",
"CONTROL_Flag")) == -1) {
        GLB_C_FLAG = DEFAULT_CFLAG;
    }
    if (GetConfFileInfo_GetStr("TPAPL_INFO",
"LogPath", GLB_TpAplLogPath) != 0) {
        strcpy(GLB_TpAplLogPath,
DEFAULT_TPAPL_LOG_PATH);
    }
    if (GetConfFileInfo_GetStr("SVRAPL_INFO",
"LogPath", GLB_SvrAplLogPath) != 0) {
        strcpy(GLB_SvrAplLogPath,
DEFAULT_SVRAPL_LOG_PATH);
    }

    strcpy(GLB_LogFilePath,
GLB_TpAplLogPath);
}

/*****
* Get information in the CONFIG file for integer
value */
/*****

```

```

int GetConfFileInfo_GetInt(char* section_name,
char* key_name) {

    char value_buf[64];
    int i;

    for (i = 0; i < 3; i++) {
        GetPrivateProfileString(section_name,
key_name, "",
                                value_buf, sizeof(value_buf),
GLB_ConfigFilePath);
        if (value_buf[0] == "") {
            /* if Key is nothing, retry getting */
            continue;
        }
        break;
    }
#ifdef PUT_INF_LOG
    TpcUserLog(LOG_LCK, "CONFIG file
information [%s %s]=[%s]", section_name,
key_name, value_buf);
#endif
    if (value_buf[0] == "") {
        /* Target key was nothing */
        return (-1);
    }
    return(atoi(value_buf));
}

/*-----*/
/* Get information in the CONFIG file for string
value */
/*-----*/
int GetConfFileInfo_GetStr(char* section_name,
char* key_name, char* str) {

    int i;
    char value_buf[1024];

    for (i = 0; i < 3; i++) {
        GetPrivateProfileString(section_name,
key_name, "",
                                value_buf, sizeof(value_buf),
GLB_ConfigFilePath);
        if (value_buf[0] == "") {
            /* if Key is nothing, retry getting */
            continue;
        }
        break;
    }
#ifdef PUT_INF_LOG
    TpcUserLog(LOG_LCK, "CONFIG file
information [%s %s]=[%s]", section_name,
key_name, value_buf);
#endif
    if (value_buf[0] == "") {
        /* Target key was nothing */
        return (-1);
    }
    strcpy(str, value_buf);
    return(strlen(value_buf));
}

.....
tpapl/GlobalArea.c
.....

/*-----*/
*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Global Area definition for common.
*
*
* CREATE by TSL 2003.12.15
*
*
* Entry Functions

```

```

* Global Area definition for common.
*
*
* CREATE by TSL 2003.12.15
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
****/
#include "forlinux.h"

#include <pthread.h>
#include <atmi.h>

#ifdef DBPRT /* for debug */
FILE *test_fp;
#endif

/* Environment of operation */
int GLB_TermBase;
int GLB_Numwh;
int GLB_Maxconnect;
int GLB_Maxterm;
int GLB_C_FLAG;
char GLB_TpAplLogPath[MAX_PATH];
char
GLB_SvrAplLogPath[MAX_PATH];

/* Configuration file path */
char
GLB_ConfigFilePath[MAX_PATH];

/* Thread key */
pthread_key_t GLB_ThreadKey;

/* Log information */
char GLB_LogFilePath[MAX_PATH];
int GLB_LogSemId;

/* TUXEDO context */
#if 0 /* * 2006.03.29 T.Motoo: Changed the type
of "GLB_TpContext". */
!TPCONTEXT_T GLB_TpContext = 0;
#endif
TPCONTEXT_T *GLB_TpContext =
NULL;

/*
* 2006.03.29 T.Motoo: Added.
*/
int GLB_ThreadLimit = 1;

.....
tpapl/GlobalArea.h
.....

/*-----*/
*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Global Area definition for common.
*
*
* CREATE by TSL 2003.12.15
*
*
* Entry Functions

```

```

* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
****/

#ifndef GLOBALAREA_H
#define GLOBALAREA_H

#ifdef DBPRT /* for debug */
extern FILE *test_fp;
#endif

extern int GLB_TermBase;
#define DEFAULT_TERMBASE 1
extern int GLB_Numwh;
#define DEFAULT_MAXWH 2000
extern int GLB_Maxconnect;
#define DEFAULT_MAXCONNECT
20000
extern int GLB_Maxterm;
#define DEFAULT_MAXTERM 2000
extern int GLB_C_FLAG;
#define DEFAULT_CFLAG 0
extern char
GLB_TpAplLogPath[MAX_PATH];
extern char
GLB_SvrAplLogPath[MAX_PATH];

/* Configuration file path */
extern char
GLB_ConfigFilePath[MAX_PATH];

/* Thread key */
extern pthread_key_t GLB_ThreadKey;

/* Log information */
extern char
GLB_LogFilePath[MAX_PATH];
extern int
GLB_LogSemId;

/* TUXEDO context */
#if 0 /* * 2006.03.29 T.Motoo: Changed the type
of "GLB_TpContext". */
!extern TPCCONTEXT_T
GLB_TpContext;
#endif
extern TPCCONTEXT_T* GLB_TpContext;

/*
* 2006.03.29 T.Motoo: "GLB_ThreadLimit" and
"TUXCDPERCTXT" were added.
*/
extern int
GLB_ThreadLimit;

/*
* Call descriptors per context (TUXEDO)
*/
#define TUXCDPERCTXT 50

#endif // GLOBALAREA_H

.....
tpapl/InitThreadEnv.c
.....

/*-----*/
*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions

```

```

*
* Entry Functions
* (1) GetThreadKey
* (2) CreateTuxEnv
* (3) DestroyThread
* (4) FreeThreadKey
* (5) GetThreadCntl
* (6) RegisTuxApl
* (7) TermChildProcess
* (7) PlainCleanup
*
* CREATE by TSL 2003.12.16
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003
*
*****
****/
#include "forlinux.h"
#include <pthread.h>
#include <atmi.h>
#include <unistd.h>

#include "httpd.h"
#include "http_config.h"
#include "http_protocol.h"
#include "ap_config.h"
#include "ap_compat.h"
#include "ap_mpm.h" /* 2006.03.29 T.Motoo:
Added for ap_mpm_query */

#include "tpccinf.h"
#include "trans.h"
#include "ThreadCntl.h"
#include "GlobalArea.h"
#include "TpAplDBDependPrototype.h"
#include "log.h"
#include "log_level.h"

/*****
* Get thread key.
* Return Value
* 0 : Success
* !0 : Fail
*
*****
****/
int GetThreadKey() {

    int ret_code;
    void DestroyThread(void* p);

#ifdef PUT_INF_LOG
    TpcUserLog (LOG_INF, "Thread key
creating start [GetThreadKey]\n");
#endif

    /* Create the thread key */
    if ((ret_code =
pthread_key_create(&GLB_ThreadKey,
DestroyThread)) != 0) {
        TpcUserLog (LOG_ERR, "Thread key fail
to creat [error:%d]\n", ret_code);
        return -1;
    }

#ifdef PUT_INF_LOG
    TpcUserLog (LOG_INF, "Thread key
creating end [GetThreadKey= %d]\n",
GLB_ThreadKey);
#endif

    return 0;
}

}

/*****
*
* Initialize environment for Thread.
*
* Return Value
* !0 : Success(pointer of
THREAD_CNTL_INFO)
* 0 : Fail
*
*****
****/
/*
* 2006.03.29 T.Motoo: The argument was
added. "id" is ID of connection managed
* by apache. Unique at any point in
time.
*/
#if 0
!THREAD_CNTL_INFO* CreateThreadEnv() {
#endif
THREAD_CNTL_INFO* CreateThreadEnv(int id)
{

    THREAD_CNTL_INFO* ThreadCntlInfo;

    void* ift_buf;
    char* resp_buf;
    char* query_str;
    int buf_leng;

#define BUF_TYPE "CARRAY"

    if ((ThreadCntlInfo =
(THREAD_CNTL_INFO*)pthread_getspecific(GL
B_ThreadKey)) == NULL) {

#ifdef PUT_INF_LOG
        TpcUserLog (LOG_INF, "Thread initialize
started \n");
#endif
        /* First execution in this thread */
#ifdef SCRTEST
        /* Regist context */
        /*
* 2006.03.29 T.Motoo: Modified because child
process came to have one or more
* contexts.
*/
        #if 0
        ! if (tpsetctx(GLB_TpContext, 0) == -1) {
        #endif
            if (tpsetctx(GLB_TpContext{((id %
GLB_ThreadLimit) /
TUXCDPERCTXT), 0) ==
-1) {
                TpcUserLog (LOG_ERR, "tpsetctx()
failed\n");
                return(0);
            }
        #endif

        /* Get query data area */
#ifdef USEPOOL_QUERY
        if ((query_str =
(char*)malloc(QUERY_STR_SIZE)) == NULL) {
            TpcUserLog (LOG_ERR, "malloc() failed
for query string buffer (size=%d)\n",
QUERY_STR_SIZE);
            return(0);
        }
        #else
        query_str = NULL;
        #endif
    }

#endif
    /* Get response editing area */
    if ((resp_buf =
(char*)malloc(RESP_BUF_SIZE)) == NULL) {
        TpcUserLog (LOG_ERR, "malloc() failed
for response editing buffer (size=%d)\n",
RESP_BUF_SIZE);
        return(0);
    }

    /* Get Thread control information area */
    if ((ThreadCntlInfo =
(THREAD_CNTL_INFO*)malloc(sizeof(THREAD
_CNTL_INFO))) == NULL) {
        TpcUserLog (LOG_ERR, "malloc() failed
for THREAD_CNTL_INFO (size=%d)\n",
sizeof(THREAD_CNTL_INFO));
        return(0);
    }

    /* Get the TUXEDO interface data area */
    buf_leng = (GetGenericDataLen() + 16) &
0xfffffff;

#ifdef CONST_TUX_BUF

#ifdef SCRTEST
        if ((ift_buf = (void *)tpalloc("CARRAY",
NULL, buf_leng)) == NULL) {
            TpcUserLog (LOG_ERR, "tpalloc() failed
for interface data buffer (size=%d)\n", buf_leng);
            return(0);
        }
    #else
        if ((ift_buf = (void *)calloc (buf_leng, 1)) ==
NULL) {
            TpcUserLog (LOG_ERR, "calloc() failed
for interface data buffer (size=%d)\n", buf_leng);
            return(0);
        }
    #endif

    #endif

    #else
        ift_buf = 0;
    #endif

    /* Set each pointer */
    ThreadCntlInfo->TrxDat = ift_buf;
    ThreadCntlInfo->TrxDatLeng = buf_leng;
    ThreadCntlInfo->QueryData = query_str;
    ThreadCntlInfo->RespBuf = resp_buf;

    /* Set thread data pointer */
    if (pthread_setspecific(GLB_ThreadKey,
(void*)ThreadCntlInfo) != 0) {
        TpcUserLog (LOG_ERR,
"pthread_setspecific() failed for
THREAD_CNTL_INFO setting \n");
#ifdef CONST_TUX_BUF

#ifdef SCRTEST
            tpfree(ift_buf);
        #else
            free(ift_buf);
        #endif

    #endif

        return(0);
    }

#ifdef PUT_INF_LOG
    TpcUserLog (LOG_INF, "Thread initialize
ended [thread key:%d]\n", GLB_ThreadKey);
#endif
}

```

```

return(ThreadCntlInfo);
}

/*****
* Destroy thread, then free allocate area.
*
* Return Value
* NONE
*****/

void DestroyThread(void* p) {

    THREAD_CNTL_INFO* ThreadCntlInfo;

#ifdef PUT_INF_LOG
    TpcUserLog(LOG_INF, "Thread
terminated start\n");
#endif

    if (p != NULL) {
        ThreadCntlInfo =
        (THREAD_CNTL_INFO*)p;

        if (ThreadCntlInfo->TrxDData != 0)
#ifdef SCRTEST
            tpfree(ThreadCntlInfo->TrxDData);
#else
            free(ThreadCntlInfo->TrxDData);
#endif

#ifdef USEPOOL_QUERY
        if (ThreadCntlInfo->QueryData != 0)
            free((void*)ThreadCntlInfo->QueryData);
#endif

        if (ThreadCntlInfo->RespBuf != 0)
            free((void*)ThreadCntlInfo->RespBuf);
        free((void*)ThreadCntlInfo);
        ThreadCntlInfo = 0;
        if (pthread_setspecific(GLB_ThreadKey,
        (void*)ThreadCntlInfo) != 0) {
            TpcUserLog(LOG_ERR,
            "pthread_setspecific() failed for Thread
            destroyed\n");
        }
    }

#ifdef PUT_INF_LOG
    TpcUserLog(LOG_INF, "Thread terminate
ended [TSD value:%08x]\n", (unsigned long)p);
    return;
#endif
}

/*****
* Free thread key.
*
* Return Value
* NONE
*****/

void FreeThreadKey() {
    int ret_code;

    if ((ret_code =
    pthread_key_delete(GLB_ThreadKey)) != 0) {
        TpcUserLog(LOG_ERR,
        "pthread_key_delete() failed [ret_code=%d]\n",
        ret_code);
    }
}

```

```

/*****
****
* Get Thread_CNTL_INFO pointer in my thread.
*
* Return Value
* !0 : Success(pointer of
THREAD_CNTL_INFO)
* 0 : Fail
*****/

THREAD_CNTL_INFO* GetThreadCntl() {
    THREAD_CNTL_INFO* ThreadCntlInfo;

    if ((ThreadCntlInfo =
    (THREAD_CNTL_INFO*)pthread_getspecific(GL
    B_ThreadKey)) == NULL) {
        TpcUserLog(LOG_ERR, "Thread cntrol
        information is not allocated.\n");
        return 0;
    }

#ifdef CONST_TUX_BUF
    /* Nothing to do */
#else
#ifdef SCRTEST
        if (ThreadCntlInfo->TrxDData = (char
        *)tpalloc("CARRAY", NULL, ThreadCntlInfo-
        >TrxDDataLeng)) == NULL) {
            TpcUserLog(LOG_ERR, "tpalloc() failed
            for interface data buffer (size=%d)\n",
            ThreadCntlInfo->TrxDDataLeng);
            return(0);
        }
    }
#else
        if (ThreadCntlInfo->TrxDData = (char
        *)calloc( ThreadCntlInfo->TrxDDataLeng, 1)) ==
        NULL) {
            TpcUserLog(LOG_ERR, "calloc() failed for
            interface data buffer (size=%d)\n",
            ThreadCntlInfo->TrxDDataLeng);
            return(0);
        }
    }
#endif
}

return ThreadCntlInfo;
}

/*****
****
* Free TUXEDO interface buffer.
*
* Return Value
* NONE
*****/

void FreeTuxBuffer(THREAD_CNTL_INFO*
ThreadCntlInfo) {

#ifdef CONST_TUX_BUF
    /* No free buffer */
#else
        if (ThreadCntlInfo->TrxDData != 0) {
#ifdef SCRTEST
            tpfree(ThreadCntlInfo->TrxDData);
#else
            free(ThreadCntlInfo->TrxDData);
#endif
        }
        ThreadCntlInfo->TrxDData = 0;
    }
}

#ifdef CONST_TUX_BUF
    /* Nothing to do */

```

```

return;
}

/*****
****
* Regist TUXEDO aplication.
*
* Return Value
* !0 : Success
* 0 : Fail
*****/

/*
* 2006.03.29 T.Motoo: The argument was
added. "p" is pool of apache. Moreover,
* some variables were added.
*/
#if 0
ITPCONTEXT_T RegistTuxApl() {
    !
    ! TPCONTEXT_T ctx = 0;
}
#endif

TPCONTEXT_T *RegistTuxApl(void *p) {

    TPCONTEXT_T *ctx = NULL; /* Contexts
*/
    int num_of_ctx = 0; /* Contexts per
child */
    int thr_per_child = 0; /* Threads per child
*/
    int i; /* Uses as counter */

    static TPINIT *tpinf = 0;

    if (tpinf == 0) {
        /* Get Initialize information area for tpinit() */
        if ((tpinf = (TPINIT *)tpalloc("TPINIT", NULL,
        sizeof(TPINIT))) == NULL) {
            TpcUserLog(LOG_ERR, "tpalloc failed
            for tpinit() (%s)\n", tpstrerror(tperrno));
            return 0;
        }
    }

    /* Execute tpinit() (Regist TUXEDO
    aplication) */
    memset((void*)tpinf, 0x00, sizeof(TPINIT));
    tpinf->flags=TPMULTICONTEXTS;

    #if 0 /* 2006.03.29 T.Motoo: Changed to get one
    or more contexts. */
        ! if (tpinit(tpinf) < 0) {
            ! /* tpinit() abnormal end */
            ! TpcUserLog(LOG_ERR, "tpinit() faild
            (%s)\n", tpstrerror(tperrno));
            ! return 0;
            ! }
        ! }
        !
        ! /* Get my context */
        ! if (tpgetctx(&ctx, 0) == -1) {
            ! TpcUserLog(LOG_ERR, "Failed to get
            Tuxedo context (%s)\n", tpstrerror(tperrno));
            ! return 0;
            ! }
        #endif
        /*
        * Gets "ThreadsPerChild" and
        "ThreadLimit".
        */

    ap_mpm_query(AP_MPMO_MAX_THREADS,
    &thr_per_child);

```



```
ap_mpm_query(AP_MPMQ_HARD_LIMIT_THR
EADS, &GLB_ThreadLimit);
```

```
/*
 * Gets the number of contexts.
 */
num_of_ctx = ((thr_per_child - 1) /
TUXCDPERCTXT) + 1;
```

```
/*
 * Allocates the memory for contexts in the
pool.
 */
ctx = (TPCONTEXT_T
*)ap_palloc((apr_pool_t*)p,
sizeof(TPCONTEXT_T)
* num_of_ctx);
```

```
if (ctx == NULL) {
    TpcUserLog(LOG_ERR, "ap_palloc
failed for contexts\n");
    return 0;
}
```

```
for (i = 0; i < num_of_ctx; i++) {
    /*
     * Joins the TUXEDO.
     */
    if (tpinit(tpinf) < 0) {
        /* tpinit() abnormal end */
        TpcUserLog(LOG_ERR, "tpinit() failed
(%)s\n",
                tpstrerror(tperrno));
        return 0;
    }
```

```
/*
 * Gets the context.
 */
if (tpgetctx((ctx + i), 0) == -1) {
    TpcUserLog(LOG_ERR, "Failed to
get Tuxedo context (%)s\n",
                tpstrerror(tperrno));
    return 0;
}
}
```

```
return ctx;
}
```

```
*****
 * Termmnate child process.
 *
 * Return Value
 * Always SUCCESS
*****
```

```
apr_status_t TermChildProcess(void* p) {
```

```
#ifdef PUT_INF_LOG
    TpcUserLog(LOG_INF, "Child process
terminated start. \n");
#endif
```

```
/* Leave from TUXEDO aplication */
if (GLB_TpContext != 0) {
    if (tpterm() == -1) {
        TpcUserLog(LOG_ERR, "tpterm() failed
for Thread destroyed\n");
    }
    GLB_TpContext = 0;
}
```

```
/* Delete TSD key */
FreeThreadKey();
```

```
#ifdef PUT_INF_LOG
    TpcUserLog(LOG_INF, "Child process
terminated end. \n");
#endif
```

```
return(APR_SUCCESS);
```

```
}
/*****
 * Plain cleanup.
 * Return Value
 * Always SUCCESS
*****
```

```
apr_status_t PlainCleanup(void* p) {
```

```
/* Notheng to do */
return(APR_SUCCESS);
}
```

```
.....
tpapl/MakeShell_lib
.....
#!/bin/sh
cd /home/tpc/client_apl/tpapl
make -f Makefile_lib > make_result.txt 2>&1
```

```
.....
tpapl/MakeShell_tpapl
.....
#!/bin/sh
```

```
# Output object from library
cd /home/tpc/client_apl/tpapl/trnexe
echo "==" > ./make_result.txt
echo "=====" Output object"
>> ./make_result.txt
```

```
rm *.o >> ./make_result.txt
ar -xv libtrnexe_$1.a ConvTime.o
>> ./make_result.txt
ar -xv libtrnexe_$1.a CreateTranErrReason.o
>> ./make_result.txt
ar -xv libtrnexe_$1.a TestFunction.o
>> ./make_result.txt
ar -xv libtrnexe_$1.a TransactionDataLen.o
>> ./make_result.txt
ar -xv libtrnexe_$1.a TrxDelivery.o
>> ./make_result.txt
ar -xv libtrnexe_$1.a TrxNewOrder.o
>> ./make_result.txt
ar -xv libtrnexe_$1.a TrxOrderStatus.o
>> ./make_result.txt
ar -xv libtrnexe_$1.a TrxPayment.o
>> ./make_result.txt
ar -xv libtrnexe_$1.a TrxStockLevel.o
>> ./make_result.txt
```

```
.....
tpapl/Makefile
.....
##
## Makefile -- Build procedure for sample tpapl
Apache module
## Autogenerated via ``apxs -n tpapl -g``.
##
```

```
builddir=.
top_srcdir=/etc/httpd
top_builddir=/etc/httpd
include /usr/lib/httpd/build/special.mk
```

```
# the used tools
APXS=apxs
APACHECTL=apachectl
```

```
# additional defines, includes and libraries
#DEFS=-Dmy_define=my_value
#INCLUDES=-Imy/include/dir
#LIBS=-Lmy/lib/dir -lmylib

# the default target
all: local-shared-build
```

```
ar -rv libtpapl.a ./trnexe/ConvTime.o
>> make_result.txt
ar -rv libtpapl.a ./trnexe/CreateTranErrReason.o
>> make_result.txt
ar -rv libtpapl.a ./trnexe/TestFunction.o
>> make_result.txt
ar -rv libtpapl.a ./trnexe/TransactionDataLen.o
>> make_result.txt
ar -rv libtpapl.a ./trnexe/TrxDelivery.o
>> make_result.txt
ar -rv libtpapl.a ./trnexe/TrxNewOrder.o
>> make_result.txt
ar -rv libtpapl.a ./trnexe/TrxOrderStatus.o
>> make_result.txt
ar -rv libtpapl.a ./trnexe/TrxPayment.o
>> make_result.txt
ar -rv libtpapl.a ./trnexe/TrxStockLevel.o
>> make_result.txt
```

```
ar -rv libtpapl.a ../common/log.o
>> make_result.txt
ar -rv libtpapl.a ../common/sema.o
>> make_result.txt
ar -rv libtpapl.a ../common/shmem.o
>> make_result.txt
ar -rv
libtpapl.a ../common/GetPrivateProfileString.o
>> make_result.txt
```

```
# Make TPAPL
echo "==" >> make_result.txt
echo "=====" Make "=====" >>
make_result.txt
touch mod_tpapl.c >> make_result.txt
make -f Makefile_tpapl >>
make_result.txt 2>&1
## Not install ##make -f Makefile_tpapl install
>> make_result.txt 2>&1
```

```
# Check undefined symbol
echo "==" >> make_result.txt
echo "=====" mod_tpapl.so information "====="
>> make_result.txt
ldd -r ./libs/mod_tpapl.so >> make_result.txt
2>&1
```

```
.....
tpapl/Makefile
.....
```

```
##
## Makefile -- Build procedure for sample tpapl
Apache module
## Autogenerated via ``apxs -n tpapl -g``.
##
```

```
builddir=.
top_srcdir=/etc/httpd
top_builddir=/etc/httpd
include /usr/lib/httpd/build/special.mk
```

```
# the used tools
APXS=apxs
APACHECTL=apachectl
```

```
# additional defines, includes and libraries
#DEFS=-Dmy_define=my_value
#INCLUDES=-Imy/include/dir
#LIBS=-Lmy/lib/dir -lmylib

# the default target
all: local-shared-build
```

```

# install the shared object file into Apache
install: install-modules

# cleanup
clean:
  -rm -f mod_tpapl.o mod_tpapl.lo
  mod_tpapl.slo mod_tpapl.la

# simple test
test: reload
  lynx -mime_header http://localhost/tpapl

# install and activate shared object by reloading
Apache to
# force a reload of the shared object file
reload: install restart

# the general Apache start/restart/stop
# procedures
start:
  $(APACHECTL) start
restart:
  $(APACHECTL) restart
stop:
  $(APACHECTL) stop

.....
tpapl/Makefile_lib
.....

#-----
-----
# Makefile : Makefile for TpApl library on Linux.
#
# Created by TSL 2003.12.22
#
# All Right Rserverd, Copyright Co, FUJITSU
LIMITED 2003-2004.
#-----
-----

# GCC compile configurations
AR = ar
ARFLAGS = rv

#CFLAGS note:
# CONST_TUX_BUF defined : TUXEDO
interface buffer is created when thread initialize.
# CONST_TUX_BUF undefined : TUXEDO
interface buffer is created when transaction
processing start,
# and freed when transaction
procesing end.
# USEPOOL_QUERY define : Use query data
area in apache pool.
# USEPOOL_QUERY undefined : Allocate the
query data area, and copied query data form
apache pool.
#CFLAGS = -Wall
#CFLAGS = -Wall -DCONST_TUX_BUF
CFLAGS = -Wall -O2 -DCONST_TUX_BUF -
DUSEPOOL_QUERY
CC = gcc

# Define macros
DMACRO =

# home directory.
TOPDIR = /home/tpc/client_apl
TUXDIR = /usr/local/BEA/tuxedo8.1
APADIR = /usr/include/httpd
APAODIR = /usr/include/apr-0
APLDIR = $(TOPDIR)/tpapl

```

```

# include directory
COM_INC = -I$(TOPDIR)/common
TUX_INC = -I$(TUXDIR)/include
APA_INC = -I$(APADIR)
APA0_INC = -I$(APAODIR)
APL_INC = -I$(APLDIR)

# header file directory
HDFDIR = $(APLDIR)
COMDIR = $(TOPDIR)/common

INCLUDE = $(APL_INC) $(COM_INC) -
I$(APAODIR) $(APA_INC) $(TUX_INC)
INCFILE = $(APLDIR)/delpage.h \
$(APLDIR)/GlobalArea.h \
$(APLDIR)/log_level.h \
$(APLDIR)/menupage.h \
$(APLDIR)/newpage.h \
$(APLDIR)/odrpge.h \
$(APLDIR)/paypage.h \
$(APLDIR)/stpage.h \
$(APLDIR)/ThreadCntl.h \
$(APLDIR)/tpapl.h \
$(APLDIR)/TpAplDBDependPrototype.h \
$(APLDIR)/TpAplPrototype.h \
$(APLDIR)/tpccinf.h \
$(APLDIR)/tpcinweb.h \
$(APLDIR)/tpcweb.h \
$(APLDIR)/trans.h \
$(APLDIR)/SampleInfo.h \
$(COMDIR)/log.h \
$(COMDIR)/sema.h

# target object
OBS = TpAplHandler.o ClientMonitor.o
ConvInt.o ConvOther.o ConvString.o \
  ErrPage.o GetTerminalInfo.o GlobalArea.o
InitThreadEnv.o tpaplFunction.o
ARCH_LIB = $(APLDIR)/libtpapl.a

$(ARCH_LIB) : $(OBS) $(INCFILE)
$(AR) $(ARFLAGS) $(ARCH_LIB) $(OBS)

.SUFFIXES: .o .c
.c.o:
  $(CC) -o $@ -c $(CFLAGS) $(INCLUDE)
$(DMACRO) $<

$(OBS) : $(INCFILE)

clean:
  rm $(TIER_ARCH_LIB) $(TIER_OBJS)

.....
tpapl/Makefile_tpapl
.....

#-----
-----
# Makefile : Makefile for TpApl library on Linux.
#
# Created by TSL 2003.12.18
#
# All Right Rserverd, Copyright Co, FUJITSU
LIMITED 2003-2004.
#-----
-----

builddir=.

```

```

top_srcdir=/etc/httpd
top_builddir=/etc/httpd
include /usr/lib/httpd/build/special.mk

# the used tools
APXS=apxs
APACHECTL=apachectl

# additional defines, includes and libraries
#DEFS=-Dmy_define=my_value
#INCLUDES=-Imy/include/dir
#LIBS=-Lmy/lib/dir -lmylib

TPAHOME = /home/tpc/client_apl
TUXHOME = /usr/local/BEA/tuxedo8.1

#LIBS=-L$(TPAHOME)/tpapl -L$(TUXHOME)/lib
\
# -ltpapl \
# -ltux -lbuft -lfml -lfml32 -lengine \
# -ldl -lpthread

LIBS=-L$(TPAHOME)/tpapl -L$(TUXHOME)/lib \
-ltpapl \
-ltux

# the default target
all: local-shared-build

# install the shared object file into Apache
install: install-modules

# cleanup
clean:
  -rm -f mod_tpapl.o mod_tpapl.lo
  mod_tpapl.slo mod_tpapl.la

# simple test
test: reload
  lynx -mime_header http://localhost/tpapl

# install and activate shared object by reloading
Apache to
# force a reload of the shared object file
reload: install restart

# the general Apache start/restart/stop
# procedures
start:
  $(APACHECTL) start
restart:
  $(APACHECTL) restart
stop:
  $(APACHECTL) stop

.....
tpapl/SampleInfo.h
.....

/*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Performance information definition
*
* CREATE by TSL 2004.01.18
*
*
*****/

```

```

* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2004 *

*****
*/
/* Performans sampling faunctions */
int ClientMonitor(int func_no, char*
html_buf);
void ClientLogCheck(char* html_buf);
void ClientSetSample(char* html_buf);
void ClientInfSample(char* html_buf);
void ClientSampleInit();
void ClientSampleSelfCsv(time_t cur_sec);

/* Structure of performance sampling area */
typedef struct _sampling_data {
// Number of DB server executed transactions
unsigned int NumNewOrder;
unsigned int NumPayment;
unsigned int NumOrderStatus;
unsigned int NumDelivery;
unsigned int NumStockLevel;

// Response time (ms) from DB server (total
time in sampling interval)
unsigned int RspTimeNewOrder;
unsigned int RspTimePayment;
unsigned int RspTimeOrderStatus;
unsigned int RspTimeDelivery;
unsigned int RspTimeStockLevel;

// Max response time (ms) from DB server
(total time in sampling interval)
unsigned int SMaxRspTimeNewOrder;
unsigned int SMaxRspTimePayment;
unsigned int SMaxRspTimeOrderStatus;
unsigned int SMaxRspTimeDelivery;
unsigned int SMaxRspTimeStockLevel;

// Number of request from RTE
unsigned int NumReqNewOrder;
unsigned int NumReqPayment;
unsigned int NumReqOrderStatus;
unsigned int NumReqDelivery;
unsigned int NumReqStockLevel;

// Answer time (ms) to RTE (total time in
sampling interval)
unsigned int AnsNewOrder;
unsigned int AnsPayment;
unsigned int AnsOrderStatus;
unsigned int AnsDelivery;
unsigned int AnsStockLevel;

// NOTE : Under the members are not cleared
by sampling interval.
// Max response time (ms) from DB server (all
of sampling time)
unsigned int MaxRspTimeNewOrder;
unsigned int MaxRspTimePayment;
unsigned int MaxRspTimeOrderStatus;
unsigned int MaxRspTimeDelivery;
unsigned int MaxRspTimeStockLevel;

// Number of executing and waiting
transactions
unsigned int NumQueNewOrder;
unsigned int NumQuePayment;
unsigned int NumQueOrderStatus;
unsigned int NumQueDelivery;
unsigned int NumQueStockLevel;

// Self sampling information
char CsvFilePath[MAX_PATH];
unsigned int CsvOutTime;

```

```

unsigned int SamplingInterval;
int SelfSamplingOutput;
#define SELFOUTPUT_ENABLE 1
#define SELFOUTPUT_DISABLE 0
int DataSampling;
#define DATASAMPLE_ENABLE 0
#define DATASAMPLE_DISABLE 1

// wait timer for 2tier.
unsigned int WaitTimer;

} SAMPLING_DATA;

/* ===== */
/* Macros */
/* ===== */
/* Path */
#define SAMPLING_SEMPATH
"/home/tpc/conf"
#define SAMPLING_SHMPATH
"/home/tpc/bin"

/* Sampling informaion */
#define MAC_SampleGlobalArea \
int GLBSMP_shared_mem = 0; \
SAMPLING_DATA* \
GLBSMP_shared_mem = 0;

extern int GLBSMP_shared_mem;
extern SAMPLING_DATA* \
GLBSMP_shared_mem;

/* Initialize semafore and shared memory */
#define MAC_SampleInitParent \
GLBSMP_shared_mem = \
InitSem(SAMPLING_SEMPATH, \
SEM_SAMPLING_PERFORMANCE); \
GLBSMP_shared_mem = \
(SAMPLING_DATA*)InitShmem(SAMPLING_SH \
MPATH, \
SHMEM_SAMPLING_PERFORMANCE, \
sizeof(SAMPLING_DATA)); \
memset(GLBSMP_shared_mem, 0x00, \
sizeof(SAMPLING_DATA));

#define MAC_SampleInitChild \
GLBSMP_shared_mem = \
GetSem(SAMPLING_SEMPATH, \
SEM_SAMPLING_PERFORMANCE); \
GLBSMP_shared_mem = \
(SAMPLING_DATA*)GetShmem(SAMPLING_S \
HMPATH, \
SHMEM_SAMPLING_PERFORMANCE, \
sizeof(SAMPLING_DATA));

#define MAC_SampleInitPerformance \
ClientSampleInit();

/* Fancions work area */
#define MAC_SampleWork \
struct timeval \
sample_start_time; \
struct timeval \
sample_end_time; \
unsigned int el_time;

/* Get start time */
#define MAC_SampleStartTime \
if (GLBSMP_shared_mem- \
>DataSampling == DATASAMPLE_ENABLE) { \
gettimeofday(&sample_start_time, \
NULL); \
}

```

```

/*sleep(10);*/ \
}

/* Transaction queue up/down */
#define MAC_SampleQueueUp(count_area) \
if (GLBSMP_shared_mem- \
>DataSampling == DATASAMPLE_ENABLE) { \
LockSem(GLBSMP_shared_mem); \
GLBSMP_shared_mem- \
>count_area++; \
UnlockSem(GLBSMP_shared_mem); \
/*sleep(10);*/ \
}

#define \
MAC_SampleQueueDown(count_area) \
if (GLBSMP_shared_mem- \
>DataSampling == DATASAMPLE_ENABLE) { \
LockSem(GLBSMP_shared_mem); \
GLBSMP_shared_mem->count_area-- \
; \
UnlockSem(GLBSMP_shared_mem); \
}

/* Compute execution time */
#define MAC_SampleExecuteTime \
if (GLBSMP_shared_mem- \
>DataSampling == DATASAMPLE_ENABLE) { \
gettimeofday(&sample_end_time, \
NULL); \
el_time = ((unsigned \
int)sample_end_time.tv_sec*1000 + (unsigned \
int)sample_end_time.tv_usec/1000) \
- ((unsigned \
int)sample_start_time.tv_sec*1000 + (unsigned \
int)sample_start_time.tv_usec/1000); \
}

/* SvrApl sampling sequence
* (1) MAC_SampleWork
* (2) MAC_SampleStartTime
* (3) Processing transaction on DB server
* (4) Except Delivery MAC_SampleDBSrvResp
* Only Delivery
MAC_SampleDBSrvRespDel
*/
#define \
MAC_SampleRespMax(max_resp_time, \
smp_max_resp_time) \
if (GLBSMP_shared_mem- \
>max_resp_time < el_time) \
GLBSMP_shared_mem- \
>max_resp_time = el_time; \
if (GLBSMP_shared_mem- \
>smp_max_resp_time < el_time) \
GLBSMP_shared_mem- \
>smp_max_resp_time = el_time;

/* For except Delivery */
#define MAC_SampleDBSrvResp(resp_time, \
max_resp_time, smp_max_resp_time, \
proc_trans) \
if (GLBSMP_shared_mem- \
>DataSampling == DATASAMPLE_ENABLE) { \
MAC_SampleExecuteTime; \
LockSem(GLBSMP_shared_mem); \
GLBSMP_shared_mem->resp_time \
+= el_time; \
MAC_SampleRespMax(max_resp_time, \
smp_max_resp_time); \
GLBSMP_shared_mem- \
>proc_trans++; \
UnlockSem(GLBSMP_shared_mem); \
}

```

```

    }

/* For only Delivery */
#define MAC_SampleDBSrvRespDel() \
    if (GLBSMP_shared_mem-
>DataSampling == DATASAMPLE_ENABLE) { \
    MAC_SampleExecuteTime; \
    LockSem(GLBSMP_semid); \
    GLBSMP_shared_mem-
>RspTimeDelivery += el_time; \

MAC_SampleRespMax(MaxRspTimeDelivery,
SMaxRspTimeDelivery); \
    GLBSMP_shared_mem-
>NumDelivery++; \
    GLBSMP_shared_mem-
>NumQueDelivery--; \
    UnlockSem(GLBSMP_semid); \
}

/* TpApl sampling sequence for except Delivery
* (1) MAC_SampleWork
* (2) MAC_SampleStartTime
* (3) MAC_SampleQueueUp
* (4) Processing transaction on TUXEDO and
DB server
* (5) Except Delivery MAC_SampleTuxResp
* Only Delivery MAC_SampleTuxRespDel
*/
/* For except Delivery */
#define MAC_SampleTuxResp(ans_time,
proc_trans, trans_que) \
    if (GLBSMP_shared_mem-
>DataSampling == DATASAMPLE_ENABLE) { \
    MAC_SampleExecuteTime; \
    LockSem(GLBSMP_semid); \
    GLBSMP_shared_mem->ans_time +=
el_time; \
    GLBSMP_shared_mem-
>proc_trans++; \
    GLBSMP_shared_mem->trans_que--;
\
    UnlockSem(GLBSMP_semid); \
}

/* For only Delivery */
#define MAC_SampleTuxRespDel \
    if (GLBSMP_shared_mem-
>DataSampling == DATASAMPLE_ENABLE) { \
    MAC_SampleExecuteTime; \
    LockSem(GLBSMP_semid); \
    GLBSMP_shared_mem->AnsDelivery
+= el_time; \
    GLBSMP_shared_mem-
>NumReqDelivery++; \
    UnlockSem(GLBSMP_semid); \
}

/*
* Output Self pafrotmance log
*/
#define MAC_SampleOutPutCsvLog \

ClientSampleSelfCsv(sample_end_time.tv_sec)

.....
tpapl/ThreadCntl.h
.....

/*****
*
* TPC-C Client Application Program Source
*
*****/

```

```

* Entry Functions
* Function definition for TUXEDO control
information.
*
* CREATE by TSL 2003.12.26
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003
*
*****
*****/

typedef struct _THREAD_CNTL_INFO {
    void* TrxDData;
    int TrxDDataLeng;
    char* QueryData;
    char* RespBuf;
} THREAD_CNTL_INFO;

.....
tpapl/TpAplDBDependPrototype.h
.....

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

* TPC-C Client Application Program Source
*
*
* Entry Functions
* Function definition for common.
*
* CREATE by TSL 2002.10.01
*
* GHANGE by TSL 2003.12.15 for COM+ -->
TUXEDO
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002
*
*****
*****/

int str2int(char *str, int field_len);
short str2short(char *str, int field_len);
int str2dbl(char *str, int field_len);

void int2str(char *str, int len, int num);
void int3str(char *str, int len, int num);
void dec2str(char *str, int len, double num);
void sigdec2str(char *str, int len, double num);
int str2str(char *str, int field_len);
void alp2str(char *str, int len, char *alp);
void date2str(char *str, char *time);
void zip2str(char *str, char *zip);
void phone2str(char *str, char *phone);

char* para_split(char *para, char delimita);
int checkHTMLform(char *str, char *buffer);
void convert_time(char *save_p, double
time);
void convert_date(char *save_p, double
time);

void time2str(char *str, char *time);

int set_errHTML(char *page, char *err_inf,
int cookie, char *errname);
/*
!int set_oraerr(char *page, char *err_inf, int
cookie);

```

```

*/
/* Replaced 03.01.15 */
#if 0
!int set_tuxerr(char *page, char *err_inf, int
cookie);
#endif
int set_SvrAplErr(char *page, char *err_inf,
int cookie);
/* Replaced end */
int set_errpage(char *buf, int user, int
err_no, int err_inf, int sub_inf2);

int NewOrder(char *s_buf,
RTE_INPUT_DATA *in_data, int cookie);
int Delivery(char *s_buf,
RTE_INPUT_DATA *in_data, int cookie);
int Payment(char *s_buf,
RTE_INPUT_DATA *in_data, int cookie);
int StockLevel(char *s_buf,
RTE_INPUT_DATA *in_data, int cookie);
int OrderStatus(char *s_buf,
RTE_INPUT_DATA *in_data, int cookie);
long GetGenericDataLen();

THREAD_CNTL_INFO* GetThreadCntl();
void FreeTuxBuffer(THREAD_CNTL_INFO*
ThreadCntlInfo);

.....
tpapl/TpAplHandler.c
.....

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

* TPC-C Client Application Program Source
*
*
* Entry Functions
* (1) TpAplHandler
* (2) OutputResultForm
* (3) GetConfigInfo
* (4) InitNewChildCreate
* (5) CreateTpAplSvrConf
*
* CREATE by TSL 2003.12.17
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003
*
*****
*****/
#include "forlinux.h"
#include <sys/types.h>
#include <unistd.h>
#include <atmi.h>

#include "stdio.h"
#include "httpd.h"
#include "http_config.h"
#include "http_protocol.h"
#include "ap_config.h"
#include "ap_compat.h"

#include "trans.h"
#include "ThreadCntl.h"
#include "GlobalArea.h"
#include "TpAplPrototype.h"
#include "log_level.h"
#include "log.h"
#include "menupage.h"
#include "sema.h"
#include "shmem.h"

```

```

#include "SampleInfo.h"

/*****
* TpApl HTTP processing handler
*
* Return Value
* OK      : Normal end
* DECLINED : Abnormal end
*
*****/
int TpAplHandler(request_rec *r)
{
    int cookie = -1;
    int rtn;
    char* S_BUF;

    RTE_INPUT_DATA in_data_area;
    THREAD_CNTL_INFO* ThreadCntlInfo;

    void OutputResultForm(request_rec *r, char*
buf_body);

    /* Check handler executing conditions */
    if (strcmp(r->handler, "tpapl") {
        return DECLINED;
    }

#ifdef PUT_INF_LOG
    TpccUserLog (LOG_INF, "#####
TpAplHandler start #####\n");
#endif

    if (r->header_only) {
        /* Request is header only */
        TpccUserLog (LOG_WRN, "Request is http
header only.\n");
        r->content_type = "text/html";
        goto OK_RETURN;
    }

    /* Initialize thread environment */
    #if 0 /* 2006.03.29 T.Motoo: Modified because
the argument had been changed. */
    ! ThreadCntlInfo = CreateThreadEnv();
    #endif
    ThreadCntlInfo = CreateThreadEnv(r-
>connection->id);

    if (ThreadCntlInfo == 0) {
        TpccUserLog (LOG_ERR, "Can't
Initialize\n");
        /* Initialization failure */
        OutputResultForm(r, initerr);
        goto OK_RETURN;
    }
    S_BUF = (char*)ThreadCntlInfo->RespBuf;

    /* Get Query string in to own area & analyze
requested data */
    #ifndef USEPOOL_QUERY
        strcpy(ThreadCntlInfo->QueryData, r->args);
    #else
        ThreadCntlInfo->QueryData = r->args;
    #endif

#ifdef PUT_INF_LOG
    TpccUserLog (LOG_INF, "Recieved request
[%dbytes][%s]\n",
                strlen(ThreadCntlInfo-
>QueryData), ThreadCntlInfo->QueryData);
#endif

    memset(&in_data_area, 0x00,
sizeof(in_data_area));
    cookie = anly_para ((char *)ThreadCntlInfo-
>QueryData, &in_data_area );

    /* Terminal Number Check
* If terminal number is not valid then send
error message.
*/
    if ( cookie < GLB_TermBase || cookie >=
(GLB_TermBase + GLB_Maxterm) ){

        if (ClientMonitor(cookie, S_BUF) == 0) {
            if (cookie != -3) /* -3:reuest od
performance sampling */
                TpccUserLog (LOG_INF, "Extended
function executing [function number:%d]\n",
cookie);
        }
        else {
            sprintf (S_BUF, badterm,
GLB_TermBase, GLB_TermBase +
GLB_Maxterm - 1, cookie);
            TpccUserLog (LOG_ERR, "Terminal
number over the range[Terminal number:%d]\n",
cookie);
        }

        OutputResultForm(r, S_BUF);
        goto OK_RETURN;
    }

    /* Execute the taransaction data */
    rtn = select_trn ( &in_data_area, S_BUF,
cookie );

    /* Response output form */
    OutputResultForm(r, S_BUF);

OK_RETURN:
#ifdef PUT_INF_LOG
    TpccUserLog (LOG_INF, "=====  

TpAplHandler end =====\n");
#endif
    return OK;
}

/*****
* Output Processing result form.
*
* Argument
* buf_body :
* Output message on screen
*
* Return Value
* NONE
*
*****/
void OutputResultForm(request_rec *r, char*
buf_body) {
    //int len=strlen(buf_body);

    r->content_type = "text/html";
    // ap_send_http_header(r);
    ap_rputs(buf_body, r);
    //buf_body[100]=0;
    //TpccUserLog (LOG_INF, "Content len=%d
data=(%s)\n", len,buf_body);
    return;
}

}

/*****
* Get configuration information
*
* Return Value
* char* NULL : allways
*
*****/
module tpapl_module;

char* GetConfigInfo(cmd_parms* parms, void*
mconfig, char* path) {
    char work_path[MAX_PATH];
    int i;
    char *conf;

    /* Set default log path */
    strcpy(GLB_TpAplLogPath,
DEFAULT_TPAPL_LOG_PATH);
    strcpy(GLB_LogFilePath,
DEFAULT_TPAPL_LOG_PATH);
    TpccUserLog (LOG_LCK, "Directive
processing start [GetConfigInfo]\n");

    /* Get configuration informaion (set to global
area) */
    strcpy(GLB_ConfigFilePath, path);
    GetConfigFile();

    /* Initialize TPAPL semaphore for log */
    strcpy(work_path, GLB_TpAplLogPath);
    for(i = strlen(work_path) - 1; i > 0 &&
work_path[i] != '/' ; i--);
    work_path[i] = '\0';

    if ((GLB_LogSemId = InitSem(work_path,
SEM_TPAPL_PROJID)) == -1) {
        TpccUserLog (LOG_LCK, "InitSem() faille for
TpApl log\n");
        return NULL;
    }

    /* Initialize SVRAPL semaphore for log */
    strcpy(work_path, GLB_SvrAplLogPath);
    for(i = strlen(work_path) - 1; i > 0 &&
work_path[i] != '/' ; i--);
    work_path[i] = '\0';

    if (InitSem(work_path,
SEM_SVRAPL_PROJID) == -1) {
        TpccUserLog (LOG_LCK, "InitSem() faille for
SvrApl log\n");
        return NULL;
    }

    /* Set server configuration */
    conf = (char*)ap_get_module_config(parms-
>server->module_config, &tpapl_module);
    strcpy(conf, path);

    /* Initialize client performance monitor */
    MAC_SampleInitPerformance;

    TpccUserLog (LOG_INF, "Directive processing
ended [GetConfigInfo]\n");
    return NULL;
}

/*****
* Initialize child process creates.
*
*****/

```

```
* Return Value
* NONE

void InitNewChildCreate(apr_pool_t* p,
server_rec* s) {

    TpcUserLog(LOG_INF, "Child creating
process start [InitNewChildCreate]\n");

    /* Get TSD key */
    GetThreadKey();

    /* Regist TUXEDO application */
    #if 0 /* 2006.03.29 T.Motoo: Modified because
the argument had been changed. */
    ! if ((GLB_TpContext = RegistTuxApl()) == 0) {
    #endif
        if ((GLB_TpContext = RegistTuxApl(p)) == 0) {
            TpcUserLog(LOG_ERR,
"RegistTuxApl() failed\n");
        }

        /* Regist cleanup entry */
        apr_pool_cleanup_register(p, NULL,
PlainCleanup, TermChildProcess);

        /* Initialize performance sampling */
        MAC_SampleInitChild;

        TpcUserLog(LOG_INF, "Child creating
process end [InitNewChildCreate]\n");

        return;
    }

}

/* Create server configuration
* Return Value
* Configuration area pointer

void* CreateTpAplSvrConf(apr_pool_t* p,
server_rec* s) {

    char* conf;

    /* Set default log path */
    strcpy(GLB_LogFilePath,
DEFAULT_TPAPL_LOG_PATH);

    TpcUserLog(LOG_LCK, "Create server
config start [CreateTpAplSvrConf]\n");
    if ((conf = (char*)ap_palloc(p, MAX_PATH))
== 0) {
        TpcUserLog(LOG_LCK, "Server config
area allocation failed\n");
        return (void*) conf;
    }
    *conf = '\0';
    TpcUserLog(LOG_LCK, "Create server
config ended [CreateTpAplSvrConf]\n");

    return (void*) conf;
}

tpapl/TpAplPrototype.h

*****
/* Header data */
#define h_del1 "\
<HTML><HEAD><TITLE>TPC-WINDOW
</TITLE></HEAD><BODY>\n\n\
<CENTER>Delivery<br></CENTER>\
<font size=4>\n<PRE>"

/* Screen data */
#define h_del2 "\
Warehouse: \n\n\
Carrier Number: \n\n\
Execution Status: \n\n\

*****
/* Tailer data */
#define h_del3 "\
</PRE><FORM ACTION=\"%s\
METHOD=\"%GET%\"\n\n\
<INPUT TYPE=\"%hidden\" NAME=\"%c\
VALUE=\"%d>\n\n\
<INPUT TYPE=\"%submit\" NAME=\"%b\
VALUE=\"%New order%\"\n\
<INPUT TYPE=\"%submit\" NAME=\"%b\
VALUE=\"%Payment%\"\n\
<INPUT TYPE=\"%submit\" NAME=\"%b\
VALUE=\"%Delivery%\"\n\
<INPUT TYPE=\"%submit\" NAME=\"%b\
VALUE=\"%Order Status%\"\n\
<INPUT TYPE=\"%submit\" NAME=\"%b\
VALUE=\"%Stock Level%\"\n\
<INPUT TYPE=\"%submit\" NAME=\"%b\
VALUE=\"%Quit%\"\n\
</FORM></BODY></HTML>\n"

/* Offset to field which should set data */
int delp[] = { 0xb, 0x23, 0x3d }; /* w_id, carrier
number, status */

*****
/* TPC-C Client Application Program Source
*
* Entry Functions
* Function definition for common.
*
* CREATE by TSL 2003.12.01
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 */

int anly_para (char *para,
RTE_INPUT_DATA *in_data);
int select_tm ( RTE_INPUT_DATA
*in_data, char *s_buf, int cookie);

int GetThreadKey();
#if 0 /* 2006.03.29 T.Motoo: The argument was
changed. */
!THREADCNTL_INFO* CreateThreadEnv();
#endif
THREADCNTL_INFO* CreateThreadEnv(int
id);
void FreeThreadKey();
void GetConfFileInfo();
#if 0 /* 2006.03.29 T.Motoo: The argument was
changed. */
!TPCONTEXT_T RegistTuxApl();
#endif
TPCONTEXT_T* RegistTuxApl(void *p);
apr_status_t TermChildProcess(void* p);
apr_status_t PlainCleanup(void* p);

tpapl/delpage.h

*****
/* COPYRIGHT FUJITSU LIMITED 2002
* CREATE:1998.02.23 FJH

*****

#define PUT_INF_LOG //
Information log
#define PUT_FNC_ENTRY_LOG //
Function entry point log
#define PUT_FNC_EXIT_LOG //
Function exit log

/* Function entry point log macro */
#ifdef PUT_FNC_ENTRY_LOG
#define MAC_PutFncEntryLog(func)
TpcUserLog(LOG_INF, ">>>>>func start
>>>>>");
#else
#define MAC_PutFncEntryLog(func) ;
#endif

/* Function exit point log */
#ifdef PUT_FNC_EXIT_LOG
#define MAC_PutFncExitLog(func)
TpcUserLog(LOG_INF, "<<<<<func end
<<<<<");
#else
#define MAC_PutFncExitLog(func) ;
#endif

tpapl/menupage.h
```

```

.....
/*****
*
* COPYRIGHT FUJITSU LIMITED 2002
* CREATE:1999.08.19 FJH
*
*****/

/* -----
-----
    menupage.h
----- */

/* Transaction select screen data */
#define h_menu "\
<HTML><HEAD><TITLE>TPC-  
WINDOW</TITLE></HEAD><BODY><CENTER>Transaction  
Menu<BR></CENTER><PRE></PRE></BODY></HTML>\n\n  
<FORM ACTION=\"%s\" METHOD=\"GET\">\n  
<INPUT TYPE=\"hidden\" NAME=\"c\"  
VALUE=%d>\n  
<INPUT TYPE=\"submit\" NAME=\"b1\"  
VALUE=\"New order!>\n  
<INPUT TYPE=\"submit\" NAME=\"b1\"  
VALUE=\"Payment!>\n  
<INPUT TYPE=\"submit\" NAME=\"b1\"  
VALUE=\"Delivery!>\n  
<INPUT TYPE=\"submit\" NAME=\"b1\"  
VALUE=\"Order Status!>\n  
<INPUT TYPE=\"submit\" NAME=\"b1\"  
VALUE=\"Stock Level!>\n  
<INPUT TYPE=\"submit\" NAME=\"b1\"  
VALUE=\"Quit!>\n</FORM></BODY></HTML>\n\n"

/* If client can not connect server then use this  
format */
#define noconnt "\
<HTML><HEAD><TITLE>ERROR: Can't  
Connect</TITLE></HEAD><BODY><PRE>\n  
<P>Sorry, all %d database connections are  
currently in use. <br>\n  
Please try again later. terminal number is %d.  
</P></BODY></HTML>\n\n"

/* If client can not connect server then use this  
format */
#define conntmax "\
<HTML><HEAD><TITLE>ERROR: Can't  
Connect</TITLE></HEAD><BODY><PRE>\n  
<P>The terminal number you entered isn't  
valid.<BR>\n  
It must be an integer in the range 1  
to %d.<P>\n\n  
The terminal number which you specified  
is %d.</P></BODY></HTML>\n\n"

/* If client can not connect server then use this  
format */
#define badterm "\
<HTML><HEAD><TITLE>ERROR: Can't  
Connect</TITLE></HEAD><BODY><PRE>\n  
<P>The terminal number you entered isn't  
valid.<BR>\n  
This Client process the terminal of %d  
to %d.<P>\n\n  
The terminal number which you specified  
is %d.</P></BODY></HTML>\n\n"

```

```

/* If connect client by WWW browser then use  
this format */
/* Terminal verification use only */
#define loginpage "\
<HTML><HEAD><TITLE>Welcome to TPC-  
C : %s</TITLE></HEAD><BODY><PRE>\n  
<P>Please identify your terminal number for this  
session.</P>\n\n  
<FORM ACTION=\"%s\" METHOD=\"GET\">\n\n  
<INPUT TYPE=\"hidden\" NAME=\"f1\"  
VALUE=\"%i\">\n\n  
Your Terminal Number:<INPUT NAME=\"c1\"  
SIZE=6><BR><HR>\n\n  
<INPUT TYPE=\"submit\">\n\n</FORM></BODY></HTML>\n\n"

/* LOGIN ERROR PAGE */
/* If terminal number not find then use this format  
*/
#define loginerr2 "\
<HTML><HEAD><TITLE>ERROR :Welcome to  
TPC-C</TITLE></HEAD><BODY><PRE>\n  
<P>Please identify your terminal number for this  
session.</P>\n\n  
<FORM ACTION=\"%s\" METHOD=\"GET\">\n\n  
<INPUT TYPE=\"hidden\" NAME=\"f1\"  
VALUE=\"%i\">\n\n  
Your Terminal Number:<INPUT NAME=\"c1\"  
SIZE=5><BR><HR>\n\n  
<INPUT TYPE=\"submit\">\n\n</FORM></BODY></HTML>\n\n"

/*
#define endpage "\
<HTML><HEAD><TITLE>Welcome to TPC-  
C</TITLE></HEAD><BODY><PRE>\n  
<P>Please identify your Warehouse and District  
for this session.</P>\n\n  
<FORM ACTION=\"%s\" METHOD=\"GET\">\n\n  
<INPUT TYPE=\"hidden\" NAME=\"c1\"  
VALUE=%d>\n\n  
<INPUT TYPE=\"hidden\" NAME=\"f1\"  
VALUE=\"%i\">\n\n  
Your Warehouse ID: <INPUT NAME=\"W1\"  
SIZE=4><BR>\n\n  
Your District ID: <INPUT NAME=\"d1\"  
SIZE=2><BR><HR>\n\n  
<INPUT TYPE=\"submit\">\n\n</FORM></BODY></HTML>\n\n"
*/

/* If client can not Initialize then use this format */
#define initerr "\
<HTML><HEAD><TITLE>ERROR: Can't  
Initialize</TITLE></HEAD><BODY><PRE>\n  
<P>Initialization was  
failed.</P></BODY></HTML>\n\n"

#define cntllr "\
<HTML><HEAD><TITLE>ERROR: Can't  
execute  
transaction</TITLE></HEAD><BODY><PRE>\n  
<P>Basic control was  
failed.</P></BODY></HTML>\n\n"

.....
tpapl/mod_tpapl.c
.....
/*****
****
*
* TPC-C Client Application Program Source
*
****

```

```

*
* Entry Functions
* (1) command_rec (table)
* (2) tpapl_register_hooks (function)
*
* (3) MODULE_DECLARE_DATA (table)
*
* CREATE by TSL 2003.12.17
*
* All Right Reserved, Copyright Co. FUJITSU  
LIMITED 2003 *
*****
****/

#include "httpd.h"
#include "http_config.h"
#include "http_protocol.h"
#include "ap_config.h"

/* Prototype for handlers */
int TpAplHandler(request_rec *r);
const char* GetConfigInfo(cmd_parms* parms,
void* mconfig, char* path);
void InitNewChildCreate(apr_pool_t* p,
server_rec* s);
void* CreateTpAplSvrConf(apr_pool_t* p,
server_rec* s);

static command_rec tpapl_cmds[] =
{
    {
        "TpAplConf", // Name of  
directive
        GetConfigInfo, // Directive  
handler
        NULL, // Offset
        OR_ALL, // Scope of  
directive
        TAKE1, // Form of  
argument of directive
        "full path of configuration file" //  
Description
    },
    {NULL}
};

static void tpapl_register_hooks(apr_pool_t* p)
{
    ap_hook_child_init(InitNewChildCreate, NULL,
NULL, APR_HOOK_MIDDLE);
    ap_hook_handler(TpAplHandler, NULL, NULL,
APR_HOOK_MIDDLE);
}

/* Dispatch list for API hooks */
module AP_MODULE_DECLARE_DATA
tpapl_module = {
    STANDARD20_MODULE_STUFF,
    NULL, // create per-dir config  
structures */
    NULL, // merge per-dir config  
structures */
    CreateTpAplSvrConf, // create per-server  
config structures */
    NULL, // merge per-server config  
structures */
    tpapl_cmds, // table of config file  
commands */
    tpapl_register_hooks // register hooks  
*/
};

```



```

Date: - - : : \r\n
\r\n
Warehouse:      District: \r\n
"

#define h_pay4 "\
\r\n
\r\n
Customer:   Cust-Warehouse:   Cust-
District: \r\n
Name:           Since: - -
\r\n"
/*
Credit: \r\n
%%Disc: . \r\n

*/

#define h_pay5 "\
- Phone: - - -
\r\n
\r\n
Amount Paid: $ . New Cust-Balance:
$ . \r\n
Credit Limit: $ . \r\n
\r\n
Cust-Data: "
/*
Cust-Data:      \r\n
\r\n
\r\n
\r\n"
*/

/* Trailer data */
#define h_pay3 "\
</PRE>\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n
<INPUT TYPE=\"hidden\" NAME=\"cl\"
VALUE=%d>\r\n
<INPUT TYPE=\"submit\" NAME=\"bl\"
VALUE=\"New order\">\
<INPUT TYPE=\"submit\" NAME=\"bl\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"bl\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"bl\"
VALUE=\"Order Status\">\
<INPUT TYPE=\"submit\" NAME=\"bl\"
VALUE=\"Stock Level\">\
<INPUT TYPE=\"submit\" NAME=\"bl\"
VALUE=\"Quit\">\
</FORM></BODY></HTML>"

/* Offset to field which should set data */
int payp[] = {
0x06,
0x29, 0x51,
0x55, 0x7e,
0x94, 0xbd,
0xd3, 0xe8, 0xeb, 0xf1, 0xfc, 0x111, 0x114,
0x11a,
0x12c, 0x142, 0x158,
0x164, 0x175, 0x178, 0x195, /* 18 - 21 */
0x1a8, 0x1d9, /* 22, 23 */
0x1e5, 0x216, /* 24, 25 */
0x225, 0x23a, 0x23d, 0x256,
0x284, 0x2a4,
0x2c5,
0x2e1, /* offset 0x3e */
0x320,
0x35f,
0x39e};

```

```

.....:
tpapl/stopage.h
.....:

/*****
*
* COPYRIGHT FUJITSU LIMITED 2002
* CREATE:1999.08.19 FJH
*
*****/

/* -----
-----
stopage.h
data of Stock Level transaction result screen
(HTML form)
----- */

/* Header data */
#define h_stock1 "\
<HTML><HEAD><TITLE>TPC-
WINDOW</TITLE></HEAD><BODY>\
<CENTER>Stock-Level<BR><CENTER>\
<font size=4>\r\n<PRE>"

/* Screen data*/
#define h_stock2 "\
Warehouse:  District: \r\n
\r\n
\r\n
Stock Level Threshold: \r\n
\r\n
low stock:  \r\n
\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n"

/* Trailer data */
#define h_stock3 "\
</PRE>\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n
<INPUT TYPE=\"hidden\" NAME=\"cl\"
VALUE=%d>\r\n
<INPUT TYPE=\"submit\" NAME=\"bl\"
VALUE=\"New order\">\
<INPUT TYPE=\"submit\" NAME=\"bl\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"bl\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"bl\"
VALUE=\"Order Status\">\
<INPUT TYPE=\"submit\" NAME=\"bl\"
VALUE=\"Stock Level\">\
<INPUT TYPE=\"submit\" NAME=\"bl\"
VALUE=\"Quit\">\
</FORM></BODY></HTML>\n"

/* Offset to field which should set data */
int stockp[] = {
0x0A, 0x1C,
0x39,
0x4a};

.....:
tpapl/tpapl.h
.....:

/*****
*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* struct definition.

```

```

*
* CREATE by TSL 2003.12.22
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
*****/

/* Http SO file path */
#define SOPATH "/tpapl"

/* HTML editing buffer size */
#define WORK_S 2400

/* Flags */
#define OK 1
#define NG 0

/* Make w_id d_id form terminal no. */
#define MAC_w_id(cookie) (cookie - 1)/10 + 1
#define MAC_d_id(cookie) (cookie - 1)%10 + 1

.....:
tpapl/tpaplFunction.c
.....:

/*****
*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* (1) only_para
* (2) select_trn
* (3) fast_menu
*
* CREATE by TSL 2002.10.01
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****
*****/

#include "forlinux.h"

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <pthread.h>
#include <sys/time.h>
#include <atmi.h>
#include "trans.h"
#include "log.h"

// HTML-Page Data
#include "tpcweb.h"
#include "tpcinweb.h"
#include "menupage.h"
#include "tpccinf.h"
#include "tpapl.h"
#include "ThreadCntl.h"
#include "GlobalArea.h"
#include "TpApIddbDependPrototype.h"
#include "sema.h"
#include "shmem.h"
#include "SampleInfo.h"

/*
only_para :
QueryString-----

```

```

-----
-----: -----
-----

Gets the query string and finds every
variable=value pair contained
within it. For every pair, it runs the variable
name through a really
big compound switch statement that matches
for specific variables we
want to catch. When we find a known variable
name, we stick a pointer
to its corresponding value into the appropriate
member of 'ptrs.'

query - a 1024 byte buffer that contains the
query string.
ptrs - a raw_form_data structure to hold
pointers.
*/
int only_para (char *para, RTE_INPUT_DATA
*in_data) {
    char *val, *rest;

    if(!para) return 0;
    if(*para == '0') return 0;

    while(para) {

        rest = para_split(para, '&'); /* next parameta
point */
        val = para_split(para, '='); /* now value
point */

        switch(para[0]) {
        case 'c':
            in_data->cookie = val;    break;

        case 'b':
            in_data->button = val;    break;

        case 'f':
            in_data->form = val;    break;

        case 't':
            in_data->threshold = val;    break;

        case 'D':
            in_data->D_ID = val;    break;

        case 'H':
            in_data->H_AMOUNT = val;    break;

        case 'C':
            switch(para[1]) {
            case 'I':
                in_data->C_ID = val;    break;

            case 'W':
                in_data->C_W_ID = val;    break;

            case 'L':
                in_data->C_LAST = val;    break;

            case 'D':
                in_data->C_D_ID = val;    break;
            }
            break;

        case 'O':
            switch(para[1]) {
            case 'C':
                in_data->O_CARRIER_ID = val;
break;

```

```

        case 'S':
            switch(para[2]) {
            case '0':
                if (para[3] >= 0x31 && para[3] <=
0x39){
                    //num = (int)(para[3] - 0x30);
                    if (strlen(val) != 0)
                        in_data-
>OL_SUPPLY_W_ID[(int)(para[3] - 0x30) - 1] =
val;
                }
                break;

            case '1':
                if (para[3] >= 0x30 && para[3] <=
0x35 ){
                    //num = (int)(para[3] - 0x30) + 10;
                    if (strlen(val) != 0)
                        in_data-
>OL_SUPPLY_W_ID[(int)(para[3] - 0x30) + 10 -
1] = val;
                }
                break;
            }
            break;

        case 'I':
            switch(para[2]) {
            case '0':
                if (para[3] >= 0x31 && para[3] <=
0x39){
                    //num = (int)(para[3] - 0x30);
                    if (strlen(val) != 0)
                        in_data->OL_ID[(int)(para[3] -
0x30) - 1] = val;
                }
                break;

            case '1':
                if (para[3] >= 0x30 && para[3] <=
0x35 ){
                    //num = (int)(para[3] - 0x30) + 10;
                    if (strlen(val) != 0)
                        in_data->OL_ID[(int)(para[3] -
0x30) + 10 - 1] = val;
                }
                break;
            }
            break;

        case 'Q':
            switch(para[2]) {
            case '0':
                if (para[3] >= 0x31 && para[3] <=
0x39 ){
                    //num = (int)(para[3] - 0x30);
                    if (strlen(val) != 0)
                        in_data-
>OL_QUANTITY[(int)(para[3] - 0x30) - 1] = val;
                }
                break;

            case '1':
                if (para[3] >= 0x30 && para[3] <=
0x35 ){
                    //num = (int)(para[3] - 0x30) + 10;
                    if (strlen(val) != 0)
                        in_data-
>OL_QUANTITY[(int)(para[3] - 0x30) + 10 - 1] =
val;
                }
                break;
            }
            break;

```

```

        }
        break;
    }

    para = rest;
}

if (in_data->cookie != 0)
    return(atoi (in_data->cookie));
else
    return(0);
}

/* -----
-
select_trn:
RTE-----
-----: -----
s_buf-----HTML-----
-----

interprets information from the user's input
data to determine which
page should be displayed back to the user.

query - the query string that comes back from
ParseFormData
ptrs - a pointer to a raw_form_data structure
with pointers
to values in 'query'.

-----
-----
*/
int select_trn ( RTE_INPUT_DATA *in_data,
char *s_buf, int cookie ) {

    int length = 0;
    int rtn = 0;

    MAC_SampleWork; /* Performance sampling
work area */

    if (in_data->form && (in_data->form[0] != 'M') )
    {

        if (in_data->form[0] == 'I'){
            /* send the transaction select screen
page */
            /* Replaced T,Kato 03.07,28 Speed up */
            /* rtn = fast_menu (s_buf, in_data,
cookie);*/
            sprintf(s_buf, h_menu, SOPATH,
cookie);
            /* Replaced end */
            return rtn;
        }
        else{

            MAC_SampleStartTime;

            /* check transaction type */
            switch(in_data->form[0]) {

                case 'N':

MAC_SampleQueueUp(NumQueNewOrder);
rtn = NewOrder (s_buf, in_data,
cookie);
                MAC_SampleTuxResp(AnsNewOrder,
NumReqNewOrder, NumQueNewOrder);
                break;

```

```

        case 'D':
MAC_SampleQueueUp(NumQueDelivery);
    rtn = Delivery(s_buf, in_data, cookie);
    MAC_SampleTuxRespDel;
    break;

        case 'P':
MAC_SampleQueueUp(NumQuePayment);
    rtn = Payment (s_buf, in_data, cookie);
    MAC_SampleTuxResp(AnsPayment,
NumReqPayment, NumQuePayment);
    break;

        case 'S':
MAC_SampleQueueUp(NumQueStockLevel);
    rtn = StockLevel(s_buf, in_data,
cookie);
    MAC_SampleTuxResp(AnsStockLevel,
NumReqStockLevel, NumQueStockLevel);
    break;

        case 'O':
MAC_SampleQueueUp(NumQueOrderStatus);
    rtn = OrderStatus (s_buf, in_data,
cookie);

MAC_SampleTuxResp(AnsOrderStatus,
NumReqOrderStatus, NumQueOrderStatus);
    break;

        default:
/* uninput transaction type */
    set_errpage(s_buf, cookie, 1, -4, 0, 0);
    rtn = 1;
    break;
    }
/* Output self performance log */
    MAC_SampleOutPutCsvLog;

    return rtn;
}
else if(in_data->button) {

/* send the data input screen page */
    switch(in_data->button[0]) {
        case 'N':
/*length = sprintf(s_buf, in_newpage,
SOPATH, cookie, srv->m_tcctxt[user_id].w_id);*/
        length = sprintf(s_buf, in_newpage,
SOPATH, cookie, MAC_w_id(cookie));
        strcpy(s_buf+length-1, in_newpage2);
        break;

        case 'D':
/*sprintf(s_buf, in_delpage, SOPATH,
cookie, srv->m_tcctxt[user_id].w_id);*/
        sprintf(s_buf, in_delpage, SOPATH,
cookie, MAC_w_id(cookie));
        break;

        case 'P':
/*sprintf(s_buf, in_paypage, SOPATH,
cookie, srv->m_tcctxt[user_id].w_id);*/
        sprintf(s_buf, in_paypage, SOPATH,
cookie, MAC_w_id(cookie));
        break;

        case 'S':
/*sprintf(s_buf, in_stkpage, SOPATH,
cookie,
        srv->m_tcctxt[user_id].w_id, srv-
>m_tcctxt[user_id].d_id);*/
        sprintf(s_buf, in_stkpage, SOPATH,
cookie, MAC_w_id(cookie), MAC_d_id(cookie));
        break;

        case 'O':
/*sprintf(s_buf, in_odrpage, SOPATH,
cookie, srv->m_tcctxt[user_id].w_id);*/
        sprintf(s_buf, in_odrpage, SOPATH,
cookie, MAC_w_id(cookie));
        break;

        case 'Q':
        sprintf (s_buf, loginpage , VLDATA,
SOPATH);
/* Replaced 03.01.15 Can't LeaveCriticalSection
*/
        #if 0
        !    return rtn;
        #endif
        break;
/* Replaced end */

        default:
/* uninput transaction type */
        set_errpage(s_buf, cookie, 0, -4, 0, 0);
        break;
    }
    return rtn;
}
else {

/* if there is not parameter then send login
page data.
    this part use WWW browser only */
    sprintf (s_buf, loginpage, VLDATA,
SOPATH);
    return 0;
}
}

/* Deleted T,Kato 03.07.28 Speed up */
#if 0
/*
! fast_menu:
! This function reads a user's responses to the
login form, sets
! up the user context, and returns the menu
page.
!*/
!
!int fast_menu ( char *s_buf, RTE_INPUT_DATA
*in_data, int cookie){
!
! //for warning
! in_data;
!
! sprintf(s_buf, h_menu, SOPATH, cookie);
! return 0;
!}
#endif

.....
tpapl/tpccinf.h
.....

/******
*
* COPYRIGHT FUJITSU LIMITED 2002
* CREATE:1999.08.19 FJH
*
*****/

/* -----
-----
tpcinweb.h
Transaction input data screen data
----- */

/* -----
-----
delivery page
* -----*/

#define in_delpage "\
<HTML><HEAD><TITLE>TPC-C:
Delivery</TITLE></HEAD>\r\n\
<BODY><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"D\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\"
VALUE=%d>\r\n\
<center>Delivery<br></center>\r\n\
<font size=4><PRE>Warehouse:%6d\r\n\
\r\n\

```



```
Customer: <INPUT NAME="CI" SIZE=4
maxlength=4> Cust-Warehouse: <INPUT
NAME="CW" SIZE=5 maxlength=6> Cust-
District: <INPUT NAME="CD" SIZE=2
maxlength=2>\r\n
Name: <INPUT NAME="CL"
SIZE=17 maxlength=16> Since:\r\n
Credit:\r\n
%%Disc:\r\n
Phone:\r\n
\r\n
Amount Paid $<INPUT NAME="H"
SIZE=7 maxlength=7> New Cust-
Balance:\r\n
Credit Limit:\r\n
\r\n
Cust-Data:\r\n
\r\n
\r\n
</PRE>\r\n
<INPUT
TYPE="submit"></FORM></BODY></HTML>"
```

```
/* -----
stock level page
* -----*/
#define in_stkpage "\
<HTML><HEAD><TITLE>TPC-C: Stock-
Level</TITLE></HEAD>\r\n
<BODY><FORM ACTION="%s"
METHOD="GET">\r\n
<INPUT TYPE="hidden" NAME="f"
VALUE="S">\r\n
<INPUT TYPE="hidden" NAME="c"
VALUE=%d>\r\n
<center>Stock-Level<br></center>\r\n
<font size=4><PRE>Warehouse:%6d
District: %2d\r\n
\r\n
Stock Level Threshold: <INPUT NAME="t"
SIZE=2 maxlength=2>\r\n
\r\n
low stock:\r\n
</PRE>\r\n
<INPUT
TYPE="submit"></FORM></BODY></HTML>"
```

```
#define in_stkpage2 "\
<HTML><HEAD><TITLE>TPC-C: Stock-
Level</TITLE></HEAD>\r\n
<BODY><FORM ACTION="%s"
METHOD="GET">\r\n
<INPUT TYPE="hidden" NAME="f"
VALUE="S">\r\n
<INPUT TYPE="hidden" NAME="c"
VALUE=%d>\r\n
<center>Stock Level<br></center>\r\n
<font size=3><PRE>\r\n
Warehouse:%6d District:%2d\r\n
\r\n
Stock Level Threshold:<INPUT NAME="t"
SIZE=2 maxlength=2>\r\n
\r\n
low stock:\r\n
</PRE>\r\n
<INPUT
TYPE="submit"></FORM></BODY></HTML>\r
n"
```

```
.....
tpapl/tpcweb.h
.....
/*-----
```

```
*
* COPYRIGHT FUJITSU LIMITED 2002
* CREATE:1998.08.06 FJH
*
*****/
/* -----
-----
tpcweb.h
-----*/
```

```
/* If transaction input data is abnormal then use
this format. */
#define errhtml "\
<HTML><HEAD><TITLE>ERROR: TPC-
C</TITLE></HEAD><BODY>\
<p>You did something bad. The error message
was:</p>\
<PRE>%s</PRE>\
<p>Either hit the "back" button on your browser
and fix the problem, \
or hit the "Quit" button below to terminate this
session. </P><HR>\
<P><FORM ACTION="%s"
METHOD="GET">\
<INPUT TYPE="hidden" NAME="c"
VALUE=%d>\
<INPUT TYPE="submit" NAME="b"
VALUE="Quit">\
</FORM></P></BODY></HTML>\r\n"
```

```
/* If TP application terminated abnormally then
use this format. */
#define tuxerr "\
<HTML><HEAD><TITLE>ERROR: Tuxedo
</TITLE></HEAD><BODY>\
<P>The database could not process your
request. \
tpcall terminated abnormally.</P>\
<HR><PRE>%s</PRE><HR>\
<FORM ACTION="%s" METHOD="GET">\
<INPUT TYPE="hidden" NAME="c"
VALUE=%d>\
<INPUT TYPE="submit" NAME="b"
VALUE="Quit">\
</BODY></HTML>"
```

```
/* If application terminated abnormally then use
this format. */
#define errorpage "\
<HTML><HEAD><TITLE>ERROR: %s
</TITLE></HEAD><BODY>\
<P>The database could not process your
request. \
Transaction terminated abnormally.</P>\
<HR><PRE>%s</PRE><HR>\
<FORM ACTION="%s" METHOD="GET">\
<INPUT TYPE="hidden" NAME="c"
VALUE=%d>\
<INPUT TYPE="submit" NAME="b"
VALUE="Quit">\
</BODY></HTML>"
```

```
#if 0 /* oraerr.symfoerr -> errorpage */
/* [oraerr]-[symfoerr]--"TITLE"-----
-----*/
/* Since "TITLE" was only different, [oraerr] and
[symfoerr] were changed so that it might be
common and could use.*/
/* If Oracle application terminated abnormally
then use this format. */
#define oraerr "\
!<HTML><HEAD><TITLE>ERROR: ORACLE
</TITLE></HEAD><BODY>\
```

```
<P>The database could not process your
request. \
!Transaction terminated abnormally.</P>\
!<HR><PRE>%s</PRE><HR>\
!<FORM ACTION="%s" METHOD="GET">\
!<INPUT TYPE="hidden" NAME="c"
VALUE=%d>\
!<INPUT TYPE="submit" NAME="b"
VALUE="Quit">\
!</BODY></HTML>"
!
/* If SymfoWare application terminated
abnormally then use this format. */
#define symfoerr "\
!<HTML><HEAD><TITLE>ERROR:
SYMFOWARE</TITLE></HEAD><BODY>\
!<P>The database could not process your
request. \
!Transaction terminated abnormally.</P>\
!<HR><PRE>%s</PRE><HR>\
!<FORM ACTION="%s" METHOD="GET">\
!<INPUT TYPE="hidden" NAME="c"
VALUE=%d>\
!<INPUT TYPE="submit" NAME="b"
VALUE="Quit">\
!</BODY></HTML>"
#endif
```

```
/* If TPINIT() abnormally then use this format. */
#define tuxerr "\
<HTML><HEAD><TITLE>ERROR: Tuxedo-init
</TITLE></HEAD><BODY>\
<P>The database could not process your
request. \
%s terminated abnormally.</P>\
</BODY></HTML>"
```

```
.....
tpapl/trans.h
.....
/*-----
*
* COPYRIGHT FUJITSU LIMITED 2002
* CREATE:1999.10.28 FJH
*
*****/
/*=====
=====+
FILENAME : trans.h
the work struct according to transaction is
declared.
+=====
=====*/
```

```
/* RTE - Client interface struct */
typedef struct {
char *button,
*cookie,
*form,
*O_CARRIER_ID,
*threshold,
*D_ID,
*C_ID,
*C_W_ID,
*C_D_ID,
*C_LAST,
*H_AMOUNT,
*OL_SUPPLY_W_ID[15],
*OL_ID[15],
```

```

*OL_QUANTITY[15];
} RTE_INPUT_DATA;
//) rte_input_data;

.....
tpapl/trnexe/ConvTime.c
.....

/*****
*
*          *
*   TPC-C Client Application Program Source
*
*          *
* Entry Functions
* (1) time2str
*
* CREATE by TSL 2002.10.01
*
*          *
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****
*****/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "forlinux.h"

#include "trans.h"
#include "ThreadCntl.h"
#include "TpAplDBDependPrototype.h"

/*
time2str:
Outputs a date and time in the supplied buffer
in the following format:
DD-MM-YYYY hh:mm:ss

field = the destination field
date = date and time to be converted and
displayed
*/
void time2str (char *str, char *time)
{
short mon;
int year, day, hour, min, sec;
char month[4];

#ifdef DBPRT
printf (test_fp, "time2: %s\n", time);
#endif
/* Modified by TSL -- BEGIN -- 2006.03.17 */
#if 0
! sscanf( time, "%2d-%3s-%2d.%2d:%2d:%2d",
! &day, month, &year, &hour, &min,
&sec );

! if(strcmp(month, "jan") == 0)
! strcpy(month, "01");
! if(strcmp(month, "feb") == 0)
! strcpy(month, "02");
! if(strcmp(month, "mar") == 0)
! strcpy(month, "03");
! if(strcmp(month, "apr") == 0)
! strcpy(month, "04");
! if(strcmp(month, "may") == 0)
! strcpy(month, "05");
! if(strcmp(month, "jun") == 0)
! strcpy(month, "06");
! if(strcmp(month, "jul") == 0)
! strcpy(month, "07");

```

```

! if(strcmp(month, "aug") == 0)
! strcpy(month, "08");
! if(strcmp(month, "sep") == 0)
! strcpy(month, "09");
! if(strcmp(month, "oct") == 0)
! strcpy(month, "10");
! if(strcmp(month, "nov") == 0)
! strcpy(month, "11");
! if(strcmp(month, "dec") == 0)
! strcpy(month, "12");
!
! int3str (str, 2, day);
! str[2] = '-';
!
! mon = atoi(month);
! int3str (&str[3], 2, mon);
! str[5] = '-';
!
! /* ----- */
! if ( year >= 70 )
! year += 1900;
! else
! year += 2000;
!
! int3str (&str[6], 4, year);
!
#endif
sscanf( time, "%2d-%2d-%4d.%2d:%2d:%2d",
&day, &mon, &year, &hour, &min, &sec );

int3str (str, 2, day);
str[2] = '-';
int3str (&str[3], 2, mon);
str[5] = '-';
int3str (&str[6], 4, year);

/* Modified by TSL -- END -- 2006.03.17 */

str[10] = ' ';

int3str (&str[11], 2, hour);
str[13] = ' ';

int3str (&str[14], 2, min);
str[16] = ' ';

int3str (&str[17], 2, sec);
}

.....
tpapl/trnexe/CreateTranErrReason.c
.....

/*****
*
*          *
*   TPC-C Client Application Program Source
*
*          *
* Entry Functions
* (1) CreateTranErrReason
*
* CREATE by TSL 2003.12.15
*
*          *
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
*****/
#include <stdio.h>
#include <stdlib.h>

```

```

#include <string.h>
#include "forlinux.h"

#include "atmi.h"
#include "tpcc.h"

int CreateTranErrReason(long errno_code, int
reason_code, char** reason_message) {
/* errno_code ..... return value of "tpcall" or
"tpacall"
* reason_code ..... xxxout.terror
* reason message ... convert message
*/
switch (errno_code) {

/* tpcall/tpacall error */
case -1:
TpccUserLog (LOG_ERR, "tpcall/tpacall
execution error occurred. [errno_code=%d]\n",
errno_code);
*reason_message = "Irrecoverable error in
tpcall/tpacall.";
return -2;
break;

/* Normal end */
default:
switch(reason_code) {
/* Normaol end */
case NOERR:
return 0;

/* Irrecoverable error */
case IRRECERR:
TpccUserLog (LOG_ERR, "Transaction
processing error [IRRECERR] occurred.\n");
*reason_message = "Irrecoverable error
in transaction processing.";
return -1; /* Execution error */

/* Retry */
default:
return 1;
}
}
}

.....
tpapl/trnexe/MakeShell
.....

#!/bin/sh
cd /home/tpc/client_apl/tpapl/trnexe
echo ""-----"" >
make_result.txt
echo ""----FOR WARE HOUSE BIND-----"" >>
make_result.txt
echo ""-----"" >>
make_result.txt
make BIND_TYPE="WH_BIND" >>
make_result.txt 2>&1
echo ""-----"" >>
make_result.txt
echo ""----FOR TRANSACTION BIND-----"" >>
make_result.txt
echo ""-----"" >>
make_result.txt
rm *.o >> make_result.txt
2>&1
make BIND_TYPE="TRNS_BIND" >>
make_result.txt 2>&1

```

```

.....
tpapl/trnexe/Makefile
.....

#-----
-----
# Makefile : Makefile for TpApl library on Linux.
#
# Created by TSL 2003.12.18
#
# All Right Rserverd, Copyright Co, FUJITSU
LIMITED 2003-2004.
#-----
-----

# GCC compile configurations
AR = ar
ARFLAGS = rv
CFLAGS = -Wall -O2
CC = gcc

# MACRO definition (input parameter)
# BIND_TYPE = TRNS_BIND ... Transaction
bind
# WH_BIND ..... Ware house bind
DMACRO = -D$(BIND_TYPE)

# home directory.
TOPDIR = /home/tpc/client_apl
TUXDIR = /usr/local/BEA/tuxedo8.1
APADIR = /usr/include/httpd
APLDIR = $(TOPDIR)/tpapl
SVRDIR = $(TOPDIR)/svrapl
ORADIR = /usr/local/oracle

# include directory
TPA_INC = -I$(APLDIR)/trnexe
COM_INC = -I$(TOPDIR)/common
TUX_INC = -I$(TUXDIR)/include
APA_INC = -I$(APADIR)
APL_INC = -I$(APLDIR)
SVR_INC = -I$(SVRDIR)
ORA_INC = -I$(ORADIR)/rdbms/demo -
-I$(ORADIR)/rdbms/public

# header file directory
HDFDIR = $(APLDIR)/trnexe
COMDIR = $(TOPDIR)/common

INCLUDE = $(TPA_INC) $(COM_INC)
$(APA_INC) $(TUX_INC) $(APL_INC)
$(SVR_INC) $(ORA_INC)
INCFILE = $(SVRDIR)/tpcc_info.h \
$(HDFDIR)/OracleInfo.h \
$(HDFDIR)/OracleFunction.h \
$(HDFDIR)/log_level.h \
$(APLDIR)/GlobalArea.h \
$(APLDIR)/trans.h \
$(APLDIR)/tpcweb.h \
$(APLDIR)/TpAplDBDependPrototype.h \
$(APLDIR)/tpapl.h \
$(APLDIR)/ThreadCntl.h \
$(APLDIR)/stpage.h \
$(APLDIR)/paypage.h \
$(APLDIR)/odrpage.h \
$(APLDIR)/newpage.h \
$(APLDIR)/delpage.h \
$(COMDIR)/log.h \
$(COMDIR)/forlinux.h \

# target object
OBS = ConvTime.o CreateTranErrReason.o
TestFunction.o TransactionDataLen.o \

```

```

TrxDelivery.o TrxNewOrder.o
TrxOrderStatus.o TrxPayment.o TrxStockLevel.o
ARCH_LIB =
$(APLDIR)/trnexe/libtrnexe_$(BIND_TYPE).a

$(ARCH_LIB) : $(OBS)
$(AR) $(ARFLAGS) $(ARCH_LIB) $(OBS)

.SUFFIXES: .o .c
.c.o:
$(CC) -o $@ -c $(CFLAGS) $(INCLUDE)
$(DMACRO) $<

$(OBS) : $(INCFILE)

clean:
# rm $(ARCH_LIB) $(OBS)

.....
.....
tpapl/trnexe/OracleFunction.h
.....

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

*
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Function definition for Oracle.
*
* CREATE by TSL 2002.10.01
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *

*****
*****/

// -----
// TrxNewOrder.cpp
// -----
int chk_NOdata (NewOrderData *bp, int cnt,
RTE_INPUT_DATA *in_data, int svcnt);
int setNOdata (char *s_work, int OF, int cnt,
NewOrderData *bp, RTE_INPUT_DATA
*in_data);

// -----
// TestProc.cpp
// -----
void dummy_delivery ( DeliveryData *bp );
void dummy_stocklvl ( StockLevelData *bp );
void dummy_payment ( PaymentData *bp );
void dummy_orderstat ( OrderStatusData *bp );
void dummy_neworder ( NewOrderData *bp );
void oder_dsp (RTE_INPUT_DATA *in_data,
OrderStatusData *bp, int w_id, int d_flag);
void pay_dsp (RTE_INPUT_DATA *in_data,
PaymentData *bp, int w_id, int d_flag);
void sto_dsp (RTE_INPUT_DATA *in_data,
StockLevelData *bp, int w_id, int d_id, int
d_flag);
void new_dsp (RTE_INPUT_DATA *in_data,
NewOrderData *bp, int w_id, int d_flag, int cnt);

int CreateTranErrReason (long errno_code, int
reason_code, char** reason_message);

// -----Oracle--Symfo-----
// used in common by Oracle and Symfo.

```

```

#define MAC_errHTML(page, err_inf, cookie )
set_errHTML(page, err_inf, cookie, "ORACLE" );
#define MAC_errHTML_TUXEDO(page, err_inf,
cookie ) set_errHTML(page, err_inf, cookie,
"TUXEDO" );

.....
.....
tpapl/trnexe/OracleInfo.h
.....

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

*
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Oracle Area definition.
*
* CREATE by TSL 2002.10.01
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *

*****
*****/

#ifndef ORACLEINFO_H
#define ORACLEINFO_H

#define INTNULL 0

#endif

.....
.....
tpapl/trnexe/TestFunction.c
.....

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

*
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* (1) get_datetimestr
* (2) get_datestr
* (3) dummy_delivery
* (4) dummy_stocklvl
* (5) dummy_payment
* (6) dummy_orderstat
* (7) dummy_neworder
* (8) oder_dsp
* (9) pay_dsp
* (10) sto_dsp
* (11) new_dsp
* (12) tsp
*
* CREATE by TSL 2002.10.01
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *

*****
*****/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "forlinux.h"

```

```

#include "trans.h"
#include "tpcc_info.h"

//
// dmy.h : DLL-----
//
// ----- --SCRTEST--#define-----
// -----DBPRT--#define-----
//
//
// -----
//
#endif SCRTEST

char *get_dattimestr( char *buf )
{
    struct tm    *tm;
    time_t      tim;

    time( &tim );
    tm = localtime( &tim );

    sprintf( buf, "%2d-%2d-%4d %2d:%2d:%2d",
tm->tm_mday, tm->tm_mon+1,
    tm->tm_year+1900, tm->tm_hour, tm-
>tm_min, tm->tm_sec );

    return buf;
}

char *get_datestr( char *buf )
{
    struct tm    *tm;
    time_t      tim;

    time( &tim );
    tm = localtime( &tim );

    sprintf( buf, "%2d-%2d-%4d",
tm->tm_mday, tm->tm_mon+1, tm-
>tm_year+1900 );
    return buf;
}

void dummy_delivery( DeliveryData *bp )
{
    bp->delout.terror = NOERR;

    return;
}

void dummy_stocklvl( StockLevelData *bp )
{
    int i;

    bp->stoout.terror = NOERR;

    do{
        i = rand()%1000;
    } while ( i > bp->stoin.threshold );

    bp->stoout.low_stock = i;

    return;
}

```

```

void dummy_payment( PaymentData *bp )
{
    bp->payout.terror = NOERR;
    strcpy( bp->payout.h_date, "11-oct-
02.16:37:15" );
    strcpy( bp->payout.w_street_1, "Baker
street" );
    strcpy( bp->payout.w_street_2, "221B" );
    strcpy( bp->payout.w_city, "London" );
    strcpy( bp->payout.w_state, "GB" );
    strcpy( bp->payout.w_zip, "88033000" );

    strcpy( bp->payout.d_street_1, "Minato-ku" );
    strcpy( bp->payout.d_street_2, "Azabu 10" );
    strcpy( bp->payout.d_city, "Tokyo" );
    strcpy( bp->payout.d_state, "JP" );
    strcpy( bp->payout.d_zip, "102 1234" );

    bp->payout.c_id = 777;
    strcpy( bp->payout.c_first, "John" );
    strcpy( bp->payout.c_middle, "H" );
    strcpy( bp->payout.c_last, "Watson" );
    strcpy( bp->payout.c_street_1, "Baker
street" );
    strcpy( bp->payout.c_street_2, "221B" );
    strcpy( bp->payout.c_credit, "GC" );
    bp->payout.c_discount = (float)20.00;
// check
    strcpy( bp->payout.c_city, "London" );
    strcpy( bp->payout.c_state, "GB" );
    strcpy( bp->payout.c_zip, "888 1234" );
    strcpy( bp->payout.c_phone,
"12345678901234567" );
    bp->payout.c_balance = 67876;
    bp->payout.c_credit_lim = 77777;
    strcpy( bp->payout.c_since, "11-10-2002" );

    strcpy( bp->payout.c_data,
"Migyamigyamigyamigyamigyamigya"
    "migyamigyamigyamigyamigya" );
    return;
}

void dummy_orderstat( OrderStatusData *bp )
{
    int i, j;

    bp->ordout.terror = NOERR;
    bp->ordout.c_id = rand()%10000;
    strcpy( bp->ordout.c_first, "Robert" );
    strcpy( bp->ordout.c_middle, "L" );
    strcpy( bp->ordout.c_last, "Fish" );
    bp->ordout.c_balance =
( ( rand()*rand()%19999999 )-9999999 ) /
(double)100.0;

    bp->ordout.o_id = rand()%10000;
    strcpy( bp->ordout.o_entry_d, "11-oct-
02.16:25:45" );
    bp->ordout.o_carrier_id = rand()%100;

    bp->ordout.o_ol_cnt = ( rand()%11 )+5;
    j = bp->ordout.o_ol_cnt;
    for ( i = 0; i < j; i++ )
    {
        bp->ordout.ol_supply_w_id[i] =
( rand()%100000 )+1;
        bp->ordout.ol_i_id[i] = ( rand()%100000 )+1;
        bp->ordout.ol_quantity[i] = ( rand()%99 )+1;
        bp->ordout.ol_amount[i] = (float)rand();
// check

        sprintf( bp->ordout.ol_delivery_d[i], "%02d-10-
2002",i + 1 );

```

```

    }

    return;
}

void dummy_neworder( NewOrderData *bp )
{
    static int o_id = 3001;
    int i;

    bp->newout.terror = NOERR;

    strcpy( bp->newout.c_last, "Holmes" );
    strcpy( bp->newout.c_credit, "GC" );
    bp->newout.o_id = o_id++;

    strcpy( bp->newout.o_entry_d, "11-oct-
02.15:10:30" );
    bp->newout.c_discount =
(float)(( rand()%101 )/10000.0); // check
    bp->newout.w_tax =
(float)(( rand()%2001 )/10000.0); // check
    bp->newout.d_tax =
(float)(( rand()%2001 )/10000.0); // check
    bp->newout.total_amount = 0; //
check

    for ( i = 0; i < 15; i++ ){
        if ( bp->newin.ol_supply_w_id[i] == 0 ) {
            break;
        }
        if ( bp->newin.ol_i_id[i] == -1 ) {
        }

        sprintf(bp-
>newout.i_name[i],"ItemName%02d",i);
        bp->newout.s_quantity[i] = ( rand()%10 )+1;
        bp->newout.brand_generic[i] =
( rand()%26 )+'A';
        bp->newout.i_price[i] =
(float)(( ( rand()%10000 )+1 )/100.0); // check
        bp->newout.ol_amount[i]
        = bp->newout.i_price[i] * bp-
>newin.ol_quantity[i]; // check
        bp->newout.total_amount += bp-
>newout.ol_amount[i]; // check
    }
    bp->newout.o_ol_cnt = i;

    return;
}

#endif

//
// -----
//
#endif DBPRT
void oder_dsp(RTE_INPUT_DATA *in_data,
    OrderStatusData *bp, int w_id, int
d_flag)
{
    int i;

    if ( d_flag == 0 ){
        fprintf ( test_fp, "----- in data area -----\n\n" );
        fprintf ( test_fp, "w_id = %d ", w_id );
        fprintf ( test_fp, "d_id = %s ", in_data->D_ID );

        if ( in_data->C_ID != 0 )
            fprintf ( test_fp, "c_id = %s \n", in_data-
>C_ID );
        if ( in_data->C_LAST != 0 )

```



```

    fprintf(test_fp, "c_last = %s\n", in_data-
>C_LAST);

    fprintf(test_fp, "----- trans buf area -----
\n\n");
    fprintf(test_fp, "w_id = %d ", bp->w_id);
    fprintf(test_fp, "d_id = %d ", bp->d_id);
    fprintf(test_fp, "c_id = %d\n", bp->c_id);
}
else {
    fprintf(test_fp, "----- trans buf area (after) ----
\n\n");
    fprintf(test_fp, "w_id = %d ", bp->w_id);
    fprintf(test_fp, "d_id = %d ", bp->d_id);
    fprintf(test_fp, "c_id = %d\n", bp->c_id);
    fprintf(test_fp, "c_first=%s ", bp->c_first);
    fprintf(test_fp, "c_middl=%s ", bp-
>c_middle);
    fprintf(test_fp, "c_last =%s\n", bp->c_last);

    fprintf(test_fp, "c_balan=%f ", bp-
>c_balance);
    fprintf(test_fp, "o_id =%d ", bp->o_id);
    fprintf(test_fp, "o_entry_d=%s\n", bp-
>o_entry_d); // check

    if ( bp->o_carrier_id != 0 ) {
        fprintf(test_fp, "o_carrier_id=%d\n", bp-
>o_carrier_id);
    }

    for( i = 0; i < bp->o_ol_cnt; i++ ){
        fprintf(test_fp, "ol_supp=%d ", bp-
>ol_supply_w_id[i]);
        fprintf(test_fp, "ol_i_id=%d ", bp-
>ol_i_id[i]);
        fprintf(test_fp, "ol_quan=%d ", bp-
>ol_quantity[i]);
        fprintf(test_fp, "ol_amou=%f\n", bp-
>ol_amount[i]);
    }
}

void pay_dsp(RTE_INPUT_DATA *in_data,
PaymentData *bp, int w_id, int d_flag)
{
    int i;

    if (d_flag == 0){
        fprintf(test_fp, "----- in data area -----
\n\n");

        fprintf(test_fp, "w_id = %d ", w_id);
        fprintf(test_fp, "d_id = %s ", in_data->D_ID);
        fprintf(test_fp, "c_w_id=%s ", in_data-
>C_W_ID);
        fprintf(test_fp, "c_d_id=%s ", in_data-
>C_D_ID);
        fprintf(test_fp, "h_amount=%s\n", in_data-
>H_AMOUNT);

        if (in_data->C_ID != 0)
            fprintf(test_fp, "c_id = %s\n", in_data-
>C_ID);
        if (in_data->C_LAST != 0)
            fprintf(test_fp, "c_last = %s\n", in_data-
>C_LAST);

        fprintf(test_fp, "----- trans buf area -----
\n\n");
        fprintf(test_fp, "w_id = %d ", bp->w_id);
        fprintf(test_fp, "d_id = %d ", bp->d_id);
        fprintf(test_fp, "c_id = %d ", bp->c_id);
        fprintf(test_fp, "c_w_id=%d ", bp->c_w_id);

```

```

    fprintf(test_fp, "c_d_id=%d ", bp->c_d_id);
    fprintf(test_fp, "h_amount=%f\n", bp-
>h_amount);
}
else {
    fprintf(test_fp, "----- trans buf area (after) ---
\n\n");
    fprintf(test_fp, "w_id = %d ", bp->w_id);
    fprintf(test_fp, "d_id = %d ", bp->d_id);
    fprintf(test_fp, "c_id = %d\n", bp->c_id);

    fprintf(test_fp, "w_str_1=%s ", bp-
>w_street_1);
    fprintf(test_fp, "w_str_2=%s\n", bp-
>w_street_2);
    fprintf(test_fp, "d_str_1=%s ", bp-
>d_street_1);
    fprintf(test_fp, "d_str_2=%s\n", bp-
>d_street_2);
    fprintf(test_fp, "w_city=%s ", bp->w_city);
    fprintf(test_fp, "w_state=%s\n", bp->w_state);
    fprintf(test_fp, "d_city=%s ", bp->d_city);
    fprintf(test_fp, "d_state=%s\n", bp->d_state);

    fprintf(test_fp, "c_w_id=%d ", bp->c_w_id);
    fprintf(test_fp, "d_w_id=%d\n", bp->c_d_id);

    fprintf(test_fp, "c_first=%s ", bp->c_first);
    fprintf(test_fp, "c_middl=%s ", bp-
>c_middle);
    fprintf(test_fp, "c_last =%s\n", bp->c_last);

    fprintf(test_fp, "c_str_1=%s ", bp-
>c_street_1);
    fprintf(test_fp, "c_str_2=%s\n", bp-
>c_street_2);
    fprintf(test_fp, "c_city=%s\n", bp->c_city);
    fprintf(test_fp, "c_credi=%s ", bp->c_credit);
    fprintf(test_fp, "c_state=%s\n", bp->c_state);

    fprintf(test_fp, "c_balan=%f\n", bp-
>c_balance);

    i = strlen( bp->c_data );
    fprintf(test_fp, "c_date=%s\n", bp->c_data);
}

void sto_dsp(RTE_INPUT_DATA *in_data,
StockLevelData *bp, int w_id, int d_id, int
d_flag)
{
    if (d_flag == 0){
        fprintf(test_fp, "----- in data area -----
\n\n");

        fprintf(test_fp, "w_id = %d ", w_id);
        fprintf(test_fp, "d_id = %d ", d_id);
        fprintf(test_fp, "threshold= %s\n", in_data-
>threshold);

        fprintf(test_fp, "----- trans buf area -----
\n\n");
        fprintf(test_fp, "w_id = %d ", bp->w_id);
        fprintf(test_fp, "d_id = %d ", bp->d_id);
        fprintf(test_fp, "threshold= %d\n", bp-
>threshold);
    }
    else{
        fprintf(test_fp, "----- trans buf area (after) ---
\n\n");

        fprintf(test_fp, "w_id = %d ", bp->w_id);
        fprintf(test_fp, "d_id = %d ", bp->d_id);

```

```

    fprintf(test_fp, "threshold= %d ", bp-
>threshold);
    fprintf(test_fp, "low_stock= %d\n", bp-
>low_stock);
}
}

void new_dsp(RTE_INPUT_DATA *in_data,
NewOrderData *bp, int w_id, int d_flag,
int cnt)
{
    int i, loop;

    if (d_flag == 0){
        fprintf(test_fp, "----- in data area -----
\n\n");

        fprintf(test_fp, "w_id = %d ", w_id);
        fprintf(test_fp, "d_id = %s ", in_data->D_ID);
        fprintf(test_fp, "c_id = %s\n", in_data-
>C_ID);

        for (i = 0; i < cnt; i++){

            if (in_data->OL_SUPPLY_W_ID[i] != 0 ){
                fprintf(test_fp, "ol_sup_w_id=%s
", in_data->OL_SUPPLY_W_ID[i]);
            }

            if (in_data->OL_I_ID[i] != 0 ){
                fprintf(test_fp, "ol_i_id=%s ", in_data-
>OL_I_ID[i]);
            }

            if (in_data->OL_QUANTITY[i] != 0 ){
                fprintf(test_fp, "ol_quan=%s\n",
in_data->OL_QUANTITY[i]);
            }
        }

        fprintf(test_fp, "----- trans buf area -----
\n\n");
        fprintf(test_fp, "w_id = %d ", bp->w_id);
        fprintf(test_fp, "d_id = %d ", bp->d_id);
        fprintf(test_fp, "c_id = %d\n", bp->c_id);

        for (i = 0; i <= cnt; i++){

            fprintf(test_fp, "ol_sup_w_id=%d ", bp-
>ol_supply_w_id[i]);
            fprintf(test_fp, "ol_i_id=%d ", bp-
>ol_i_id[i]);
            fprintf(test_fp, "ol_quan=%d\n", bp-
>ol_quantity[i]);
        }
    }
    else{
        fprintf(test_fp, "----- trans buf area (after) ---
\n\n");

        fprintf(test_fp, "c_last=%s ", bp->c_last);
        fprintf(test_fp, "c_credit=%s\n", bp-
>c_credit);
        fprintf(test_fp, "o_id=%d ", bp->o_id);

        fprintf(test_fp, "o_entry_d=%s\n", bp-
>o_entry_d); // check
        fprintf(test_fp, "c_discnt=%f\n", bp-
>c_discount * 100.0);

        fprintf(test_fp, "o_ol_cnt=%d ", bp-
>o_ol_cnt);

        fprintf(test_fp, "w_tax=%f ", bp->w_tax *
100.0);

```

```

    fprintf (test_fp, "d_tax=%f\n", bp->d_tax *
100.0);

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

        fprintf(test_fp, "-----
no_sup_w_id=%d ",
            bp->ol_supply_w_id[i]);
        fprintf(test_fp, "o_i_id=%d ", bp-
>o_i_id[i]);
        fprintf(test_fp, "i_name=%s\n", &bp-
>i_name[i][0]);
        fprintf(test_fp, "o_quant=%d ", bp-
>o_ol_quantity[i]);
        fprintf(test_fp, "s_quant=%d ", bp-
>s_quantity[i]);
        fprintf(test_fp, "brand=%c ", bp-
>brand_generic[i]);
        fprintf(test_fp, "i_price=%f ", bp-
>i_price[i]); // check
        fprintf(test_fp, "ol_amnt=%f\n", bp-
>o_ol_amount[i]); // check
    }
    fprintf (test_fp, "total_a=%f\n", bp-
>total_amount); // check
}

#endif

#ifdef TIMEST
int tsp(int id, char flag, char type){

// struct tm times;
SYSTEMTIME systemTime; // for IIS Version

GetLocalTime(&systemTime);

fprintf (TIMES, "ID=%d, FL=%d,
T=%c : %d:%d:%d.%d\n",
    id, flag, type, (int)systemTime.wHour,
    (int)systemTime.wMinute,
    (int)systemTime.wSecond,
    (long)systemTime.wMilliseconds);

fflush (TIMES);
return 0; }

#endif

.....
tpapl/trnexe/TransactionDataLen.c
.....

/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* (1) GetGenericDataLen
* (2) GetDeliveryDataLen
*
* CREATE by TSL 2002.10.01
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****
****/

```

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "forlinux.h"

#include "tpcc_info.h"

/*****
****
* Get transaction data size.
* Return Value
* transaction data size
*****
****/
long GetGenericDataLen() {
    long max_len = 0;

    if (max_len < sizeof(NewOrderData)) max_len
= sizeof(NewOrderData);
    if (max_len < sizeof(OrderStatusData))
max_len = sizeof(OrderStatusData);
    if (max_len < sizeof(PaymentData)) max_len
= sizeof(PaymentData);
    if (max_len < sizeof(StockLevelData))
max_len = sizeof(StockLevelData);
    if (max_len < sizeof(DeliveryData)) max_len =
sizeof(DeliveryData);

    return max_len;
}

/*****
****
* Get delivery transaction data size.
*
* Return Value
* Delivery transaction data size
*****
****/
long GetDeliveryDataLen() {
    return sizeof(struct delstruct);
}

.....
tpapl/trnexe/TrxDelivery.c
.....

/*****
****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* (1) Delivery
*
* CREATE by TSL 2003.12.15
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
****/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "forlinux.h"

#include <sys/time.h>

```

```

#include "atmi.h"

#include "trans.h"
#include "tpcc_info.h"
#include "delpage.h"

#include "ThreadCntl.h"
#include "TpAplDBDependPrototype.h"
#include "log.h"
#include "log_level.h"

#include "tpapl.h"
// #include "GlobalArea.h" // Common
#include "OracleFunction.h"

/*****
-----
Delivery : this function processes the delivery
transaction.
----- */
int Delivery (char *s_buf, RTE_INPUT_DATA
"in_data, int cookie)
{
    DeliveryData *bp;
    char S_WORK[WORK_S];

    struct timeval tv;

#ifdef TRNS_BIND
static char* svr_name = "DELIVERY";
#else
static char* svr_name = "OPSTUXSERVER";
#endif

    int h_del1_len;
    int h_del2_len;
    int h_del3_len;

    THREAD_CNTL_INFO* ThreadCntlInfo;

    //SvrAPL return value
#ifdef SCRTST
int ret_val;
#endif

    MAC_PutFncEntryLog("Delivery");

    /* Create execution environment */
    ThreadCntlInfo = GetThreadCntl();
    if (ThreadCntlInfo == 0) {
        sprintf(S_WORK, "thread control
information is not allocated [DEL]\n");
        MAC_errHTML(s_buf, S_WORK, cookie);
        TpccUserLog (LOG_ERR, S_WORK);
        return (-1);
    }
    bp = (DeliveryData*)ThreadCntlInfo->TrxDData;
    memset(bp, 0x00, sizeof(DeliveryData));

    /* ----- Check
the Input data */
    bp->delin.w_id = MAC_w_id(cookie);

    bp->delin.o_carrier_id = str2short (in_data-
>O_CARRIER_ID, 2);

    if (bp->delin.o_carrier_id < 1 || bp-
>delin.o_carrier_id > 10) {
        TpccUserLog (LOG_ERR, "Input data error
[DEL] (o_carrier_id = %s)[Return_Value:%d]\n",
            in_data->O_CARRIER_ID, bp-
>delin.o_carrier_id);
    }
}

```

```

    return set_errpage(s_buf, cookie, 5, (int)bp-
>delin.o_carrier_id, 0, 0);
}

/* ----- Execute
Delivery transaction */

/* Get Derivery start time */
gettimeofday(&tv, NULL);
bp->delin.startsec = (long)tv.tv_sec;
bp->delin.startusec = (long)(tv.tv_usec /
1000);

#ifdef SCRTST

/* Replaced 2003.12.15 Transaction processeing
interface COM+ --> TUXEDO */
#ifdef TRNS_BIND
/* Set transaction type for Warehouse bind */
bp->retval = 4;
#endif

resend_delivery:
    ret_val = tpacall(svr_name, (char*)bp,
sizeof(NewOrderData), 0 | TPNOTIME |
TPNOREPLY);
    if (ret_val == -1) {
        /* Display messege */
        switch ( tpermno ) {
            case TPELIMIT: /* -----
----- */
            case TPETIME: /* -----
----- */
            case TPGOTSIG: /* -----
----- */
                /* Because it is an executable again error,
processing is executed again. */
                TpccUserLog (LOG_WRN, "Delivery
retry reason by termno=%d\n", tpermno);
                goto resend_delivery;
                break;

            default:
                /* The error which was not able to be
executed again occurred */
                sprintf( S_WORK, "tpacall failed in
Delivery: tpermno = %d\n"
                " svc = '%s' carrier = %d\n", tpermno,
                svr_name, bp->delin.o_carrier_id );

                MAC_errHTML_TUXEDO( s_buf,
S_WORK, cookie );
                TpccUserLog (LOG_ERR, S_WORK);
                FreeTuxBuffer(ThreadCntlInfo);
                return (-1);
            }
        }
    }
#else
    dummy_delivery(bp);
#endif

/* ----- The execution result data notified RTE
is made by the HTML form */
/* Replaced T.kato 03.04.18 Speed up */
//sprintf (S_WORK, h_del2);
strcpy(S_WORK, h_del2);
h_del2_leng = strlen(S_WORK);
/* Replaced end */

int2str ((S_WORK + delp[0]), 6, (int)bp-
>delin.w_id);

int2str ((S_WORK + delp[1]), 2, (int)bp-
>delin.o_carrier_id);
alp2str ((S_WORK + delp[2]), 25, "Delivery
has been queued");

```

```

/* Replaced T.Kato 03.04.18 */
#if 0
! sprintf(s_buf, h_del1);
! strcat (s_buf, S_WORK);
!
! sprintf(S_WORK, h_del3, SOPATH, cookie);
! strcat (s_buf, S_WORK);
#endif
strcpy(s_buf, h_del1);
h_del1_leng = strlen(s_buf);
memcpy(s_buf + h_del1_leng, S_WORK,
h_del2_leng);
h_del3_leng = sprintf(S_WORK, h_del3,
SOPATH, cookie);
memcpy(s_buf + h_del1_leng + h_del2_leng,
S_WORK, h_del3_leng);
*(s_buf + h_del1_leng + h_del2_leng +
h_del3_leng) = '\0';
/* Replaced end */

FreeTuxBuffer(ThreadCntlInfo);
return 0;
}

.....:
tpapl/trnexe/TrxNewOrder.c
.....:

/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* (1) NewOrder
* (2) chk_NOData
* (3) setNOData
*
* CREATE by TSL 2002.10.01
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "forlinux.h"

#include "atmi.h"

#include "trans.h"
#include "tpcc_info.h"
#include "newpage.h"

#include "ThreadCntl.h"
#include "TpAplDBDependPrototype.h"
#include "log.h"
#include "log_level.h"

#include "tpapl.h"
//include "GlobalArea.h" // Common
#include "OracleFunction.h"

/* Added T.Kato 04.05.13 Speed up */
int leng_h_new1 = strlen(h_new1);
int leng_h_new2 = strlen(h_new2);
/* Added end */

```

```

/*-----
-----

NewOrder : this function processes the
NewOrder transaction.

-----*/
int NewOrder (char *s_buf, RTE_INPUT_DATA
*in_data, int cookie)
{
    NewOrderData *bp;

    /*int user_id, i;*/
    int i;
    int ol_cnt, cnt, rtn;

    char S_WORK[WORK_SZ];

#ifdef TRNS_BIND
static char* svr_name = "NEWORDER";
#else
static char* svr_name = "OPSTUXSERVER";
#endif
    long olen;

    int h_new1_leng;
    int h_new2_leng;
    int h_new3_leng;

    //SvrAPL return value
#ifdef SCRTST
    int ret_value;
    int ret_val;
    char* tran_errmsg;
#endif

    THREAD_CNTL_INFO* ThreadCntlInfo;
    int return_value;

    MAC_PutFncEntryLog("NewOrder");

    /*user_id = cookie - GLB_TermBase;*/

    /* Create execution environment */
    ThreadCntlInfo = GetThreadCntl();
    if (ThreadCntlInfo == 0) {
        sprintf( S_WORK, "thread contorl
information is not allocated [NEW]\n");
        MAC_errHTML( s_buf, S_WORK, cookie );
        TpccUserLog (LOG_ERR, S_WORK);
        return (-1);
    }
    bp = ( NewOrderData * )ThreadCntlInfo-
>TrxData;
    memset(bp, 0x00, sizeof(NewOrderData));

/* ----- check
the Input data */
    bp->newin.w_id = MAC_w_id(cookie);

    if((bp->newin.d_id = str2int (in_data->D_ID,
2)) < 1 ) {
        TpccUserLog (LOG_ERR, "Input data error
[NEW] (d_id = %s)[Retuen_value:%d]\n",
                in_data->D_ID, bp-
>newin.d_id);
        FreeTuxBuffer(ThreadCntlInfo);
        return set_errpage(s_buf, cookie, 2, (int)bp-
>newin.d_id, 0, 0);
    }

    if((bp->newin.c_id = str2int (in_data->C_ID,
4)) < 0 ) {

```

```

    TpcUserLog (LOG_ERR, "Input data error
[NEW] (c_id = %s)[Return_value:%d]\n",
    in_data->C_ID, bp-
>newin.c_id);
    FreeTuxBuffer(ThreadCntlInfo);
    return set_errpage(s_buf, cookie, 6, bp-
>newin.c_id, 0, 0);
}

    ol_cnt = 0;
    for (cnt = 0; cnt < 15; cnt++){

        if ((rtn = chk_NOdata( bp, cnt, in_data,
ol_cnt)) < 0){
            TpcUserLog (LOG_ERR, "Error end
chk_NOdata() [NEW]
(Line:%d)[Return_Value:%d]\n",
                cnt, rtn);
            FreeTuxBuffer(ThreadCntlInfo);
            return set_errpage(s_buf, cookie, 13 +
cnt, rtn, 0, 0);
        }
        else if (rtn == 1){
            ol_cnt++;
        }
    }

    /* nothing order line data */
    if ( cnt >= 15 && ol_cnt == 0 ) {
        TpcUserLog (LOG_ERR, "nothing order
line data [NEW]\n");
        FreeTuxBuffer(ThreadCntlInfo);
        return set_errpage(s_buf, cookie, 13, -8, 0,
0);
    }

    /* if ol_cnt < 15 then the last order line set
NULL */
    if ( ol_cnt < 15 ){
        bp->newin.ol_i_id[ol_cnt] = 0;
        bp->newin.ol_quantity[ol_cnt] = 0;
        bp->newin.ol_supply_w_id[ol_cnt] = 0;
    }

    bp->newout.o_ol_cnt = ol_cnt;

    /* ----- Execute
NewOrder transaction */
    #ifndef SCRTST
    resend_neworder;

    /* Replaced 2003.12.15 Transaction processeing
interface COM+ --> TUXEDO */
    #ifndef TRNS_BIND
        /* Set transaction type for Warehouse bind */
        bp->retval = 1;
    #endif

        ret_val = tpcall(svr_name,
(char*)ThreadCntlInfo->TrxDData,
sizeof(NewOrderData),
                (char*)&ThreadCntlInfo-
>TrxDData, &olen, 0)TPNOTIME);
        bp = ( NewOrderData *)ThreadCntlInfo-
>TrxDData;
        ret_value = CreateTranErrReason(ret_val, bp-
>newout.terror, &tran_errmsg);

        switch(ret_value) {
        case 0:
            /* Success */
            break;

        case 1:

```

```

        /* Retry NewOrder transaction */
        TpcUserLog (LOG_WRN, "NewOrder
retry\n");
        goto resend_neworder;

        case -1:
            /* Oracle failed */
            sprintf( S_WORK, "Oracle failed to process
NewOrder Transaction.(%s)\n"
                "ret_value = %d d_id = %d c_id = %d
lines = %d cookie = %d\n",
                tran_errmsg, ret_value,
                bp->newin.d_id, bp->newin.c_id,
ol_cnt, cookie );

                MAC_errHTML( s_buf, S_WORK, cookie );
                TpcUserLog (LOG_ERR, S_WORK);
                FreeTuxBuffer(ThreadCntlInfo);
                return (-1);

            default:
                /* Tuxedo failed */
                sprintf( S_WORK, "tpcall failed to process
NewOrder Transaction.(tperno=%d)\n"
                    "ret_value = %d d_id = %d c_id = %d
lines = %d cookie = %d\n",
                    tperno, ret_value,
                    bp->newin.d_id, bp->newin.c_id,
ol_cnt, cookie );

                    MAC_errHTML_TUXEDO( s_buf, S_WORK,
cookie );
                    TpcUserLog (LOG_ERR, S_WORK);
                    FreeTuxBuffer(ThreadCntlInfo);
                    return (-1);
                }
            /* Changed end */
            #else
            dummy_neworder( bp );
            #endif

            /* Replaced T.Kato 04.05.13 Speed up */
            #if 0
            /* Replaced T.Kato 03.04.18 Speed up */
            ! //sprintf( S_WORK, h_new2);
            ! strcpy(S_WORK, h_new2);
            ! h_new2_leng = strlen(S_WORK);
            /* Replaced end */
            #endif

            int2str ((S_WORK + newp[0]), 6, (int)bp-
>newin.w_id);

            int2str ((S_WORK + newp[1]), 2, (int)bp-
>newin.d_id);
            int2str ((S_WORK + newp[3]), 4, bp-
>newin.c_id);

            alp2str ((S_WORK + newp[4]), 16, bp-
>newout.c_last);
            alp2str ((S_WORK + newp[5]), 2, bp-
>newout.c_credit);
            int2str ((S_WORK + newp[7]), 8, (int)bp-
>newout.o_id);

            cnt = bp->newout.o_ol_cnt;

            time2str((S_WORK + newp[2]),bp-
>newout.o_entry_d);
            dec2str ((S_WORK + newp[6]),5,(double)(bp-
>newout.c_discount*100.0));

```

```

            int2str ((S_WORK + newp[8]),2,(int)bp-
>newout.o_ol_cnt);
            dec2str ((S_WORK + newp[9]),5, (double)(bp-
>newout.w_tax * 100.0));
            dec2str ((S_WORK +
newp[10]),5,(double)(bp->newout.d_tax *
100.0));

            for ( i = 0; i < cnt; i++ ) {
                return_value = setNOdata (S_WORK,
0x50*i, i, bp, in_data);
                if (return_value != 0) {
                    TpcUserLog (LOG_ERR, "Error end
setNOdata() [NEW]
(Line:%d)[Return_Value:%d]\n",
                        i, return_value);
                }
            }

            /* "Item number is not valid" or "" ('0) */
            // Oracle Web Server use
            if (strcmp(bp->newout.status, "I") > 0)
                alp2str ((S_WORK + newp[19]), 24, bp-
>newout.status);

            dec2str ((S_WORK + newp[20]), 8,
(double)(bp->newout.total_amount)); // check

            /* ----- The execution result data notified RTE
is make by the HTML form */

            /* Replaced T.Kato 04.05.13 Speed up */
            #if 0
            /* Replaced T.Kato 03.04.18 Speed up */
            #if 0
            !! sprintf(s_buf, h_new1);
            !! strcat (s_buf, S_WORK);
            !!
            !! sprintf(S_WORK, h_new3, SOPATH,
cookie);
            !! strcat (s_buf, S_WORK);
            #endif
            ! strcpy(s_buf, h_new1);
            ! h_new1_leng = strlen(s_buf);
            ! memcpy(s_buf + h_new1_leng, S_WORK,
h_new2_leng);
            ! h_new3_leng = sprintf(S_WORK, h_new3,
SOPATH, cookie);
            ! memcpy(s_buf + h_new1_leng +
h_new2_leng, S_WORK, h_new3_leng);
            ! *(s_buf + h_new1_leng + h_new2_leng +
h_new3_leng) = '\0';
            /* Replaced end */
            #endif
            strcpy(s_buf, h_new1);
            h_new1_leng = leng_h_new1;
            memcpy(s_buf + h_new1_leng, S_WORK,
h_new2_leng);
            h_new3_leng = sprintf(S_WORK, h_new3,
SOPATH, cookie);
            memcpy(s_buf + h_new1_leng +
h_new2_leng, S_WORK, h_new3_leng);
            *(s_buf + h_new1_leng + h_new2_leng +
h_new3_leng) = '\0';
            /* Replaced end */

            FreeTuxBuffer(ThreadCntlInfo);
            return (0);
        }

        #define SUPPLY_NG 0x01
        #define I_ID_NG 0x02
        #define QUANTITY_NG 0x04

```

```

/* -----
-----
chk_NOdata :
VerifyNewOrderLine verifies that a user's
inputs for a line in
the New Order form are okay.
return -5 : w_id abnormal value : Not
Number
return -6 : i_id abnormal value : Not
Number
return -7 : ol_quantity abnormal value : Not
Number

98.8.3 : ----- (-15, -16, -17----:
outside range )

-----
----- */
int chk_NOdata (NewOrderData *bp, int cnt,
RTE_INPUT_DATA *in_data, int svcnt)
{

char flag = 0;

if( in_data->OL_SUPPLY_W_ID[cnt] == 0 &&
in_data->OL_I_ID[cnt] == 0 &&
in_data->OL_QUANTITY[cnt] == 0 ){
/* Order line nothing : 1----- */
return 16; /* change return code */
}

if( in_data->OL_SUPPLY_W_ID[cnt] != 0 ){
if((bp->newin.ol_supply_w_id[svcnt] =
str2int( in_data->OL_SUPPLY_W_ID[cnt],
6) < 1 )
return -5; /* w_id abnormal */
}
else {
flag |= SUPPLY_NG;
}

if( in_data->OL_I_ID[cnt] != 0 ){

if((bp->newin.ol_i_id[svcnt] =
str2int( in_data->OL_I_ID[cnt], 6) < 0 )
return -6; /* i_id abnormal value
*/

/* sv-apl ----- 99.12.20 */
else if (bp->newin.ol_i_id[svcnt] == 0)
bp->newin.ol_i_id[svcnt] = -1;
}
else{
flag |= I_ID_NG;
}

if( in_data->OL_QUANTITY[cnt] != 0 ){
if(((bp->newin.ol_quantity[svcnt] =
str2int( in_data->OL_QUANTITY[cnt], 2))
< 1) ||
bp->newin.ol_quantity[svcnt] > 10 ){

if ( bp->newin.ol_quantity[svcnt] < 0 )
return -7; /* ol_quantity
abnormal value */
else
return -17; /* outside range */
}
}
else{
flag |= QUANTITY_NG;
}

if (flag != 0){

```

```

/* the order lien data is abnormal : there is a
uninput item */
if((flag & SUPPLY_NG) != 0 ) return -8;
if((flag & I_ID_NG) != 0 ) return -1;
if((flag & QUANTITY_NG) != 0) return -2;
return 1;
}
else{
/* the order lien data is normal */
return 1;
}
}

/* -----
-----
setNOdata : This function set the execution
result data of the TP
applicatin program.

OF is an offset value to the next line data.
cnt is line number
-----
----- */
int setNOdata (char *s_work,int OF,int cnt,
NewOrderData *bp,RTE_INPUT_DATA
*in_data)
{
/*for warning
in_data;

if((bp->newin.ol_i_id[cnt] ) {
alp2str ((s_work + OF + newp[11]), 78, " ");
return -1;
}
else {
int2str((s_work + OF + newp[11]), 6, (int)bp-
>newin.ol_supply_w_id[cnt]);

if (bp->newin.ol_i_id[cnt] == -1 )
bp->newin.ol_i_id[cnt] = 0;
int2str((s_work + OF + newp[12]), 6, bp-
>newin.ol_i_id[cnt]);

alp2str((s_work + OF + newp[13]), 24, bp-
>newout.i_name[cnt]);

int2str((s_work + OF + newp[14]), 2, (int)bp-
>newin.ol_quantity[cnt]);
int2str((s_work + OF + newp[15]), 3, (int)bp-
>newout.s_quantity[cnt]);
alp2str((s_work + OF + newp[16]), 1, &bp-
>newout.brand_generic[cnt]);

dec2str((s_work + OF + newp[17]),
6,(double)bp->newout.i_price[cnt]); // check
dec2str((s_work + OF + newp[18]),
7,(double)bp->newout.ol_amount[cnt]); // check
return 0;
}
}

:-----:
tpapl/trnexe/TrxOrderStatus.c
:-----:

/*****
****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
*

```

```

* (1) OrderStatus
*
* CREATE by TSL 2003.12.15
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *

-----
****/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "forlinux.h"

#include "atmi.h"

#include "trans.h"
#include "tpcc_info.h"
#include "odpage.h"

#include "ThreadCntl.h"
#include "TpAplDBDependPrototype.h"
#include "log.h"
#include "log_level.h"

#include "tpapl.h"
// #include "GlobalArea.h" // Common
#include "OracleInfo.h"
#include "OracleFunction.h"

/* Added T.Kato 04.05.13 Speed up */
int leng_h_order1 = strlen(h_order1);
int leng_h_order2 = strlen(h_order2);
/* Added end */

/*-----
-----
OrderStatus : this function processes the
Orderstatus transaction
-----
----- */
int OrderStatus (char *s_buf,
RTE_INPUT_DATA *in_data, int cookie)
{
OrderStatusData *bp;
int i, rtn;

char S_WORK[WORK_S];
char c_id_flag = NG;

#ifdef TRNS_BIND
static char* svr_name = "ORDERSTATUS";
#else
static char* svr_name = "OPSTUXSERVER";
#endif

long olen;

int h_order1_leng;
int h_order2_leng;
int h_order3_leng;

//SvrAPL return value
#ifdef SCRTEST
int ret_value;
int ret_val;
char* tran_errmsg;
#endif

THREAD_CNTL_INFO* ThreadCntlInfo;

MAC_PutFuncEntryLog("OrderStatus");

ThreadCntlInfo = GetThreadCntl();

```

```

if (ThreadCntlInfo == 0) {
    sprintf( S_WORK, "thread contorl
information is not allocated [ODR]\n");
    MAC_errHTML( s_buf, S_WORK, cookie );
    TpcUserLog (LOG_ERR, S_WORK);
    return (-1);
}
bp = ( OrderStatusData * )ThreadCntlInfo-
>TrxDat;
memset(bp, 0x00, sizeof(OrderStatusData));

/* ----- check
the Input data */
bp->ordin.w_id = MAC_w_id(cookie);

/* check d_id data */
if ((bp->ordin.d_id = str2short (in_data->D_ID,
2)) < 1) {
    TpcUserLog (LOG_ERR, "Input data error
[ORD] (d_id = %s)[Return_Value:%d]\n",
in_data->D_ID, bp->ordin.d_id);
    FreeTuxBuffer(ThreadCntlInfo);
    return set_errpage(s_buf, cookie, 2, (int)bp-
>ordin.d_id, 0, 0);
}

if ((bp->ordin.c_id = str2int(in_data->C_ID,
4)) != -3){

    if (bp->ordin.c_id < 0) {
        TpcUserLog (LOG_ERR, "Input data
error [ORD] (c_id = %s)[Return_Value:%d]\n",
in_data->C_ID, bp-
>ordin.c_id);
        FreeTuxBuffer(ThreadCntlInfo);
        return set_errpage(s_buf, cookie, 6, bp-
>ordin.c_id, 0, 0);
    }
    else{
        c_id_flag = OK;
    }
}
else{
    bp->ordin.c_id = 0;
}

/* check c_last data */
if((rtn = str2str(in_data->C_LAST, 16)) < 0){
    c_id_flag = OK;
}
else{
    if ( rtn == 0 || *(in_data->C_LAST) == '\0' ) {
        bp->ordin.bylastname = 0; /* Oracle
use only */
        bp->ordin.c_last[0] = '\0';
    }
    else {
        strcpy (bp->ordin.c_last, in_data-
>C_LAST);
        bp->ordin.bylastname = 1; /* Oracle
use only */
        c_id_flag = OK;
    }
}

/* c_id and c_last is nothing */
if (c_id_flag == NG) {
    TpcUserLog (LOG_ERR, "c_id and c_last
is nothing [ORD]\n");
    FreeTuxBuffer(ThreadCntlInfo);
    return set_errpage(s_buf, cookie, 11, -4, 0,
0);
}

```

```

}

/* ----- Execute
Orderstatus transaction */
#ifdef SCRTST
resend_orderstatus:

/* Replaced 2003.12.15 Transaction
processeing interface COM+ --> TUXEDO */
#ifdef TRNS_BIND
/* Set transaction type for Warehouse bind */
bp->retval = 3;
#endif

    retval = tpcall(svr_name,
(char*)ThreadCntlInfo->TrxDat,
sizeof(NewOrderData),
(char*)&ThreadCntlInfo-
>TrxDat, &olen, 0|TPNOTIME);
    bp = ( OrderStatusData * )ThreadCntlInfo-
>TrxDat;
    ret_value = CreateTranErrReason(retval, bp-
>ordout.terror, &tran_errmsg);

    switch(retval) {
    case 0:
        /* Success */
        break;

    case 1:
        /* Retry OrderStatus transaction */
        TpcUserLog (LOG_WRN, "OrderStatus
retry\n");
        goto resend_orderstatus;

    case -1:
        /* Oracle failed */
        sprintf( S_WORK, "Oracle failed to
process Order Status Transaction.(%s)\n"
"ret_value = %d d_id = %d c_id = %d
c_last = %s' cookie = %d\n",
tran_errmsg, ret_value, bp-
>ordin.d_id, bp->ordin.c_id,
bp->ordin.c_last, cookie );

        MAC_errHTML( s_buf, S_WORK, cookie );
        TpcUserLog (LOG_ERR, S_WORK);
        FreeTuxBuffer(ThreadCntlInfo);
        return (-1);

    default:
        /* Tuxedo failed */
        sprintf( S_WORK, "tpcall failed to process
NewOrder Transaction.(tperno=%d)\n"
"ret_value = %d d_id = %d c_id = %d
c_last = %s' cookie = %d\n",
tperno, ret_value, bp->ordin.d_id, bp-
>ordin.c_id,
bp->ordin.c_last, cookie );

        MAC_errHTML_TUXEDO( s_buf, S_WORK,
cookie );
        TpcUserLog (LOG_ERR, S_WORK);
        FreeTuxBuffer(ThreadCntlInfo);
        return (-1);
    }
}
/* Changed end */

/* ----- Check the
execution result */

#else
    dummy_orderstat( bp );
#endif

```

```

/* Replaced T.Kato 04.05.13 Speed up */
#if 0
/* Replaced T.Kato 03.04.18 Speed up */
! //sprintf(S_WORK, h_order2);
! strcpy(S_WORK, h_order2);
! h_order2_leng = strlen(S_WORK);
/* Replaced end */
#endif
    strcpy(S_WORK, h_order2);
    h_order2_leng = leng_h_order2;
/* Relaced end */

    int2str ((S_WORK + orderp[0]), 6, (int)bp-
>ordin.w_id);
    int2str ((S_WORK + orderp[1]), 2, (int)bp-
>ordin.d_id);
    int2str ((S_WORK + orderp[2]), 4, bp-
>ordout.c_id);
    alp2str ((S_WORK + orderp[3]), 16, bp-
>ordout.c_first);
    alp2str ((S_WORK + orderp[4]), 2, bp-
>ordout.c_middle);
    alp2str ((S_WORK + orderp[5]), 16, bp-
>ordout.c_last);
    sigdec2str ((S_WORK + orderp[6]), 9, bp-
>ordout.c_balance);
    int2str ((S_WORK + orderp[7]), 8, (int)bp-
>ordout.o_id );
    time2str ((S_WORK + orderp[8]), bp-
>ordout.o_entry_d );

    if ( bp->ordout.o_carrier_id != INTNULL ) {
        int2str ((S_WORK + orderp[9]), 2, bp-
>ordout.o_carrier_id);
    }

    /* 0x39 is an offset value to the same filed of
the next line */
    for ( i = 0; i < bp->ordout.o_ol_cnt; i++ ){

        int2str ((S_WORK+i*0x3a+orderp[10]), 6,
(int)bp->ordout.ol_supply_w_id[i]);

        int2str ((S_WORK+i*0x3a+orderp[11]), 6,
(int)bp->ordout.ol_i_id[i]);
        int2str ((S_WORK+i*0x3a+orderp[12]), 2,
(int)bp->ordout.ol_quantity[i]);
        sigdec2str ((S_WORK+i*0x3a+orderp[13]),
8, (double)bp->ordout.ol_amount[i]);

        if( strcmp( bp->ordout.ol_delivery_d[i],
"NOT DELIVR", 10 ) != 0 ){

            date2str ((S_WORK+i*0x3a+orderp[14]),
bp->ordout.ol_delivery_d[i]);
        }
    }

    /* ----- The execution result data notified RTE
is make by the HTML form */
/* Replaced T.Kato 04.05.13 Speed up */
#if 0
/* Replaced T.Kato 03.04.18 Speed up */
! #if 0
!! sprintf(s_buf, h_order1); /* set Header Data
*/
!! strcat (s_buf, S_WORK); /* set Result
Data */
!!
!! sprintf (S_WORK, h_order3, SOPATH,
cookie); /* set Tailer Data */
!! strcat (s_buf, S_WORK);
! #endif
! strcpy(s_buf, h_order1);
! h_order1_leng = strlen(s_buf);

```

```

! memcpy(s_buf + h_order1_leng, S_WORK,
h_order2_leng);
! h_order3_leng = sprintf (S_WORK, h_order3,
SOPATH, cookie);
! memcpy(s_buf + h_order1_leng +
h_order2_leng, S_WORK, h_order3_leng);
! *(s_buf + h_order1_leng + h_order2_leng +
h_order3_leng) = '\0';
!/* Replaced end */
#endif
strcpy(s_buf, h_order1);
h_order1_leng = leng_h_order1;
memcpy(s_buf + h_order1_leng, S_WORK,
h_order2_leng);
h_order3_leng = sprintf (S_WORK, h_order3,
SOPATH, cookie);
memcpy(s_buf + h_order1_leng +
h_order2_leng, S_WORK, h_order3_leng);
*(s_buf + h_order1_leng + h_order2_leng +
h_order3_leng) = '\0';
!/* Replaced end */

FreeTuxBuffer(ThreadCntlInfo);
return 0;
}

.....
tpapl/trnexe/TrxPayment.c
.....
!*****
****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* (1) Payment
*
* CREATE by TSL 2003.12.15
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
****/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "forlinux.h"

#include "atmi.h"

#include "trans.h"
#include "tpcc_info.h"
#include "paypage.h"

#include "ThreadCntl.h"
#include "TpAplDBDependPrototype.h"
#include "log.h"
#include "log_level.h"

#include "tpapl.h"
#include "GlobalArea.h" // Common
#include "OracleFunction.h"

!/* Added T.Kato 04.03.10 Speed up */
#define SP1_DATA " "
#define SP2_DATA " "
#define SP3_DATA " "
#define CREDIT_DATA " Credit:
"
#define DISC_DATA " %Disc: "

```

```

int leng_h_pay1 = strlen(h_pay1);
int leng_h_pay2 = strlen(h_pay2);
int leng_h_pay4 = strlen(h_pay4);
int leng_h_pay5 = strlen(h_pay5);
int leng_sp1_data = strlen(SP1_DATA);
int leng_sp2_data = strlen(SP2_DATA);
int leng_sp3_data = strlen(SP3_DATA);
int leng_credit_data =
strlen(CREDIT_DATA);
int leng_disc_data = strlen(DISC_DATA);
!/* Added end */

!-----
-----

Payment : this function processes the
Payment transaction.

!-----*/
int Payment (char *s_buf, RTE_INPUT_DATA
*in_data, int cookie)
{
PaymentData *bp;
int i, rtn;

float h_amount; /* For work */

char c_id_flag = NG;
char S_WORK[WORK_SZ];

char buffer[128]; /* check HTML form */
char buffer2[128];
char buffer3[512];
int newlength;

#ifdef TRNS_BIND
static char* svr_name = "PAYMENT";
#else
static char* svr_name = "OPSTUXSERVER";
#endif
long olen;

//SvrAPL return value
#ifndef SCRTEST
int ret_value;
int ret_val;
char* tran_errmsg;
#endif

THREAD_CNTL_INFO* ThreadCntlInfo;
#ifndef SCRTEST
#endif

!/* Added T.Kato 04.03.10 */
int next_pos;
int swork_pos;
!/* Added end */

MAC_PutFncEntryLog("Payment");

ThreadCntlInfo = GetThreadCntl();
if (ThreadCntlInfo == 0) {
sprintf (S_WORK, "thread contorl
information is not allocated [PAY]\n");
MAC_errHTML (s_buf, S_WORK, cookie );
TpccUserLog (LOG_ERR, S_WORK);
return (-1);
}
bp = ( PaymentData *)ThreadCntlInfo-
>TrxData;
memset(bp, 0x00, sizeof(PaymentData));

```

```

!/* ----- check
the Input data */
bp->payin.w_id = MAC_w_id(cookie);

!/* check d_id data */
if((bp->payin.d_id = str2short (in_data->D_ID,
2)) < 1 ) {
TpccUserLog (LOG_ERR, "Input data error
[PAY] (d_id = %s)[Return_Value:%d]\n",
in_data->D_ID, bp->payin.d_id);
FreeTuxBuffer(ThreadCntlInfo);
return set_errpage(s_buf, cookie, 2, (int)bp-
>payin.d_id, 0, 0);
}

!/* check c_id data */
if((bp->payin.c_id = str2int (in_data->C_ID,
4)) != -3){

if (bp->payin.c_id < 0) {
TpccUserLog (LOG_ERR, "Input data
error [PAY] (c_id = %s)[Return_Value:%d]\n",
in_data->C_ID, bp-
>payin.c_id);
FreeTuxBuffer(ThreadCntlInfo);
return set_errpage(s_buf, cookie, 6, bp-
>payin.c_id, 0, 0);
}
else{
c_id_flag = OK;
}
}
else{
bp->payin.c_id = 0;
}

!/* check c_last data */
if((rtn = str2str(in_data->C_LAST, 16)) < 0){
c_id_flag = OK;
}
else{

if (rtn == 0 || *(in_data->C_LAST) == '\0') {
bp->payin.bylastname = 0; /*
Oracle use only */
bp->payin.c_last[0] = '\0';
} else {
strcpy (bp->payin.c_last, in_data-
>C_LAST);
bp->payin.bylastname = 1; /*
Oracle use only */
c_id_flag = OK;
}
}

!/* c_id and c_last data is nothing */
if (c_id_flag == NG) {
TpccUserLog (LOG_ERR, "c_id and c_last
data is nothing [PAY]\n");
FreeTuxBuffer(ThreadCntlInfo);
return set_errpage(s_buf, cookie, 11, -4, 0,
0);
}

!/* check c_w_id data */
!/* Replaced T.Kato 03.08.20 Bug fix --effect
floating point-- */
!/* if((bp->payin.c_w_id = str2dbl (in_data-
>C_W_ID, 5) / 100) < 1) (*/*
if((bp->payin.c_w_id = str2int (in_data-
>C_W_ID, 6)) < 1) {
!/* Replaced end */

```

```

    TpccUserLog (LOG_ERR, "Input data error
[PAY] (c_w_id = %s)[Return_Value:%d]\n",
    in_data->C_W_ID, bp-
>payin.c_w_id);
    FreeTuxBuffer(ThreadCntlInfo);
    return set_errpage(s_buf, cookie, 9, (int)bp-
>payin.c_w_id, 0, GLB_Numwh);
}

/* check c_d_id data */
if((bp->payin.c_d_id = str2short (in_data-
>C_D_ID, 2)) < 1 ) {
    TpccUserLog (LOG_ERR, "Input data error
[PAY] (c_d_id = %s)[Return_Value:%d]\n",
    in_data->C_D_ID, bp-
>payin.c_d_id);
    FreeTuxBuffer(ThreadCntlInfo);
    return set_errpage(s_buf, cookie, 10,
(int)bp->payin.c_d_id, 0, 0);
}

if((bp->payin.h_amount = (long)str2dbl
(in_data->H_AMOUNT, 7)) < 100 ||
    bp->payin.h_amount > 500000) {
    TpccUserLog (LOG_ERR, "Input data error
[PAY] (h_amount = %s)[Return_Value:%d]\n",
    in_data->H_AMOUNT, bp-
>payin.h_amount);
    FreeTuxBuffer(ThreadCntlInfo);
    return set_errpage(s_buf, cookie, 8, (int)bp-
>payin.h_amount, 0, 0);
}

/* ----- Execute
Payment transaction */
#ifdef SCRTTEST
resend_payment:

/* Replaced 2003.12.15 Transaction processeing
interface COM+ --> TUXEDO */
#ifdef TRNS_BIND
/* Set transaction type for Warehouse bind */
    bp->retval = 2;
#endif

    ret_val = tpcall(svr_name,
(char*)ThreadCntlInfo->TrxDData,
sizeof(NewOrderData),
(char*)&ThreadCntlInfo-
>TrxDData, &olen, 0|TPNOTIME);
    bp = (PaymentData *)ThreadCntlInfo-
>TrxDData;
    ret_value = CreateTranErrReason(ret_val, bp-
>payout.terror, &tran_errmsg);

    switch(ret_value) {
    case 0:
        /* Success */
        break;

    case 1:
        /* Retry Payment transaction */
        TpccUserLog (LOG_WRN, "Payment
retry\n");
        goto resend_payment;

    case -1:
        /* Oracle failed */
        sprintf (S_WORK, "Oracle failed to
process Payment Transaction.(%s)\n"
            "ret_value = %d d_id = %d c_id = %d
c_last = %s\n"
            "c_w_id = %d, c_d_id = %d, h_amount
= %d cookie = %d\n",

```

```

    tran_errmsg, ret_value,
    bp->payin.d_id, bp->payin.c_id, bp-
>payin.c_last,
    bp->payin.c_w_id, bp->payin.c_d_id,
    bp->payin.h_amount, cookie );

    MAC_errHTML( s_buf, S_WORK, cookie );
    TpccUserLog (LOG_ERR, S_WORK);
    FreeTuxBuffer(ThreadCntlInfo);
    return (-1);

    default:
        /* Tuxedo failed */
        sprintf( S_WORK, "tpcall failed to process
NewOrder Transaction.(tperno=%d)\n"
            "ret_value = %d d_id = %d c_id = %d
c_last = %s\n"
            "c_w_id = %d, c_d_id = %d, h_amount
= %d cookie = %d\n",
            tperno, ret_value,
            bp->payin.d_id, bp->payin.c_id, bp-
>payin.c_last,
            bp->payin.c_w_id, bp->payin.c_d_id,
            bp->payin.h_amount, cookie );
        MAC_errHTML_TUXEDO( s_buf, S_WORK,
cookie );
        TpccUserLog (LOG_ERR, S_WORK);
        FreeTuxBuffer(ThreadCntlInfo);
        return (-1);
    }
/* Changed end */

#else
    dummy_payment( bp );
#endif

/* Replaced T.Kato 04.03.10 Speed up */
#if 0
/* Replaced T.Kato 03.04.18 Speed up */
! //sprintf (S_WORK, h_pay2);
! strcpy(S_WORK, h_pay2);
/* Replaced end */
#endif

    memcpy(S_WORK, h_pay2, leng_h_pay2+1);
    swork_pos = leng_h_pay2;
/* Replaced end */

    time2str ((S_WORK + payp[0]), bp-
>payout.h_date );
    int2str ((S_WORK + payp[1]), 6, (int)bp-
>payin.w_id);
    int2str ((S_WORK + payp[2]), 2, (int)bp-
>payin.d_id);

    // check HTML form

    alp2str (&buffer2[0], 20, bp-
>payout.w_street_1);
    buffer2[20] = 0;

/* Replaced T.kato 04.03.10 Speed up */
#if 0
! newlength = checkHTMLform ( &buffer2[0],
&buffer[0]);
! strcpy (&buffer3[0], &buffer[0]);
! strcat (buffer3, " ");
#endif

    newlength = checkHTMLform ( buffer2,
buffer3);
    memcpy(buffer3+newlength, SP1_DATA,
leng_sp1_data+1);
    next_pos = newlength + leng_sp1_data;
/* Replaced end */

```

```

    alp2str (buffer2, 20, bp->payout.d_street_1);
    buffer2[20] = 0;
    newlength = checkHTMLform ( &buffer2[0],
&buffer[0]);

/* Replaced T.Kato 04.03.10 Speed up */
#if 0
! strcat (buffer3, &buffer[0]);
! strcat (buffer3, "\r\n");
#endif

    memcpy(buffer3+next_pos, buffer,
newlength+1);
    next_pos += newlength;
    memcpy(buffer3+next_pos, "\r\n", 2+1);
    next_pos += 2;
/* Replaced end */

    alp2str (buffer2, 20, bp->payout.w_street_2);
    buffer2[20] = 0;
    newlength = checkHTMLform ( &buffer2[0],
&buffer[0]);

/* Replaced T.Kato 04.03.10 Speed up */
#if 0
! strcat (buffer3, &buffer[0]);
! strcat (buffer3, " ");
#endif

    memcpy(buffer3+next_pos, buffer,
newlength+1);
    next_pos += newlength;
    memcpy(buffer3+newlength, SP1_DATA,
leng_sp1_data+1);
    next_pos = newlength + leng_sp1_data;
/* Replaced end */

    alp2str (buffer2, 20, bp->payout.d_street_2);
    buffer2[20] = 0;
    newlength = checkHTMLform ( &buffer2[0],
&buffer[0]);

/* Replaced T.Kato 04.03.10 Speed up */
#if 0
! strcat (buffer3, &buffer[0]);
! strcat ( S_WORK, buffer3 );
#endif

    memcpy(buffer3+next_pos, buffer,
newlength+1);
    next_pos += newlength;
    memcpy(buffer3+next_pos, "\r\n", 2+1);
    next_pos += 2;

    memcpy(S_WORK+swork_pos, buffer3,
next_pos+1);
    swork_pos += next_pos;
/* Replaced end */

    // check HTML form
/* Replaced T.Kato 04.03.10 Speed up */
#if 0
/* Replaed T.Kato 03.04.18 Speed up */
! //sprintf ( buffer3, h_pay4 );
! strcpy ( buffer3, h_pay4 );
/* Replaced end */
#endif

    memcpy(buffer3, h_pay4, leng_h_pay4+1);
/* Replaced end */

    alp2str ((&buffer3[0] + payp[7] - 0xd3), 20, bp-
>payout.w_city);

```



```

    alp2str ((&buffer3[0] + payp[8] - 0xd3), 2, bp-
->payout.w_state);
    zip2str ((&buffer3[0] + payp[9] - 0xd3), bp-
->payout.w_zip);
    alp2str ((&buffer3[0] + payp[11] - 0xd3), 20,
bp->payout.d_city);
    alp2str ((&buffer3[0] + payp[12] - 0xd3), 2, bp-
->payout.d_state);
    zip2str ((&buffer3[0] + payp[13] - 0xd3), bp-
->payout.d_zip);

    int2str ((&buffer3[0] + payp[15] - 0xd3), 4, bp-
->payout.c_id);
    int2str ((&buffer3[0] + payp[16] - 0xd3), 6,
(int)bp->payin.c_w_id);
    int2str ((&buffer3[0] + payp[17] - 0xd3), 2,
(int)bp->payin.c_d_id);

    alp2str ((&buffer3[0] + payp[18] - 0xd3), 16,
bp->payout.c_first);
    alp2str ((&buffer3[0] + payp[19] - 0xd3), 2, bp-
->payout.c_middle);
    alp2str ((&buffer3[0] + payp[20] - 0xd3), 16,
bp->payout.c_last);

    date2str ((&buffer3[0] + payp[21] - 0xd3), bp-
->payout.c_since);

/* Replaced T.Kato 04.03.10 Speed up */
/*strcat (S_WORK, buffer3);*/

    memcpy(S_WORK+swork_pos, buffer3,
leng_h_pay4+1);
    swork_pos += leng_h_pay4;
/* Replaced end */

/* Replaced T.Kato 04.03.10 Speed up*/
/*strcpy (&buffer3[0], " ");*/

    memcpy(buffer3, SP2_DATA,
leng_sp2_data+1);
    next_pos = leng_sp2_data;
/* Replaced end */

    alp2str (buffer2, 20, bp->payout.c_street_1);
    buffer2[20] = 0;
    newlength = checkHTMLform (&buffer2[0],
&buffer[0]);

/* Replaced T.Kato 04.03.10 */
#if 0
!   strcat (buffer3, &buffer[0]);
!   strcat (buffer3, "          Credit:");
#endif

    memcpy(buffer3+next_pos, buffer,
newlength+1);
    next_pos += newlength;
    memcpy(buffer3+next_pos, CREDIT_DATA,
leng_credit_data+1);
    next_pos += leng_credit_data;
/* Replaced end */

    alp2str (buffer2, 2, bp->payout.c_credit);
    buffer2[2] = 0;

/* Replaced T.Kato 04.03.10 */
#if 0
!   strcat (buffer3, &buffer2[0]);
!   strcat (buffer3, "\r\n");
#endif

    memcpy(buffer3+next_pos, buffer2, 2);
    memcpy(buffer3+next_pos+2, "\r\n", 3);
    next_pos += 4;

```

```

/* Replaced end */

/* Replaced T.Kato 04.03.10 */
/*strcat (buffer3, " ");*/

    memcpy(buffer3+next_pos, SP2_DATA,
leng_sp2_data+1);
    next_pos += leng_sp2_data;
/* Replaced end */

    alp2str (buffer2, 20, bp->payout.c_street_2);
    buffer2[20] = 0;
    newlength = checkHTMLform (&buffer2[0],
&buffer[0]);

/* Replaced T.Kato 04.03.10 */
#if 0
!   strcat (buffer3, &buffer[0]);
!   strcat (buffer3, "          %Disc:");
!   strcat (S_WORK, buffer3);
#endif

    memcpy(buffer3+next_pos, buffer,
newlength+1);
    next_pos += newlength;
    memcpy(buffer3+next_pos, DISC_DATA,
leng_disc_data+1);
    next_pos += leng_disc_data;

    memcpy(S_WORK+swork_pos, buffer3,
next_pos+1);
    swork_pos += next_pos;
/* Replaced end */

    dec2str (&buffer3[0], 5,
(double)((double)(bp->payout.c_discount) *
(double)100.0));

/* Replaced T.Kato 04.03.10 Speed up */
#if 0
!   sprintf (&buffer3[5], "\r\n");
!   strcat (S_WORK, buffer3);
#endif

    buffer3[5] = '\r';
    buffer3[6] = '\n';
    buffer3[7] = '\0';

    memcpy(S_WORK+swork_pos, buffer3, 7+1);
    swork_pos += 7;
/* Replaced end */

/* Replaced T.Kato 04.03.10 Speed up */
#if 0
!/* Replaced T.Kato 03.04.18 Speed up */
! //sprintf (buffer3, h_pay5);
!   strcpy (buffer3, h_pay5);
!/* Replaced end */
#endif

    memcpy(buffer3, h_pay5, leng_h_pay5+1);
/* Replaced end */

    alp2str ((&buffer3[0] + payp[26] - 0x21D), 20,
bp->payout.c_city);
    alp2str ((&buffer3[0] + payp[27] - 0x21D), 20,
bp->payout.c_state);
    zip2str ((&buffer3[0] + payp[28] - 0x21D), bp-
->payout.c_zip);
    phone2str ((&buffer3[0] + payp[29] - 0x21D),
bp->payout.c_phone);

    h_amount = (float)bp->payin.h_amount /
(float)100;

```

```

    dec2str ((&buffer3[0] + payp[30] - 0x21D), 7,
(double)h_amount);

    sigdec2str ((&buffer3[0] + payp[31] - 0x21D),
14, bp->payout.c_balance);
    dec2str ((&buffer3[0] + payp[32] - 0x21D), 13,
bp->payout.c_credit_lim);

/* Replaced T.Kato 04.03.10 */
/*strcat (S_WORK, buffer3);*/

    memcpy(S_WORK+swork_pos, buffer3,
leng_h_pay5+1);
    swork_pos += leng_h_pay5;
/* Replaced end */

    if ((i = strlen (bp->payout.c_data)) <= 0) {

/* Replaced T.Kato 04.03.10 Speed up */
/*sprintf (&buffer3[0], "\r\n\r\n\r\n");*/

        memcpy(buffer3, "\r\n\r\n\r\n\r\n", 8+1);
/* Replaced end */

    }
    else{
        alp2str (buffer2, 50, bp->payout.c_data);
        buffer2[50] = 0;
        newlength = checkHTMLform (&buffer2[0],
&buffer[0]);

/* Replaced T.Kato 04.03.10 Speed up */
#if 0
!   strcpy (&buffer3[0], &buffer[0]);
!   strcat (buffer3, "\r\n");
#endif

        memcpy(buffer3, buffer, newlength+1);
        memcpy(buffer3+newlength, "\r\n", 2+1);
        next_pos = newlength + 2;
/* Replaced end */

        if (i > 50){

            alp2str (buffer2, 50, &bp-
->payout.c_data[50]);
            buffer2[50] = 0;
            newlength = checkHTMLform
(&buffer2[0], &buffer[0]);

/* Replaced T.Kato 04.03.10 Speed up */
#if 0
!   strcat (buffer3, " ");
!   strcat (buffer3, &buffer[0]);
!   strcat (buffer3, "\r\n");
#endif

            memcpy(buffer3+next_pos, SP3_DATA,
leng_sp3_data+1);
            next_pos += leng_sp3_data;
            memcpy(buffer3+next_pos, buffer,
newlength+1);
            next_pos += newlength;
            memcpy(buffer3+next_pos, "\r\n", 2+1);
            next_pos += 2;
/* Replaced end */
            if (i > 100){

                alp2str (buffer2, 50, &bp-
->payout.c_data[100]);
                buffer2[50] = 0;
                newlength = checkHTMLform
(&buffer2[0], &buffer[0]);

```

```

/* Replaced T.Kato 04.03.10 Speed up */
#if 0
!      strcat (buffer3, " ");
!      strcat (buffer3, &buffer[0]);
!      strcat (buffer3, "\r\n");
#endif

      memcpy(buffer3+next_pos,
SP3_DATA, leng_sp3_data+1);
      next_pos += leng_sp3_data;
      memcpy(buffer3+next_pos, buffer,
newlength+1);
      next_pos += newlength;
      memcpy(buffer3+next_pos, "\r\n",
2+1);
      next_pos += 2;
/* Replaced end */

      if (i > 150){

          alp2str (buffer2, 50, &bp-
> payout.c_data[150]);
          buffer2[50] = 0;
          newlength = checkHTMLform
(&buffer2[0], &buffer[0]);

/* Replaced T.Kato 04.03.10 Speed up */
#if 0
!      strcat (buffer3, " ");
!      strcat (buffer3, &buffer[0]);
!      strcat (buffer3, "\r\n");
#endif

      memcpy(buffer3+next_pos,
SP3_DATA, leng_sp3_data+1);
      next_pos += leng_sp3_data;
      memcpy(buffer3+next_pos, buffer,
newlength+1);
      next_pos += newlength;
      memcpy(buffer3+next_pos, "\r\n",
2+1);
      next_pos += 2;
/* Replaced end */

      }
      else {

/* Replaced T.Kato 04.03.10 Speed up */
/*strcat ( buffer3, "\r\n\r\n");*/

      memcpy(buffer3+next_pos, "\r\n",
2+1);
      next_pos += 2;
/* Replaced end */
      }
      else {

/* Replaced T.Kato 04.03.10 Speed up */
/*strcat ( buffer3, "\r\n\r\n\r\n");*/

      memcpy(buffer3+next_pos, "\r\n\r\n",
4+1);
      next_pos += 4;
/* Replaced end */
      }
      }

/* Added T.Kato 04.03.10 Speed up */
else {
      memcpy(buffer3+next_pos,
"\r\n\r\n\r\n", 6+1);
      next_pos += 6;
}
/* Added end */

```

```

}

/* Replaced T.Kato 04.03.10 Speed up */
/*strcat (S_WORK, buffer3);*/

      memcpy(S_WORK+swork_pos, buffer3,
next_pos);
      swork_pos += next_pos;
/* Replaced end */

/* ----- The execution result data notified RTE
is make by the HTML form */
/* Replaced T.Kato 04.03.10 Speed up */
#if 0
/* Replaced T.Kato 03.04.18 Speed up */
! //sprintf(s_buf, h_pay1); /* set Header Data
*/
! strcpy(s_buf, h_pay1); /* set Header Data */
/* Replaced end */
!
! strcat (s_buf, S_WORK); /* set Result Data
*/
!
! sprintf(S_WORK, h_pay3, SOPATH, cookie);
/* set Tailer Data */
! strcat (s_buf, S_WORK);
#endif

      memcpy(s_buf, h_pay1, leng_h_pay1+1);
      memcpy(s_buf+leng_h_pay1, S_WORK,
swork_pos+1);
      next_pos = sprintf(S_WORK, h_pay3,
SOPATH, cookie); /* set Tailer Data */
      memcpy(s_buf+leng_h_pay1+swork_pos,
S_WORK, next_pos+1);
/* Replaced end */

      FreeTuxBuffer(ThreadCntlInfo);
      return (0);
}

.....
tpapl/trnexe/TrxStockLevel.c
.....

/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* (1) StockLevel
*
* CREATE by TSL 2003.12.15
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "forlinux.h"
#include "atmi.h"

#include "trans.h"
#include "tpcc_info.h"
#include "stpage.h"

```

```

#include "ThreadCntl.h"
#include "TpAplDBDependPrototype.h"
#include "log.h"
#include "log_level.h"

#include "tpapl.h"
#include "GlobalArea.h" // Common
#include "OracleFunction.h"

/* Added T.Kato 04.05.13 Speed up */
int leng_h_stock1 = strlen(h_stock1);
int leng_h_stock2 = strlen(h_stock2);
/* Added end */

/*-----
-----

StockLevel : this function processes the
StockLevel transaction.

-----*/
int StockLevel (char *s_buf, RTE_INPUT_DATA
*in_data, int cookie)
{
      StockLevelData *bp;

      char S_WORK[WORK_S];

#ifdef TRNS_BIND
static char *svr_name = "STOCKLEVEL";
#else
static char *svr_name = "OPSTUXSERVER";
#endif
      long olen;

      int h_stock1_leng;
      int h_stock2_leng;
      int h_stock3_leng;

      //SvrAPL return value
#ifdef SCRTEST
      int ret_value;
      int ret_val;
      char* tran_errmsg;
#endif

      THREAD_CNTL_INFO* ThreadCntlInfo;

      MAC_PutFuncEntryLog("StockLevel");

      ThreadCntlInfo = GetThreadCntl();
      if (ThreadCntlInfo == 0) {
          sprintf (S_WORK, "thread contorl
information is not allocated [STO]\n");
          MAC_errHTML( s_buf, S_WORK, cookie );
          TpcUserLog (LOG_ERR, S_WORK);
          return (-1);
      }
      bp = ( StockLevelData *)ThreadCntlInfo-
>TrxDData;
      memset(bp, 0x00, sizeof(StockLevelData));

/* ----- check
the Input data */
      bp->stoin_w_id = MAC_w_id(cookie);
      bp->stoin_d_id = MAC_d_id(cookie);

      bp->stoin.threshold = (long)str2short(in_data-
>threshold, 2);

      if(bp->stoin.threshold < 10 || bp-
>stoin.threshold > 20) {

```

```

    TpcUserLog (LOG_ERR, "Input data error
[STO] (threshold = %s)[Return_Value:%d]\n",
    in_data->threshold, bp-
>stoin.threshold);
    return set_errpage(s_buf, cookie, 3, (int)bp-
>stoin.threshold, 0, 0);
}

/* ----- Execute Stock
Level transaction */
#ifdef SCRTST
resend_stock:

/* Replaced 2003.12.15 Transaction processeing
interface COM+ --> TUXEDO */
#ifdef TRNS_BIND
    /* Set transaction type for Warehouse bind */
    bp->retval = 5;
#endif

    ret_val = tpcall(svr_name,
(char*)ThreadCntlInfo->TrxDData,
sizeof(NewOrderData),
(char*)&ThreadCntlInfo-
>TrxDData, &oLen, 0|TPNOTIME);
    bp = ( StockLevelData *)ThreadCntlInfo-
>TrxDData;
    ret_value = CreateTranErrReason(ret_val, bp-
>stout.error, &tran_errmsg);

    switch(ret_value) {
    case 0:
        /* Success */
        break;

    case 1:
        /* Retry Paymant transaction */
        TpcUserLog (LOG_WRN, "StockLevel
retry\n");
        goto resend_stock;

    case -1:
        /* Oracle failed */
        sprintf( S_WORK, "Oracle failed to
process StockLevel Transaction.(%s)\n"
"ret_value = %d threshold = %d cookie
= %d\n",
            tran_errmsg, ret_value, bp-
>stoin.threshold, cookie );

        MAC_errHTML( s_buf, S_WORK, cookie );
        TpcUserLog (LOG_ERR, S_WORK);
        FreeTuxBuffer(ThreadCntlInfo);
        return (-1);

    default:
        /* Tuxedo failed */
        sprintf( S_WORK, "tpcall failed to process
NewOrder Transaction.(tperrno=%d)\n"
"ret_value = %d threshold = %d cookie
= %d\n",
            tperrno, ret_value, bp-
>stoin.threshold, cookie );

        MAC_errHTML_TUXEDO( s_buf, S_WORK,
cookie );
        TpcUserLog (LOG_ERR, S_WORK);
        FreeTuxBuffer(ThreadCntlInfo);
        return (-1);
    }
}
/* Changed end */

#else
dummy_stocklvl ( bp );
#endif

```

```

/* Replaced T.Kato 04.05.13 Speed up */
#if 0
/* Replaced T.Kato 03.04.18 Speed up */
! //sprintf (S_WORK, h_stock2);
! strcpy (S_WORK, h_stock2);
! h_stock2_leng = strlen(S_WORK);
/* Replaced end */
#endif

    strcpy (S_WORK, h_stock2);
    h_stock2_leng = leng_h_stock2;
/* Replaced end */

    int2str ((S_WORK + stockp[0]), 6, (int)bp-
>stoin.w_id);

    int2str ((S_WORK + stockp[1]), 2, (int)bp-
>stoin.d_id);
    int2str ((S_WORK + stockp[2]), 2, (int)bp-
>stoin.threshold);
    int2str ((S_WORK + stockp[3]), 3, (int)bp-
>stout.low_stock);

/* ----- The execution result data notified RTE
is make by the HTML form */

/* Replaced T.Kato 04.05.13 Speed up */
#if 0
/* Replaced T.Kato 03.04.18 Speed up */
!#if 0
!! sprintf(s_buf, h_stock1); /* Set Header data
*/
!! strcat (s_buf, S_WORK); /* Set Result
data */
!!
!! sprintf(S_WORK, h_stock3, SOPATH,
cookie); /* Set Tailer data */
!! strcat (s_buf, S_WORK);
!#endif
! strcpy(s_buf, h_stock1);
! h_stock1_leng = strlen(s_buf);
! memcpy(s_buf + h_stock1_leng, S_WORK,
h_stock2_leng);
! h_stock3_leng = sprintf(S_WORK, h_stock3,
SOPATH, cookie);
! memcpy(s_buf + h_stock1_leng +
h_stock2_leng, S_WORK, h_stock3_leng);
! *(s_buf + h_stock1_leng + h_stock2_leng +
h_stock3_leng) = '\0';
/* Replaced end */
#endif
    strcpy(s_buf, h_stock1);
    h_stock1_leng = leng_h_stock1;
    memcpy(s_buf + h_stock1_leng, S_WORK,
h_stock2_leng);
    h_stock3_leng = sprintf(S_WORK, h_stock3,
SOPATH, cookie);
    memcpy(s_buf + h_stock1_leng +
h_stock2_leng, S_WORK, h_stock3_leng);
    *(s_buf + h_stock1_leng + h_stock2_leng +
h_stock3_leng) = '\0';
/* Replaced end */

    FreeTuxBuffer(ThreadCntlInfo);
    return (0);
}

```

```

.....
tpapl/trnexe/log_level.h
.....

```

```

/******
****

```

```

*
* TPC-C Client Application Program Source
*
*
* CREATE by TSL 2003.02.07
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *

*****
****/

#define PUT_INF_LOG //
Information log
#define PUT_FNC_ENTRY_LOG //
Function entry point log
#define PUT_FNC_EXIT_LOG //
Function exit log

/* Function entry point log macro */
#ifdef PUT_FNC_ENTRY_LOG
#define MAC_PutFncEntryLog(func)
TpcUserLog(LOG_INF, ">>>>> "func" start
>>>>>");
#else
#define MAC_PutFncEntryLog(func) ;
#endif

/* Function exit point log */
#ifdef PUT_FNC_EXIT_LOG
#define MAC_PutFncExitLog(func)
TpcUserLog(LOG_INF, "<<<<< "func" end
<<<<<");
#else
#define MAC_PutFncExitLog(func) ;
#endif

```

Appendix B: Server Source Code

```

.....
blocks/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_cit
y, :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;

.....
blocks/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
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_st
ate,
: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_c
ity,
: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;

TPC Benchmark C Full Disclosure

```

```

.....
blocks/tkvcin.sql
.....

-- The initnew package for storing variables used
in the
-- New Order anonymous block

CREATE OR REPLACE PACKAGE initppcc
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 initppcc;
/
show errors;

CREATE OR REPLACE PACKAGE BODY
initppcc 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 initppcc;
/
show errors
exit

.....
blocks/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 = initppcc.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;

.....
blocks/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,
        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,
initppcc.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,
initppcc.s_dist,
          :ol_amount, :brand_generic;
  END u2;

```



```

        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)
        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
        INTO :c_discount, :c_last, :c_credit
        FROM cust
            WHERE c_id = :c_id AND c_d_id = :d_id
        AND c_w_id = :w_id;

        SELECT w_tax
        INTO :w_tax
        FROM ware
            WHERE 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 ord (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,
            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;

    .....
    common/GetPrivateProfileString.c
    .....

    /*****
    *
    * TPC-C Client Application Program Source
    *
    * Entry Functions *
    * (1) GetPrivateProfileString
    *
    *
    * CREATE by TSL 2003.12.18
    *
    * All Right Reserved, Copyright Co. FUJITSU
    LIMITED 2003-2004 *
    *****/

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

    /*****
    *
    * Get data string corresponded key in
    cogfiguration file. *
    *****/
    
```

```

* Return Value          *
*   Get string length   *
*****
****/
int  GetPrivateProfileString(char*
section_name, /* Section name
*/
name          char* key_name, /* Key
*/
char* default_str, /* Default
string, if key nothing */
char* key_data, /* Key
data */
int  buf_size, /* Buffer
size of key data */
char* file_name) { /* File
name */

FILE* prof_file;
char  read_buf[256];
char  search[32];
char* get_str;
char* key_pos=0;
int  get_cnt;
int  i;

/* Open profile file */
if ((prof_file = fopen(file_name, "r")) == NULL)
{
goto DEFAULT_STRING;
}

/* Make searching section name "[section
name]" */
search[0] = '\0';
strcpy(&search[1], section_name);
strcat(search, ".ini");

/* Search section name */
while((get_str = fgets(read_buf,
sizeof(read_buf), prof_file)) != NULL) {

/* Search section name form to be read one
line */
if ((char*)strstr(read_buf, search) == NULL)
{
/* No match section name, next line read
*/
continue;
}
break;
}
if (get_str == NULL) {
/* Found EOF or read error */
goto DEFAULT_STRING_FCLOSE;
}

/* Make searching key name "key_name=" */
strcpy(search, key_name);
strcat(search, "=");

/* Search key name in this section */
while((get_str = fgets(read_buf,
sizeof(read_buf), prof_file)) != NULL) {
for (i = 0; read_buf[i] == ' ' || read_buf[i] ==
'\t'; i++);
if (read_buf[i] == '[') {
/* Other section started, undefined key
name */
goto DEFAULT_STRING_FCLOSE;
}
if ((key_pos = (char*)strstr(read_buf,
search)) == NULL) {

```

```

/* No match key name */
continue;
}
break;
}
if (get_str == NULL) {
/* Found EOF or read error */
goto DEFAULT_STRING_FCLOSE;
}

fclose(prof_file);

/* Get key_value, fixed format "key value" */
for (; *key_pos != '\0'; key_pos++)
key_pos++;
for (get_cnt = 0; *key_pos != '\0'; key_pos++) {
/* Get & set key value */
*key_data = *key_pos;
key_data++;
get_cnt++;
if (get_cnt >= (buf_size - 1)) {
/* Key data buffer full */
break;
}
}
*key_data = '\0';
return(get_cnt);

DEFAULT_STRING_FCLOSE:
fclose(prof_file);

DEFAULT_STRING:
strcpy(key_data, default_str, buf_size-1);
return(strlen(key_data));
}

.....
common/MakeShell
.....

#!/bin/sh
cd /home/tpc/client_apl/common
make > make_result.txt 2>&1

.....
common/Makefile
.....

#-----
-----
# Makefile : Makefile for common of TPAPL and
SVRAPL.
#
# Created by TSL 2003.12.17
#
# All Right Rserverd, Copyright Co, FUJITSU
LIMITED 2003-2004.
#-----
-----

# GCC compile configurations
AR = ar
ARFLAGS = rv
CFLAGS = -Wall -O2
CC = gcc

# MACRO definition
DMACRO = -DTSL -DPLSQLFLAG=1 -DTUX

# home directory
ORADIR = /usr/local/oracle

```

```

TUXDIR = /usr/local/BEA/tuxedo8.1
SVRDIR = /home/tpc/client_apl/svrpl

# include directory
ORA_INC = -I$(ORADIR)/rdbms/demo -
I$(ORADIR)/rdbms/public
COM_INC = -I$(SVRDIR)/common
SVR_INC = -I$(SVRDIR)
TUX_INC = -I$(TUXDIR)/include
INCLUDE = $(COM_INC) $(SVR_INC)
$(ORA_INC) $(TUX_INC)

# target object
COMOBSJ = log.o sema.o
GetPrivateProfileString.o shmем.o
COMLIB = libcom.a

INCFILS = log.h sema.h forlinux.h shmем.h

$(COMLIB) : $(COMOBSJ)
$(AR) $(ARFLAGS) $(COMLIB) $(COMOBSJ)

.SUFFIXES : .o .c
.c.o:
$(CC) -o $@ -c $(CFLAGS) $(INCLUDE)
$(DMACRO) $<

$(COMOBSJ) : $(INCFILS)

clean:
rm $(COMLIB) $(COMOBSJ)

.....
common/forlinux.h
.....

/*****
*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* definition for converting Linux.
*
*
* CREATE by TSL 2003.05.16
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *
*****
*****/
/* forlinux.h */

#include <limits.h>
#define MAX_PATH PATH_MAX /*
Windows:MAX_PATH , Linux:PATH_MAX */
#define Sleep(x) poll(0, 0, x); /* sleep unit is
a msec. */

.....
common/log.c
.....

/*****
*****

```



```

*
* TPC-C Client Application Program Source
*
* Entry Functions
* Log is outputted to a file.
*
* CREATE by TSL 2002.11.29
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002-2004 *
*****/
#include "forlinux.h"
#include <stdio.h>
#include <string.h>
#include <time.h>
#include <sys/types.h>
#include <stdarg.h>
#include <unistd.h>
#include <pthread.h>
#include <sys/types.h>
#include <sys/stat.h>
#include "sema.h"

#define LOG_MODULE
#include "log.h"

void TpcUserLog(char* file_name, int line_no,
char* type_name, char* ftmp, ...)
{
FILE* fp;
pid_t pid;
pthread_t tld;
char* fname;
int stat;

/* -- BEGIN -- Modified by Hayashi for thread-
safe. 2006/02/13 */
#if 0
! struct tm *nowtime;
#else
struct tm tt;
struct tm *nowtime=&tt;
#endif
/* -- END -- Modified by Hayashi for thread-safe.
2006/02/13 */

time_t long_time;
va_list va;

if (strcmp(type_name, "LCK") != 0) {
/* Lock semaphore */
stat = LockSem(GLB_LogSemId);
}
/* Get current time. */

time(&long_time);

/* -- BEGIN -- Modified by Hayashi for thread-
safe. 2006/02/13 */
#if 0
! nowtime = localtime(&long_time);
#else
localtime_r(&long_time, nowtime);
#endif
/* -- END -- Modified by Hayashi for thread-safe.
2006/02/13 */

/* Get process Id. */
pid = getpid();

/* Get thread Id. */

```

```

tld = pthread_self();

/* Get just file name from a path. */
fname = (char*)strchr(file_name, (int)'/');
if (fname == NULL) {
fname = file_name;
} else {
fname = fname + 1;
}

va_start(va, ftmp);

fp = fopen(GLB_LogFilePath, "a");
fprintf(fp, "%02d:%02d:%02d [%6d:%08x] %-
32s(%4d) :%s: ",
nowtime->tm_hour, nowtime->tm_min,
nowtime->tm_sec, pid, (int)tld, fname, line_no,
type_name);
vfprintf(fp, ftmp, va);

if (*(ftmp + strlen(ftmp) - 1) != '\n')
fprintf(fp, "\n");

va_end(va);

fclose(fp);

/* change mode which all users can read and
write. */
chmod(GLB_LogFilePath, S_IRUSR
|S_IWUSR |S_IRGRP|S_IWGRP| S_IROTH |
S_IWOTH);

if (strcmp(type_name, "LCK") != 0) {
// Unlock semaphore
stat = UnlockSem(GLB_LogSemId);
}

return;
}

.....
common/log.h
.....

/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* Log is outputted to a file.
*
* CREATE by TSL 2002.11.29
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *
*****/

void TpcUserLog (char *file_name, int line_no,
char *type_name, char *ftmp, ...);

extern char GLB_LogFilePath[MAX_PATH];
extern int GLB_LogSemId;

#define DEFAULT_SVRAPL_LOG_PATH
"/home/tpc/log/DBDepend_Userlog.log"
#define DEFAULT_TPAPL_LOG_PATH
"/home/tpc/log/userlog.log"

```

```

#define LOG_ERR __FILE__, __LINE__, "ERR"
#define LOG_INF __FILE__, __LINE__, "INF"
#define LOG_WRN __FILE__, __LINE__,
"WRN"
#define LOG_LCK __FILE__, __LINE__, "LCK"

#define LOG_FILE_INF __FILE__,
__LINE__, "INF"
#define LOG_FILE_LINE __FILE__,
__LINE__

.....
common/sema.c
.....

/*****
*
* TPC-C Client Application Program Source
*
* Filename :
* sema.c
* Entry Functions :
* There are functions to control semaphore.
*
* CREATE by TSL 2003.12.18
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *
*****/

#include "forlinux.h"
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#include <errno.h>
#include "log.h"
#include "sema.h"

/*****
****
* Initialize semaphore.
* Return Value
* > 0 semaphore Id. (always over 0)
*
* < 0 fail.
*
*****
****/

int InitSem(char *path, int projectId)
{
int sid;
union semun{
int val;
struct semid_ds *buf;
ushort *array;
} c_arg;

TpcUserLog(LOG_LCK, "InitSem: start
path<%s> projectId=%d\n",
path, projectId);

if ((sid = GetSem(path, projectId)) == -1) {
TpcUserLog(LOG_LCK, "GetSem() fail,
path<%s> projectId=%d\n",

```

```

        path, projectId);
    return(-1);
}
c_arg.val=1;
if (semctl(sid,0,SETVAL,c_arg)==-1) {
    TpccUserLog(LOG_LCK, "semctl fail,
sid=%d\n",sid);
    return(-1);
}
TpccUserLog(LOG_LCK, "InitSem: Get
semid=%d\n",sid);

    return(sid);
}
/*****
* Get semaphore.
* Return Value
* > 0 semaphore Id. (always over 0)
*
* < 0 fail.
*
*****/
int GetSem(char *path, int projectId)
{
    int sid;
    int key;

    if ((key = ftok(path,projectId) == -1) {
        TpccUserLog(LOG_LCK, "ftok() fail,
path<%s> projectId=%d errno=%d\n",
        path, projectId, errno);
        return(-1);
    }
    if ((sid=semget(key,1,0666|IPC_CREAT))== -
1){
        TpccUserLog(LOG_LCK, "semget() fail,
key=%d errno=%d\n",key, errno);
        return(-1);
    }

    return(sid);
}
/*****
* Reuire to lock semaphore.
*
* Return Value
* 1 success.
* -1 fail.
*
*****/
int LockSem(int sid)
{
    struct sembuf sb;

    sb.sem_num=0;
    sb.sem_op=-1;
    sb.sem_flg=0;
    if(semop(sid,&sb,1)== -1) {
        TpccUserLog(LOG_LCK, "semop() fail,
sid=%d\n",sid);
        return(-1);
    }
    return(1);
}
/*****
* Reuire to unlock semaphore.
*
* Return Value
* 1 success.
* -1 fail.
*
*****/

```

```

*****
*****/
int UnlockSem(int sid)
{
    struct sembuf sb;

    sb.sem_num=0;
    sb.sem_op=1;
    sb.sem_flg=0;
    if(semop(sid,&sb,1)== -1){
        TpccUserLog(LOG_LCK, "semop() fail,
sid=%d\n",sid);
        return(-1);
    }
    return(1);
}

:-----:
common/sema.h
:-----:

/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* Semaphore control.
*
* CREATE by TSL 2003.12.19
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *
*****/

/*== project Id =====*/
#define SEM_SVRAPL_PROJID
(int)'S'
#define SEM_TPAPL_PROJID (int)'T'
#define SEM_SAMPLING_PERFORMANCE
(int)'P'

/*=====
====*/
/* prototype definition */
/*=====
====*/
int InitSem(char *path, int projectId);
int GetSem(char *path, int projectId);
int LockSem(int sid);
int UnlockSem(int sid);

:-----:
common/shmem.c
:-----:

/*****
*
* TPC-C Client Application Program Source
*
*
* Filename :
* sema.c
* Entry Functions :
*
*****/

```

```

* There are functions to control shared
memory.
*
* CREATE by TSL 2004.01.15
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *
*****/

#include "forlinux.h"
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <errno.h>
#include "log.h"

/*****
*
* Initialize shared memory.
* Return Value
* > 0 shared memory address. (always over
0)
* < 0 fail.
*
*****/
char* InitShmem(char *path, int projectId, int
size)
{
    int shmId;
    int key;
    char *shmaddr;

    TpccUserLog(LOG_LCK, "InitShmem: start
path<%s> projectId=%d\n",
    path, projectId);

    if ((key = ftok(path,projectId) == -1) {
        TpccUserLog(LOG_LCK, "ftok() fail,
path<%s> projectId=%d errno=%d\n",
        path, projectId, errno);
        return((char *)-1);
    }
    if
((shmId=shmget(key,size,IPC_CREAT|0666))==
-1){
        TpccUserLog(LOG_LCK, "shmget() fail,
key=%d errno=%d",key, errno);
        return((char *)-1);
    }
    if( shmaddr = (char *)shmat(shmId, NULL, 0)
== (char *)-1 ) {
        TpccUserLog(LOG_LCK, "shmat() fail,
shmId=%d path<%s> projectId=%d errno=%d\n",
        shmId, path, projectId, errno);
        return ((char *)-1);
    }

    TpccUserLog(LOG_LCK, "InitShmem: Get
shmId =%d shmaddr = %08x\n",shmId,
shmaddr);

    return(shmaddr);
}
/*****
*
* Get shared memory.
* Return Value
* > 0 shared memory address. (always over
0)
* < 0 fail.
*
*****/

```

```

*****
****/
char* GetShmem(char *path, int projectld, int
size)
{
    int shmld;
    int key;
    char *shmaddr;

    if ((key = ftok(path,projectld)) == -1) {
        TpcUserLog(LOG_LCK, "ftok() fail,
path<%s> projectld=%d errno=%d\n",
        path, projectld,errno);
        return((char *)-1);
    }
    if ((shmld=shmget(key,size, 0))==-1){
        TpcUserLog(LOG_LCK, "shmget() fail,
key=%d errno=%d\n",key,errno);
        return((char *)-1);
    }
    if ((shmaddr = (char *)shmat(shmld, NULL, 0))
== (char *)-1) {
        TpcUserLog(LOG_LCK, "shmat() fail,
shmld=%d path<%s> projectld=%d errno=%d\n",
        shmld, path, projectld, errno);
        return ((char *)-1);
    }

    return(shmaddr);
}

.....
common/shmem.h
.....

/*****
*****
*
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Shared memory control.
*
*
*
* CREATE by TSL 2004.01.15
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *
*****
*****/

/== project ld =====*/
#define
SHMEM_SAMPLING_PERFORMANCE
(int)'P'

/=====
====*/
/* prototype definition */
/=====
====*/
char* InitShmem(char *path, int projectld, int
size);
char* GetShmem(char *path, int projectld, int
size);

.....

```

```

svrap/GlobalArea.c
.....
/*****
*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Global Area definition.
*
* CREATE by TSL 2003.05.16
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *
*****
*****/
#include "forlinux.h"
#include "tpcc.h"
#include "tpccflags.h"
#include "TrnCntrlInfo.h"

char GLB_LogFilePath[MAX_PATH];
char GLB_ConfigFilePath[MAX_PATH];
int GLB_LogSemld;

/* Global area for Oracle interfase. */
/* ----- */
/* Delivery (pldel.cpp) */
/* ----- */
pldelctx *pldctx;
delctx *dctx;
#ifdef DMLRETDDEL
amtctx *actx;
#endif
/* ----- */
/* NewOrder (plnew.cpp) */
/* ----- */
newctx *nctx;
/* ----- */
/* OrderStatus (plord.cpp) */
/* ----- */
ordctx *octx;
defctx cbctx;

/* Added T.Kato 2004.12.21 New Oracle10g tool
kit */
int ordcount = 0;
#ifdef DEBUG
int trace_on = 0;
#endif
/* Added end */

/* ----- */
/* Payment (plpay.cpp) */
/* ----- */
payctx *pctx;
/* ----- */
/* StockLevel (plsto.cpp) */
/* ----- */
stoctx *sctx;
/* ----- */
/* (tpccpl.cpp) */
/* ----- */
FILE *lfp;
/* Deleted T.Kato 02.10.23 for warning
!FILE *fopen ();
Deleted end */

/* Added t.Kato 02.10.24 for Delivery logging file
control */
int iflg; /* Delivery log initialize flag */

```

```

/* Added end */
int proc_no;
int logon;
int new_init;
int pay_init;
int ord_init;

#ifdef DEL_ORA8I
int del_init;
#else
int del_init_oci;
int del_init_plsql;
#endif

int sto_init;
int res_init;

int execstatus;
int errcode;

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

/* for stock-level transaction */
int w_id;
int d_id;
int c_id;

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! int threshold;
#endif

#ifdef USE_IEEE_NUMBER
float threshold;
#else
int threshold;
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

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;
int o_ol_cnt;
int ol_supply_w_id[15];
int ol_i_id[15];

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! int ol_quantity[15];
! int ol_amount[15];
#endif

#ifdef USE_IEEE_NUMBER
float ol_quantity[15];
float ol_amount[15];
#else

```

```

int ol_quantity[15];
int ol_amount[15];
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

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;

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! int h_amount;
#endif

#ifdef USE_IEEE_NUMBER
float h_amount;
#else
int h_amount;
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

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];

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! int nol_quantity[15];
! int nol_amount[15];
! int s_quantity[15];
! int i_price[15];
#endif

#ifdef USE_IEEE_NUMBER
float nol_quantity[15];
float nol_amount[15];
float s_quantity[15];
float i_price[15];
#else
int nol_quantity[15];
int nol_amount[15];
int s_quantity[15];
int i_price[15];

```

```

#endif /* USE_IEEE_NUMBER */
/* Replaced end */

int nol_quant10[15];
int nol_quant91[15];
int nol_ytdqty[15];
int o_all_local;
float w_tax;
float d_tax;
/* Deleted T.Kato 02.11.13
!float total_amount;
Deleted end */
char i_name[15][25];
char brand_gen[15];
char brand_generic[15][1];
int status;
int tracelevel;

OCIDate cr_date;
OCIDate c_since;
OCIDate o_entry_d_base;
OCIDate ol_d_base[15];
dvoid *xmem;
/* ----- */
/* (tpccsvr.cpp) */
/* ----- */
/* set up pointers for type casting */
struct newstruct *newinfo;
struct paystruct *payinfo;
struct ordstruct *ordinfo;
struct delstruct *delinfo;
struct stostruct *stoinfo;

#ifdef AVOID_DEADLOCK
int indx[NITEMS], ordl_cnt;
#endif

.....
svrap/GlobalArea.h
.....

/*****
*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions *
* Global Area definition. *
*
* CREATE by TSL 2003.05.16
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *
*****
*****/

#include "tpccflags.h"
#include "TrnCntrlInfo.h"

extern char GLB_LogFilePath[MAX_PATH];
extern char GLB_ConfigFilePath[MAX_PATH];
extern int GLB_LogSemId;

#define TPCC_CONF_FILE
"/home/tpccconf/tpapl.conf"

```

```

#define LOG_FILE_NAME_THREAD
"log\\SvrThread%05d.log"

/* Global area for Oracle interfase. */
/* ----- */
/* Delivery (pldel.cpp) */
/* ----- */
extern pldelctx *pldctx;
extern delctx *dctx;
#ifdef DMLRETDEL
extern amtctx *actx;
#endif
/* ----- */
/* NewOrder (plnew.cpp) */
/* ----- */
extern newctx *nctx;
/* ----- */
/* OrderStatus (plord.cpp) */
/* ----- */
extern ordctx *octx;
extern defctx *cbctx;

/* Added T.Kato 2004.12.21 New Oracle10g tool
kit */
extern int ordcount;
#ifdef DEBUG
extern int trace_on;
#endif
/* Added end */

/* ----- */
/* Payment (plpay.cpp) */
/* ----- */
extern payctx *pctx;
/* ----- */
/* StockLevel (plsto.cpp) */
/* ----- */
extern stoctx *sctx;
/* ----- */
/* (tpccpl.cpp) */
/* ----- */
extern FILE *lfp;
/* Deleted T.Kato 02.10.23 for warning
!FILE *fopen ();
Deleted end */

/* Added t.Kato 02.10.24 for Delivery logging file
control */
extern int iflg; /* Delivery log initialize flag */
/* Added end */
extern int proc_no;
extern int logon;
extern int new_init;
extern int pay_init;
extern int ord_init;

#ifdef DEL_ORA81
extern int del_init;
#else
extern int del_init_oci;
extern int del_init_plsql;
#endif

extern int sto_init;
extern int res_init;

extern int execstatus;
extern int errcode;

extern OCIEnv *tpcenv;
extern OCIServer *tpcsrv;

```

```

extern OCLError *errhp;
extern OCISvcCtx *tpcsvc;
extern OCISession *tpcusr;
extern OCISmt *curi;

/* for stock-level transaction */
extern int w_id;
extern int d_id;
extern int c_id;

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! extern int threshold;
#endif

#ifdef USE_IEEE_NUMBER
extern float threshold;
#else
extern int threshold;
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

extern int low_stock;

/* for delivery transaction */
extern int del_o_id[10];
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 ub4 datelen;
extern int o_carrier_id;
extern int o_ol_cnt;
extern int ol_supply_w_id[15];
extern int ol_i_id[15];

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! extern int ol_quantity[15];
! extern int ol_amount[15];
#endif

#ifdef USE_IEEE_NUMBER
extern float ol_quantity[15];
extern float ol_amount[15];
#else
extern int ol_quantity[15];
extern int ol_amount[15];
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

extern 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;

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! extern int h_amount;
#endif

#ifdef USE_IEEE_NUMBER
extern float h_amount;

```

```

#else
extern int h_amount;
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

extern char w_street_1[21];
extern char w_street_2[21];
extern char w_city[21];
extern char w_state[3];
extern char w_zip[10];
extern char d_street_1[21];
extern char d_street_2[21];
extern char d_city[21];
extern char d_state[3];
extern char d_zip[10];
extern char c_street_1[21];
extern char c_street_2[21];
extern char c_city[21];
extern char c_state[3];
extern char c_zip[10];
extern char c_phone[17];
extern ub4 sincelen;
extern text c_since_d[11];
extern float c_discount;
extern char c_credit[3];
extern int c_credit_lim;
extern char c_data[201];
extern ub4 hlen;
extern text h_date[20];

/* for new order transaction */

extern int nol_i_id[15];
extern int nol_supply_w_id[15];

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! extern int nol_quantity[15];
! extern int nol_amount[15];
! extern int s_quantity[15];
! extern int i_price[15];
#endif

#ifdef USE_IEEE_NUMBER
extern float nol_quantity[15];
extern float nol_amount[15];
extern float s_quantity[15];
extern float i_price[15];
#else
extern int nol_quantity[15];
extern int nol_amount[15];
extern int s_quantity[15];
extern int i_price[15];
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

extern int nol_quant10[15];
extern int nol_quant19[15];
extern int nol_ytdqty[15];
extern int o_all_local;
extern float w_tax;
extern float d_tax;
/* Deleted T.Kato 02.11.13
!float total_amount;
Deleted end */
extern char i_name[15][25];
extern char brand_gen[15];
extern char brand_generic[15][1];
extern int status;
extern int tracelevel;

extern OCIDate cr_date;
extern OCIDate c_since;
extern OCIDate o_entry_d_base;
extern OCIDate ol_d_base[15];

```

```

extern dvoid *xmeme;
/* ----- */
/* (tpccsvr.cpp) */
/* ----- */
/* set up pointers for type casting */
extern struct newstruct *newinfo;
extern struct paystruct *payinfo;
extern struct ordstruct *ordinfo;
extern struct delstruct *delinfo;
extern struct stostruct *stoinfo;

#ifdef AVOID_DEADLOCK
int indx[NITEMS], ordl_cnt;
#endif

.....
svrapi/MakeShell
.....

#! /bin/sh
cd /home/tpc/client_apl/svrapi
make > make_result.txt 2>&1

.....
svrapi/Makefile
.....

#-----
-----
# Makefile : Makefile for 3 tier and 2 tier
executing files on Linux.
#
# Created by TSL 2003.12.17
#
# All Right Rserverd, Copyright Co, FUJITSU
LIMITED 2003-2004.
#-----
-----

# GCC compile configurations
AR = ar
ARFLAGS = rv
CFLAGS = -Wall -O2
CC = gcc
LD = gcc

# MACRO definition
#DMACRO = -DTSL -DPLSQLFLAG=1 -DTUX
-DDGLDEF
DMACRO = -DPLSQLFLAG=1 -DTUX

# home directory.
ORADIR = /usr/local/oracle
TUXDIR = /usr/local/BEA/tuxedo8.1
SVRDIR = /home/tpc/client_apl/svrapi
TPDIR = /home/tpc/client_apl/tpapl
COMDIR = /home/tpc/client_apl/common
SVRCOMDIR = $(COMDIR)

# include directory
ORA_INC = -I$(ORADIR)/rdbms/demo -
I$(ORADIR)/rdbms/public
COM_INC = -I$(COMDIR)
TUX_INC = -I$(TUXDIR)/include
TP_INC = -I$(TPDIR)
INCLUDE = $(COM_INC) $(ORA_INC)
$(TUX_INC) $(TP_INC)
OBJDIR = $(SVRDIR)/bin

# target object

```

```

3TIERDIR = /home/tpc/client_apl/svrapl/3tier
COMDIR = /home/tpc/client_apl/common
COMOBS = tpccsvr.o GlobalArea.o
initsvrconfig.o
ALLOBS = $(COMOBS) $(MAIN_WHBOBJ)
$(MAIN_NEWOBJ) $(MAIN_PAYOBJ)
$(MAIN_DELOBJ) \
$(MAIN_STOOBJ) $(MAIN_ORDOBJ)
3TIERLIB = $(3TIERDIR)/libtier.a
COMLIB = $(COMDIR)/libcom.a

# depend on include file.
INCFIL = $(SVRDIR)/tpcc.h
$(SVRDIR)/GlobalArea.h $(SVRDIR)/prototype.h \
$(SVRDIR)/tpccflags.h
$(SVRDIR)/tpcc_info.h $(SVRDIR)/TrnCntrlInfo.h
$(SVRDIR)/tpcc_info.h \
$(COMDIR)/log.h $(COMDIR)/sema.h
$(COMDIR)/forlinux.h $(TPDIR)/SampleInfo.h

#---- transaction or warehouse main object.
MAIN_WHBOBJ = bs-whb.o
MAIN_NEWOBJ = bs-new.o
MAIN_PAYOBJ = bs-pay.o
MAIN_DELOBJ = bs-del.o
MAIN_STOOBJ = bs-sto.o
MAIN_ORDOBJ = bs-ord.o

# tuxedo
TUXLIBS = $(TUXDIR)/lib/libtux.a
$(TUXDIR)/lib/libbuff.a $(TUXDIR)/lib/libfml.a \
$(TUXDIR)/lib/libfml32.a
$(TUXDIR)/lib/libengine.a -pthread -ldl
#TUXLIBS = -L$(TUXDIR)/lib/ -ltux -lbuff -lfml -
lfml32
# Oracle
#ORALIB = -L$(ORADIR)/rdbms/demo
#ORALIBS = $(ORADIR)/lib/libocci10.a
#ORALIBS = $(ORADIR)/rdbms/lib/defopt.o
$(ORADIR)/lib/libclntst10.a
#ORALIBS = $(ORADIR)/lib/libclntst10.a

#---- execute file for 3 tier.
TARGET_WHB_3TIER =
$(OBJDIR)/3tier_tpccfmlw
TARGET_NEW_3TIER =
$(OBJDIR)/3tier_tpccfmln
TARGET_PAY_3TIER =
$(OBJDIR)/3tier_tpccfmlp
TARGET_DEL_3TIER =
$(OBJDIR)/3tier_tpccfmlf
TARGET_STO_3TIER =
$(OBJDIR)/3tier_tpccfmls
TARGET_ORD_3TIER =
$(OBJDIR)/3tier_tpccfmlf

3TIERTARGETS = $(TARGET_WHB_3TIER)
$(TARGET_NEW_3TIER)
$(TARGET_PAY_3TIER) \
$(TARGET_DEL_3TIER)
$(TARGET_ORD_3TIER)
TARGETS = $(3TIERTARGETS)

# link library.
#LDFLAGS=-L$(ORACLE_HOME)/rdbms/lib/ -
L$(ORACLE_HOME)/lib/ -dy \
# -L$(ORACLE_HOME)/rdbms/lib/ -
L$(ORACLE_HOME)/lib/ \
# $(ORACLE_HOME)/rdbms/lib/defopt.o -
lclntsh \
# -ldl -lm -pthread -lnsl

```

```

LDFLAGS=-L$(ORACLE_HOME)/rdbms/lib/ -
L$(ORACLE_HOME)/lib/ -dy \
-L$(ORACLE_HOME)/rdbms/lib/ -
L$(ORACLE_HOME)/lib/ \
$(ORACLE_HOME)/rdbms/lib/defopt.o -
lclntsh \
-ldl -lm -pthread -lnsl

$(TARGETS) : $(ALLOBS) $(3TIERLIB)
$(COMLIB)
$(LD) -o $(TARGET_WHB_3TIER)
$(MAIN_WHBOBJ) $(COMOBS) $(COMLIB)
$(3TIERLIB) $(TUXLIBS) $(ORALIBS)
$(LDFLAGS)
$(LD) -o $(TARGET_NEW_3TIER)
$(MAIN_NEWOBJ) $(COMOBS) $(COMLIB)
$(3TIERLIB) $(TUXLIBS) $(ORALIBS)
$(LDFLAGS)
$(LD) -o $(TARGET_PAY_3TIER)
$(MAIN_PAYOBJ) $(COMOBS) $(COMLIB)
$(3TIERLIB) $(TUXLIBS) $(ORALIBS)
$(LDFLAGS)
$(LD) -o $(TARGET_DEL_3TIER)
$(MAIN_DELOBJ) $(COMOBS) $(COMLIB)
$(3TIERLIB) $(TUXLIBS) $(ORALIBS)
$(LDFLAGS)
$(LD) -o $(TARGET_STO_3TIER)
$(MAIN_DELOBJ) $(COMOBS) $(COMLIB)
$(3TIERLIB) $(TUXLIBS) $(ORALIBS)
$(LDFLAGS)
$(LD) -o $(TARGET_ORD_3TIER)
$(MAIN_ORDOBJ) $(COMOBS) $(COMLIB)
$(3TIERLIB) $(TUXLIBS) $(ORALIBS)
$(LDFLAGS)

.SUFFIXES: .o .c
.c.o:
$(CC) -o $@ -c $(CFLAGS) $(INCLUDE)
$(DMACRO) $<

$(ALLOBS) : $(INCFIL)
$(ALLOBS) : Makefile

clean:
rm $(ALLOBS) $(TARGETS)

.....
svrapl/TrnCntrlInfo.h
.....

/*****
****
*
*
* TPC-C Client Application Program Source
*
*
* Entry Functions *
* Transaction structure object definition.
*
*
* CREATE by TSL 2003.05.16
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
****/

/* ----- */
/* Delivery Struct */
/* ----- */

```

```

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];
OCISmt *curd0;
OCISmt *curd1;
OCISmt *curd2;
OCISmt *curd3;
OCISmt *curd4;
OCISmt *curd5;
OCISmt *curd6;
OCISmt *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 {

    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];

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
/* Replaced T.kato 03.07.18 Replaced New
Oracle10i tool kit */
/* int sums[NDISTS];*/
#endif TSL
! int sums[NDISTS];
#else
! float sums[NDISTS];
#endif
/* Replaced end */
#endif

#ifdef USE_IEEE_NUMBER
float sums[NDISTS];
#else
int sums[NDISTS];
#endif
/* Replaced end */

    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;
    OCISmt *curp1;
    OCISmt *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;

```

```

#ifdef DMLRETEL
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;
#endif

/* ----- */
/* NewOrder Struct */
/* ----- */
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];

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! int s_remote[NITEMS];
#endif

#ifdef USE_IEEE_NUMBER
float s_remote[NITEMS];
#else
int s_remote[NITEMS];
#endif
/* Replaced end */

    char s_dist_info[NITEMS][25];
    OCISmt *curm1;
    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;
    OCISmt *curm2;
    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;

/* Replaced 03.05.15 Argument error
(OCIBNDPL). */
#if 0
! sb2 w_id_len;
#endif
ub2 w_id_len;
/* Replaced end */
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;

/* ----- */
/* OrderStatus Struct */
/* ----- */
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;

    OCISmt *curo0;
    OCISmt *curo1;
    OCISmt *curo2;
    OCISmt *curo3;
    OCISmt *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;
/* Deleted T.Kato 2004.12.21 New Oracle10g
tool kit */
/* OCIBind *o_rowid_bp;*/
/* Deleted end */

    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];
/* Deleted T.Kato 2004.12.21 New Oracle10g
tool kit */
/* OCIDefine *o_rowid_dp[2];*/
/* Deleted end */
OCIDefine *o_id_dp[2];
OCIDefine *o_entry_d_dp[2];
OCIDefine *o_cr_id_dp[2];
OCIDefine *o_ol_cnt_dp[2];
OCIDefine *o_d_d_dp;
OCIDefine *o_i_id_dp;
OCIDefine *o_supply_w_id_dp;
OCIDefine *o_quantity_dp;
OCIDefine *o_amount_dp;
OCIDefine *o_d_base_dp;
OCIDefine *c_count_dp;
OCIRowid *c_rowid_ptr[100];
OCIRowid *c_rowid_cust;
/* Deleted T.Kato 2004.12.21 New Oracle10g
tool kit */
/* OCIRowid *o_rowid;*/
/* Deleted end */
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;

/* ----- */
/* Payment Struct */
/* ----- */
struct payctx {
OCIStmt *curpi;
OCIStmt *curp0;
OCIStmt *curp1;
OCIBind *w_id_bp[2];
ub2 w_id_len;

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;

/* ----- */
/* StockLevel Struct */
/* ----- */
struct stockx {
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 stockx stockx;

.....
svrap/bs-del.c
.....

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

#if defined(__cplusplus)
extern "C" {
#endif
extern int _tmrunserver _((int));
//extern void OPSTUXSERVER _((TPSVCINFO
*));
//extern void TPCC _((TPSVCINFO *));
extern void DELIVERY _((TPSVCINFO *));
#if defined(__cplusplus)
}
#endif

//static struct tmdspchtbl_t _tmdspchtbl[] = {
// { "OPSTUXSERVER", "OPSTUXSERVER",
(void (*) _((TPSVCINFO *)) OPSTUXSERVER,
0, 0 },
// { NULL, NULL, NULL, 0, 0 }
//};
static struct tmdspchtbl_t _tmdspchtbl[] = {
{ "DELIVERY", "DELIVERY", (void (*)
_((TPSVCINFO *)) DELIVERY, 0, 0 },
{ NULL, NULL, NULL, 0, 0 }
};

#ifdef _TMDLLIMPORT
#define _TMDLLIMPORT
#endif

_TMDLLIMPORT extern struct xa_switch_t
tmnull_switch;

struct tmsvargs_t tmsvargs = {
NULL,
&_tmdspchtbl[0],
0,
tpsvrinit,
tpsvrdone,

```



```

    _tmrunserver, /* PRIVATE */
    NULL, /* RESERVED */
    NULL, /* RESERVED */
    NULL, /* RESERVED */
    NULL /* RESERVED */
};

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

.....
svrapl/bs-new.c
.....

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

#if defined(__cplusplus)
extern "C" {
#endif
extern int _tmrunserver (int);
//extern void OPSTUXSERVER _((TPSVCINFO
*));
//extern void TPCC _((TPSVCINFO *));
extern void NEWORDER (TPSVCINFO *);

#if defined(__cplusplus)
}
#endif

//static struct tmdspchtbl_t _tmdspchtbl[] = {
// { "OPSTUXSERVER", "OPSTUXSERVER",
(void *) _((TPSVCINFO *)) OPSTUXSERVER,
0, 0 },
// { NULL, NULL, NULL, 0, 0 }
//};
static struct tmdspchtbl_t _tmdspchtbl[] = {
{ "NEWORDER", "NEWORDER", (void *)
_((TPSVCINFO *)) NEWORDER, 0, 0 },
{ NULL, NULL, NULL, 0, 0 }
};

#ifdef _TMDLLIMPORT
#define _TMDLLIMPORT
#endif

```

```

_TMDLLIMPORT extern struct xa_switch_t
tnull_switch;

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

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

.....
svrapl/bs-ord.c
.....

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

#if defined(__cplusplus)
extern "C" {
#endif
extern int _tmrunserver _((int));
//extern void OPSTUXSERVER _((TPSVCINFO
*));
//extern void TPCC _((TPSVCINFO *));
extern void ORDERSTATUS _((TPSVCINFO *));
#if defined(__cplusplus)
}
#endif

//static struct tmdspchtbl_t _tmdspchtbl[] = {
// { "OPSTUXSERVER", "OPSTUXSERVER",
(void *) _((TPSVCINFO *)) OPSTUXSERVER,
0, 0 },
// { NULL, NULL, NULL, 0, 0 }
//};
static struct tmdspchtbl_t _tmdspchtbl[] = {
{ "ORDERSTATUS", "ORDERSTATUS", (void *)
_((TPSVCINFO *)) ORDERSTATUS, 0, 0 },
{ NULL, NULL, NULL, 0, 0 }
};

```

```

};

#ifdef _TMDLLIMPORT
#define _TMDLLIMPORT
#endif

_TMDLLIMPORT extern struct xa_switch_t
tnull_switch;

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

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

.....
svrapl/bs-pay.c
.....

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

#if defined(__cplusplus)
extern "C" {
#endif
extern int _tmrunserver _((int));
//extern void OPSTUXSERVER _((TPSVCINFO
*));
//extern void TPCC _((TPSVCINFO *));
extern void PAYMENT _((TPSVCINFO *));
#if defined(__cplusplus)
}
#endif

//static struct tmdspchtbl_t _tmdspchtbl[] = {
// { "OPSTUXSERVER", "OPSTUXSERVER",
(void *) _((TPSVCINFO *)) OPSTUXSERVER,
0, 0 },

```

```

// { NULL, NULL, NULL, 0, 0 }
//};
static struct tmdspchtbl_t _tmdspchtbl[] = {
    { "PAYMENT", "PAYMENT", (void *)
      _((TPSVCINFO *)) PAYMENT, 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,
    &_tmdspchtbl[0],
    0,
    tpsvrinit,
    tpsvrdone,
    _tmrunserver, /* PRIVATE */
    NULL, /* RESERVED */
    NULL, /* RESERVED */
    NULL, /* RESERVED */
    NULL /* RESERVED */
};

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

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

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

.....
svrapl/bs-sto.c
.....

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

#ifdef __cplusplus
extern "C" {
#endif
extern int _tmrunserver _((int));
//extern void OPSTUXSERVER _((TPSVCINFO
*));
//extern void TPCC _((TPSVCINFO *));
extern void STOCKLVL _((TPSVCINFO *));
#ifdef __cplusplus
}

```

```

#endif

//static struct tmdspchtbl_t _tmdspchtbl[] = {
// { "OPSTUXSERVER", "OPSTUXSERVER",
// (void *) _((TPSVCINFO *)) OPSTUXSERVER,
// 0, 0 },
// { NULL, NULL, NULL, 0, 0 }
//};

static struct tmdspchtbl_t _tmdspchtbl[] = {
    { "STOCKLVL", "STOCKLVL", (void *)
      _((TPSVCINFO *)) STOCKLVL, 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,
    &_tmdspchtbl[0],
    0,
    tpsvrinit,
    tpsvrdone,
    _tmrunserver, /* PRIVATE */
    NULL, /* RESERVED */
    NULL, /* RESERVED */
    NULL, /* RESERVED */
    NULL /* RESERVED */
};

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

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

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

.....
svrapl/bs-whb.c
.....

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

#include <string.h>
#include "forlinux.h"
#include "log.h"

```

```

#ifdef __cplusplus
extern "C" {
#endif
extern int _tmrunserver _((int));
extern void OPSTUXSERVER _((TPSVCINFO
*));
#ifdef __cplusplus
}
#endif

static struct tmdspchtbl_t _tmdspchtbl[] = {
    { "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,
    &_tmdspchtbl[0],
    0,
    tpsvrinit,
    tpsvrdone,
    _tmrunserver, /* PRIVATE */
    NULL, /* RESERVED */
    NULL, /* RESERVED */
    NULL, /* RESERVED */
    NULL /* RESERVED */
};

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

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

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

.....
svrapl/initsvrconfig.c
.....

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

```

```

* TPC-C Client Application Program Source
*
* Entry Functions
* (1) GetConfigFileInfo
*
* CREATE by TSL 2003.12.19
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *
*****/
#include "forlinux.h"
#include <unistd.h>
#include "tpcc.h"
#include "tpcc_info.h"
#include "GlobalArea.h"
#include "log.h"
#include "sema.h"
#include "prototype.h"
#include "shm.h"
#include "SampleInfo.h"
/* Global area for sampling. */
MAC_SampleGlobalArea;
/*****
* Get configuration file information.
*
* Return Value
* None
*****/
int GetConfFileInfo_GetStr(char* section_name,
char* key_name, char* str);

void GetConfFileInfo()
{
    /* Check INI file exist */
    if (access(GLB_ConfigFilePath, 0x00) != 0) {
        /* INI file no exist, using default value */
        TpcUserLog(LOG_LCK, "INI file nothing,
using default value");
        strcpy(GLB_LogFilePath,
DEFAULT_SVRAPL_LOG_PATH);
        return;
    }

    /* Get execution informations
*/
    /* If undefined key and illegal value, using
default value */
    if (GetConfFileInfo_GetStr("SVRAPL_INFO",
"LogPath", GLB_LogFilePath) != 0) {
        strcpy(GLB_LogFilePath,
DEFAULT_SVRAPL_LOG_PATH);
    }
}
/*****
/* Get information in the CONFIG file for string
value */
/*****
int GetConfFileInfo_GetStr(char* section_name,
char* key_name, char* str) {

    int i;
    char value_buf[1024];

    for (i = 0; i < 3; i++) {
        GetPrivateProfileString(section_name,
key_name, "",

```

```

value_buf, sizeof(value_buf),
GLB_ConfigFilePath);
    if (value_buf[0] == "") {
        /* if Key is nothing, retry getting */
        continue;
    }
    break;
}
#endif
PUT_INF_LOG
TpcUserLog(LOG_LCK, "CONFIG file
information [%s %s]=[%s]", section_name,
key_name, value_buf);
#endif
if (value_buf[0] == "") {
    /* Target key was nothing */
    return (-1);
}
strcpy(str, value_buf);
return(strlen(value_buf));
}

/*****
* Initialize configuration information
*
* Return Value
* none.
*****/

void InitSvrConfig(char* path) {

    char work_path[MAX_PATH];
    int i;

    /* Initialize share memory for sampling of
svrapl */
    MAC_SampleInitParent;

    /* Get configuration information (set to global
area) */
    strcpy(GLB_ConfigFilePath, path);

    /* Set default log path */
    strcpy(GLB_LogFilePath,
DEFAULT_SVRAPL_LOG_PATH);

    GetConfFileInfo();

    TpcUserLog(LOG_LCK, "InitSvrConfig start
\n");

    /* Initialize SVRAPL semaphore for log */
    strcpy(work_path, GLB_LogFilePath);
    for(i = strlen(work_path) - 1; i > 0 &&
work_path[i] != '\'; i--);
    work_path[i] = '\0';

    if ((GLB_LogSemId = InitSem(work_path,
SEM_SVRAPL_PROJID)) == -1) {
        TpcUserLog(LOG_LCK, "InitSem() faile for
SvrApl log\n");
        return;
    }

    return;
}

/*****
svrapl/log_level.h
/*****

```

```

*
* TPC-C Client Application Program Source
*
* CREATE by TSL 2003.02.07
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****/

#define PUT_INF_LOG //
Information log
#define PUT_FUNC_ENTRY_LOG //
Function entry point log
//define PUT_FUNC_EXIT_LOG //
Function exit log

/* Function entry point log macro */
#ifdef PUT_FUNC_ENTRY_LOG
#define MAC_PutFuncEntryLog(func)
TpcUserLog(LOG_INF, ">>>>> "func" start
>>>>>");
#else
#define MAC_PutFuncEntryLog(func) ;
#endif

/* Function exit point log */
#ifdef PUT_FUNC_EXIT_LOG
#define MAC_PutFuncExitLog(func)
TpcUserLog(LOG_INF, "<<<<< "func" end
<<<<<");
#else
#define MAC_PutFuncExitLog(func) ;
#endif

.....:
svrapl/prototype.h
.....:

/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* Function prototype definition.
*
* CREATE by TSL 2003.12.11
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003-2004 *
*****/

#include "tpccflags.h"

/* ..... */
/* Prototype */
/* ..... */

#ifdef DEL_ORA8I
int tkvcldinit ();
int tkvcninit ();
int tkvcoint ();
int tkvcpinit (void);
int tkvcsinit ();
int tkvcd ();

```

```

int tkvcn ();
int tkvcs ();
int tkvcp ();
int tkvco ();
void tkvcddone ();
void tkvcndone ();
void tkvcsdone ();
void tkvcpdone ();
void tkvcodone ();
#else
int tkvcinit (int plsqli);
int tkvcninit ();
int tkvcinit ();
int tkvcpinit (void);
int tkvcsinit ();
int tkvcd (int plsqli);
int tkvcn ();
int tkvcs ();
int tkvcp ();
int tkvco ();
void tkvcddone (int plsqli);
void tkvcndone ();
void tkvcsdone ();
void tkvcpdone ();
void tkvcodone ();
#endif

/* pldel */
void shiftdata(int from);

/* tpccpl Prototype */
int TPCinit (int id, char* uid, char* pwd);
int TPCnew (struct newstruct* str);
int TPCdel (struct delstruct* str);
int TPCpay (struct paystruct* str);
int TPCord (struct ordstruct* str);
int TPCsto (struct stostruct* str);
void TPCexit (void);

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

#ifndef AVOID_DEADLOCK
/* Added T.Kato 02.11.22 */
void swap_item(struct newstruct *str, int i, int j);
void q_sort_item(int *arr, struct newstruct *str, int
left, int right);
/* Added End */
void swap(struct newstruct *str, int i, int j);
void q_sort(int *arr, struct newstruct *str, int left,
int right);
#endif

/* Added Hayashi 03.12.24 */
void InitSvrConfig(char *);
int GetPrivateProfileString(char* section_name,
char* key_name,
char* default_str, char*
key_data,
int buf_size, char*
file_name);
/* Added End */

.....
svrapl/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
*/

#include "log.h"

/* Deleted 03.05.19 No use. */
#if 0
!typedef struct cda_def csrdef;
!typedef struct cda_def ldedef;
#endif
/* Deleted end */

/* TPC-C transaction functions */

/* 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

/* Modified by TSL --- BEGIN ---2006.03.17 */
/* #define FULLDATE "dd-mon-yy.hh24:mi:ss" */

#define FULLDATE "dd-mm-yyyy.hh24:mi:ss"
/* Modified by TSL --- END ---2006.03.17 */

#define SHORTDATE "dd-mm-yyyy"

#define DELRT 80.0

/* Deleted 03.05.19 No use. */
#if 0
extern int tkvcss (); /* for alter session to get
memory size and trace */
extern boolean multitrans;
#endif
/* Deleted end */
/* Deleted 03.05.16 For warning */
#if 0
extern int ord_init;
#endif
/* Deleted end */

/* Deleted 03.05.19 No use. */
#if 0
extern void errprt ();
#endif
/* Deleted end */

/* Added T.Kato 2003.03.25 for debug */
extern void DbgLog(char* form_dat, int arg);
#ifndef DGLDEF
#define DBGLOG(format_data, arg)
TpcUserLog(LOG_INF,format_data, arg)
#else
#define DBGLOG(format_data, arg)
#endif

#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(LOG_FILE_LINE,(errp),(function));

#define OCIBND(stmp, bndp, errp, sqlvar, progvl, ftype)\
    ocierror(LOG_FILE_LINE,(errp), \

OCIHandleAlloc((stmp),(dvoid*)&(bndp),OCI_H
TYPE_BIND,0,(dvoid**0)); \
    ocierror(LOG_FILE_LINE, (errp), \
        OCIBindByName((stmp), &(bndp), (errp), \
            (text *) (sqlvar), strlen((sqlvar)), \
            (progvl), (progvl), \
            (ftype),0,0,0,0,OCI_DEFAULT));

/* bind arrays for sql */
#define
OCIBNDRA(stmp,bndp,errp,sqlvar,progvl,ftype,indp,alen,arcode) \
    DISCARD ocierror(LOG_FILE_LINE,(errp), \

OCIHandleAlloc((stmp),(dvoid*)&(bndp),OCI_H
TYPE_BIND,0,(dvoid**0)); \
    DISCARD ocierror(LOG_FILE_LINE,(errp), \

OCIBindByName((stmp),&(bndp),(errp),(text
*)(sqlvar),strlen((sqlvar)),\

(progvl),(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(LOG_FILE_LINE,(errp), \

OCIHandleAlloc((stmp),(dvoid*)&(bndp),OCI_H
TYPE_BIND,0,(dvoid**0)); \
    DISCARD ocierror(LOG_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(LOG_FILE_LINE,(errp), \

OCIBindDynamic((bndp),(errp),(ctxp),(cbf_nodat
a),(ctxp),(cbf_data));

/* bind in/out for plsql without indicator and rcode
*/
#define
OCIBNDPL(stmp,bndp,errp,sqlvar,progvl,ftype,alen) \
    DISCARD ocierror(LOG_FILE_LINE,(errp), \

OCIHandleAlloc((stmp),(dvoid*)&(bndp),OCI_H
TYPE_BIND,0,(dvoid**0)); \
    DISCARD ocierror(LOG_FILE_LINE,(errp), \

OCIBindByName((stmp),&(bndp),(errp),(CONST
text *) (sqlvar), \
    (sb4)strlen((CONST char *)
(sqlvar)),(void *) (progvl), \

(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,progvl,progvl
,ftype,indp,alen,arcode,\
    ms,cu) \
    ocierror(LOG_FILE_LINE, (errp), \

OCIHandleAlloc((stmp),(dvoid*)&(bndp),OCI_H
TYPE_BIND,0,(dvoid**0)); \
    ocierror(LOG_FILE_LINE,(errp),\

OCIBindByName((stmp),&(bndp),(errp),(text
*)(sqlvar),strlen((sqlvar)),\

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

#define
OCIDFNR(stmp,dfnp,errp,pos,progvl,progvl,ftype) \

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(pro
gv),(progvl),\

(ftype),NULL,NULL,NULL,OCI_DEFAULT); \

#define
OCIDFNRA(stmp,dfnp,errp,pos,progvl,progvl,ftype
,indp,alen,arcode) \

OCIHandleAlloc((stmp),(dvoid*)&(dfnp),OCI_H
TYPE_DEFINE,0,\
    (dvoid**0)); \

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(pro
gv),\
    (progvl),(ftype),(indp),(alen),\
    (arcode),OCI_DEFAULT);

#define
OCIDFNDYN(stmp,dfnp,errp,pos,progvl,progvl,ftype
,indp,ctxp,cbf_data) \
    ocierror(LOG_FILE_LINE,(errp), \

OCIHandleAlloc((stmp),(dvoid*)&(dfnp),OCI_H
TYPE_DEFINE,0,\
    (dvoid**0)); \
    ocierror(LOG_FILE_LINE,(errp), \

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(pro
gv),(progvl),(ftype),\
    (indp),NULL,NULL,
OCI_DYNAMIC_FETCH)); \
    ocierror(LOG_FILE_LINE,(errp), \

OCIDefineDynamic((dfnp),(errp),(ctxp),(cbf_data
)));

/* Deleted T.Kato 02.10.23 Overrapped
tpcc_info.h */
#if 0
/* New order */
!struct newinstruct {
! int d_id;
! int c_id;
! int ol_i_id[15];
! int ol_supply_w_id[15];
! int ol_quantity[15];
!};
!
!struct newoutstruct {
! int terror;
! int o_id;
! int o_ol_cnt;
! char c_last[17];
! char c_credit[3];

! float c_discount;
! float w_tax;
! float d_tax;
! char o_entry_d[20];
! float total_amount;
! char i_name[15][25];
! int s_quantity[15];
! char brand_generic[15];
! float i_price[15];
! float ol_amount[15];
! char status[26];

```

```

! int retry;
!};
!
!struct newstruct {
! struct newinstruct newin;
! struct newoutstruct newout;
!};
!
!
!/* Payment */
!
!struct payinstruct {
! int w_id;
! int d_id;
! int c_w_id;
! int c_d_id;
! int c_id;
! int bylastname;
! int h_amount;
! char c_last[17];
!};
!
!struct payoutstruct {
! int terror;
! char w_street_1[21];
! char w_street_2[21];
! char w_city[21];
! char w_state[3];
! char w_zip[10];
! char d_street_1[21];
! char d_street_2[21];
! char d_city[21];
! char d_state[3];
! char d_zip[10];
! int c_id;
! char c_first[17];
! char c_middle[3];
! char c_last[17];
! char c_street_1[21];
! char c_street_2[21];
! char c_city[21];
! char c_state[3];
! char c_zip[10];
! char c_phone[17];
! char c_since[11];
! char c_credit[3];
! double c_credit_lim;
! float c_discount;
! double c_balance;
! char c_data[201];
! char h_date[20];
! int retry;
!};
!
!struct paystruct {
! struct payinstruct payin;
! struct payoutstruct payout;
!};
!
!
!/* Order status */
!
!struct ordinstruc {
! int w_id;
! int d_id;
! int c_id;
! int bylastname;
! char c_last[17];
!};
!
!struct ordoutstruct {
! int terror;
! int c_id;
! char c_last[17];
! char c_first[17];

```

```

! char c_middle[3];
! double c_balance;
! int o_id;
! char o_entry_d[20];
! int o_carrier_id;
! int o_ol_cnt;
! int ol_supply_w_id[15];
! int ol_i_id[15];
! int ol_quantity[15];
! float ol_amount[15];
! char ol_delivery_d[15][11];
! int retry;
!};
!
!struct ordstruct {
! struct ordinstruc ordin;
! struct ordoutstruct ordout;
!};
!
!
!/* Delivery */
!
!struct delinstruc {
! int w_id;
! int o_carrier_id;
! double qtime;
! int in_timing_int;
! int plsqflag;
!};
!
!struct deloutstruct {
! int terror;
! int retry;
!};
!
!
!struct delstruct {
! struct delinstruc delin;
! struct deloutstruct delout;
!};
!
!
!/* Stock level */
!
!struct stoinstruct {
! int w_id;
! int d_id;
! int threshold;
!};
!
!struct stooutstruct {
! int terror;
! int low_stock;
! int retry;
!};
!
!
!struct stostruct {
! struct stoinstruct stoin;
! struct stooutstruct stoout;
!};
!
!#endif
!#endif

.....
svrapl/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
|

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

+=====+
=====+

#ifndef TPCC_INFO_H
#define TPCC_INFO_H

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

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

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

struct newstruct {
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;
    int h_amount;
    char c_last[17];
};

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

struct paystruct {
    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;
};

/* Replaced T.Kato 02.10.24 for TPAPL interface */
#if 0
! double qtime;
! int in_timing_int;
#endif

    long startsec;
    long startusec;
/* Replaced end */
};

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;
};

/* 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

.....
svrapl/tpccflags.h
.....

#define DMLRETDEL

/* Deleted T.Kato 2004.12.21 New Oracle10g
tool kit */
#if 0
#ifndef TSL
#define USE_IEEE_NUMBER
#endif
#endif

.....
svrapl/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>
#include <sys/time.h>
#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

#include <unistd.h>

#include "forlinux.h"
#include "tpcc.h"
#include "tpcc_info.h"
#include "httpext.h" //ISAPI DDL information
header
#include "tpccapi.h" //this dlls specific
structure, value e.t. header
#include "GlobalArea.h"

```

```

#include "prototype.h"
#include "sema.h"
#include "shmem.h"
#include "SampleInfo.h"

#ifdef TUX

#include <lmenv.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

#if 0
// Lifted from HP FDR since they did such a nice
job
void TMlog( char *format, ... )
{
    va_list args;
    char buf[4096];
    int len;
    va_start( args, format );
    _strtime( buf );
    strcat( buf, " ");
    len = strlen( buf );
    (void)_vsprintf( buf+ len, sizeof( buf ) - len - 1,
format, args);
    buf[sizeof( buf ) - 1]= '\0';
    va_end( args );
    userlog( buf );
}
#endif

/* 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]);

/* Set default log path */
strcpy(GLB_LogFilePath,
DEFAULT_SVRAPL_LOG_PATH);
TpccUserLog(LOG_LCK, "Start tpsvrinit");

/* Initialize semaphore and log. */
InitSvrConfig(TPCC_CONF_FILE);

#ifdef TUX

srand ( (unsigned)time( NULL ) );
svrcnt = rand();

/* send 6 for all txns to be init'd */
/* fix uid and pwd for now, pull out later */
/* not passing parameters through TUX yet
*/

#if 0 /* Replaced 2003/12/12 adjust
arguments */
! if (TPCinit( svrcnt, "tpcc", "tpcc", 6) ) {
#else
if (TPCinit( svrcnt, "tpcc", "tpcc" ) ) {
#endif
TpccUserLog(LOG_FILE_INF, " FAILED
to init all txns types");
return (-1);
}

TpccUserLog(LOG_INF, "Finished
TPCinit(tpsvrinit)");

return 0;

#else // ifdef TUX for topend

#if 0 /* Replaced 2003/12/12 adjust
arguments */
! if (TPCinit( svrid, uid, pwd, txntype) ) {
#else
if (TPCinit( svrid, uid, pwd) ) {
#endif
TpccUserLog(LOG_INF, "Failed in TPCinit
(probably connecting).");
exit (1);
}

TpccUserLog(LOG_INF, "Finished TPCinit");

return (1);
#endif

}

void tpsvrdone ()

{
TpccUserLog(LOG_INF, "Start tpsvrdone");

#if 0 /* Replaced 2003/12/12 adjust arguments */
! TPCexit (0);
#else
TPCexit ();
#endif

```

```

TpccUserLog(LOG_INF, "Finished
TPCexit(tpsvrdone)");
}

/* 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

MAC_SampleWork; // Sampling area

newinfo = (struct newstruct *) msg->data;

MAC_SampleStartTime; // Start sampling.
newinfo->retval = TPCnew (newinfo); // set
return value to 0 or -1
// Finish sampling.
MAC_SampleDBSrvResp(RspTimeNewOrder,
MaxRspTimeNewOrder,
SMaxRspTimeNewOrder, NumNewOrder);

// 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

    MAC_SampleWork; // Sampling area

    payinfo = (struct paystruct *) msg->data;
    MAC_SampleStartTime; // Start sampling.
    payinfo->retval = TPCpay (payinfo); // set
return value to 1 or 0 or -1
    // Finish sampling.
    MAC_SampleDBSrvResp(RspTimePayment,
MaxRspTimePayment, SMaxRspTimePayment,
NumPayment);

    // always return treturn success - let client
side poll retval for actual error
    treturn (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
*      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

```

```

*
*              -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

    MAC_SampleWork; // Sampling area

    ordinfo = (struct ordstruct *) msg->data;
    MAC_SampleStartTime; // Start sampling.
    ordinfo->retval = TPCord (ordinfo); // set
return value to 0 or -1
    // Finish sampling.

    MAC_SampleDBSrvResp(RspTimeOrderStatus,
MaxRspTimeOrderStatus,
SMaxRspTimeOrderStatus, NumOrderStatus);

    // always return treturn success - let client
side poll retval for actual error
    treturn (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;

    result = TPCdel(deldata);

    return result;

#else

    MAC_SampleWork; // Sampling area

    delinfo = (struct delstruct *) msg->data;

    MAC_SampleStartTime; // Start sampling.
    delinfo->retval = TPCdel (delinfo); // set return
value to 0 or -1
    MAC_SampleDBSrvRespDel(); // Finish
sampling.

    // always return treturn success - let client
side poll retval for actual error
    treturn (TPSUCCESS, 0, (char *) delinfo,
sizeof (struct delstruct), 0);

#endif

}

/* Replaced T.kato 02.10.28 old version name
used */
#if 0
/* FUNCTION: int STOCKLEVEL(CLIENTDATA
*jobData, StockLevelData *stodata, int
deadlock)*/
#endif
/* FUNCTION: int STOCKLVL(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
*/

/* Replaced T.kato 02.10.28 old vaersion name
used */
#if 0
#ifdef TOP
int STOCKLEVEL(CLIENTDATA *jobData,
StockLevelData *stodata, int deadlock)
#else

```

```

Ivoid STOCKLEVEL (TPSVCINFO *msg)
#endif
#endif

#ifndef TOP
int STOCKLVL(CLIENTDATA *jobData,
StockLevelData *stodata, int deadlock)
#else
void STOCKLVL (TPSVCINFO *msg)
#endif
/* Replaced end */

{

#ifdef TOP

    int result;

    result = TPCsto(stodata);

    return result;

#else

    MAC_SampleWork; // Sampling area

    stoinfo = (struct stostruct *) msg->data;
    MAC_SampleStartTime; // Start sampling.
    stoinfo->retval = TPCsto (stoinfo); // set return
value to 0 or -1
    // Finish sampling
    MAC_SampleDBSrvResp(RspTimeStockLevel,
MaxRspTimeStockLevel,
SMaxRspTimeStockLevel, NumStockLevel);

    // always return treturn success - let client
side poll retval for actual error
    treturn (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)
#else
void OPSTUXSERVER (TPSVCINFO *msg)
#endif

```

```

{
#ifdef TOP
    int result;

    result = TPCnew(neword);

    return result;

#else
    // for Tuxedo
    /* Replaced T.Kato 03.03.19 Ununique
STRUCTURE size between Derivery and
StockLevel */
    #if 0
    ! if (msg->len == 928) { // len for neworder
    !     newinfo = (struct newstruct *) msg->data;
    !     newinfo->retval = TPCnew (newinfo); // set
return value to 0 or -1
    !
    ! // always return treturn success - let client
side poll retval for actual error
    !     treturn (TPSUCCESS, 0, (char *) newinfo,
sizeof (struct newstruct), 0);
    ! }
    ! else
    !     if (msg->len == 616) { // len for payment
    !         payinfo = (struct paystruct *) msg->data;
    !         payinfo->retval = TPCpay (payinfo); // set
return value to 1 or 0 or -1
    !
    ! // always return treturn success - let client
side poll retval for actual error
    !         treturn (TPSUCCESS, 0, (char *) payinfo,
sizeof (struct paystruct), 0);
    !     }
    !     else
    !         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 treturn success - let client
side poll retval for actual error
    !             treturn (TPSUCCESS, 0, (char *) ordinfo,
sizeof (struct ordstruct), 0);
    !         }
    !         else
    !             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 treturn success
- let client side poll retval for actual error
    !                 treturn (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 treturn success - let
client side poll retval for actual error
    !                 treturn (TPSUCCESS, 0, (char *)
stoinfo, sizeof (struct stostruct), 0);
    !             }
    !         }
#endif
}

```

```

int trx_type = *(int*)msg->data;

MAC_SampleWork; // Sampling area

if (trx_type == 1) { // type for neworder
    newinfo = (struct newstruct *) msg->data;

    DBGLOG("OPS:[New]Start",0);
    MAC_SampleStartTime; // Sampling start
    newinfo->retval = TPCnew (newinfo); // set
return value to 0 or -1

    MAC_SampleDBSrvResp(RspTimeNewOrder,
MaxRspTimeNewOrder,
SMaxRspTimeNewOrder, NumNewOrder); //
Sampling finish
    DBGLOG("OPS:[New]End >%d",newinfo-
>retval);

    // always return treturn success - let client
side poll retval for actual error

    treturn (TPSUCCESS, 0, (char *) newinfo,
sizeof (struct newstruct), 0);
}
else
if (trx_type == 2) { // type for payment
    payinfo = (struct paystruct *) msg->data;
    DBGLOG("OPS:[Pay]Start",0);
    MAC_SampleStartTime; // Sampling start
    payinfo->retval = TPCpay (payinfo); // set
return value to 1 or 0 or -1

    MAC_SampleDBSrvResp(RspTimePayment,
MaxRspTimePayment, SMaxRspTimePayment,
NumPayment); // Sampling finish
    DBGLOG("OPS:[Pay]End >%d",payinfo-
>retval);

    // always return treturn success - let client
side poll retval for actual error
    treturn (TPSUCCESS, 0, (char *) payinfo,
sizeof (struct paystruct), 0);
}
else
if (trx_type == 3) { // type for order status
    ordinfo = (struct ordstruct *) msg->data;
    DBGLOG("OPS:[Ord]Start",0);
    MAC_SampleStartTime; // Sampling start
    ordinfo->retval = TPCord (ordinfo); // set
return value to 0 or -1

    MAC_SampleDBSrvResp(RspTimeOrderStatus,
MaxRspTimeOrderStatus,
SMaxRspTimeOrderStatus, NumOrderStatus); //
Sampling finish
    DBGLOG("OPS:[Ord]End >%d",ordinfo-
>retval);

    // always return treturn success - let client
side poll retval for actual error
    treturn (TPSUCCESS, 0, (char *) ordinfo,
sizeof (struct ordstruct), 0);
}
else
if (trx_type == 4) { // type for delivery
    delinfo = (struct delstruct *) msg->data;
    DBGLOG("OPS:[Del]Start",0);
    MAC_SampleStartTime; // Start sampling.
    delinfo->retval = TPCdel (delinfo); // set
return value to 0 or -1

```

```

    MAC_SampleDBSrvRespDel(); // Finish
sampling.
    DBGLOG("OPS:[Del]End >%d",delinfo-
>retval);

    // 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;
    DBGLOG("OPS:[Sto]Start",0);
    MAC_SampleStartTime; // Start sampling.
    stoinfo->retval = TPCsto (stoinfo); // set
return value to 0 or -1

MAC_SampleDBSrvResp(RspTimeStockLevel,
MaxRspTimeStockLevel,
SMaxRspTimeStockLevel, NumStockLevel); //
Finish sampling
    DBGLOG("OPS:[Sto]End >%d",stoinfo-
>retval);

    // always return tpreturn success - let client
side poll retval for actual error
    tpreturn (TPSUCCESS, 0, (char *) stoinfo,
sizeof (struct stostruct), 0);
}
/* Replaced end */

#endif
}

.....
svrapl/3tier/MakeShell
.....

#!/bin/sh
cd /home/tpc/client_apl/svrapl/3tier
make > make_result.txt 2>&1

.....
svrapl/3tier/Makefile
.....

#-----
-----
# Makefile : Makefile for 3 tier library on Linux.
#
# Created by TSL 2003.12.17
#
# All Right Rserverd, Copyright Co, FUJITSU
LIMITED 2003-2004.
#-----
-----

# GCC compile configurations
AR = ar
ARFLAGS = rv
CFLAGS = -Wall -O2
CC = gcc

# MACRO definition
#DMACRO = -DTSL -DPLSQLFLAG=1 -DTUX
DMACRO = -DPLSQLFLAG=1 -DTUX

# home directory.
ORADIR = /usr/local/oracle
TUXDIR = /usr/local/BEA/tuxedo8.1
SVRDIR = /home/tpc/client_apl/svrapl
COMDIR = /home/tpc/client_apl/common

```

```

# include directory
ORA_INC = -$(ORADIR)/rdbms/demo -
I$(ORADIR)/rdbms/public
COM_INC = -$(COMDIR)
SRV_COM_INC = -$(SVRDIR)
TUX_INC = -$(TUXDIR)/include
INCLUDE = $(COM_INC) $(ORA_INC)
$(TUX_INC) $(SRV_COM_INC)
SVRDIR = /home/tpc/client_apl/svrapl

# depend on include file.
INCFILE = $(SVRDIR)/tpcc.h
$(SVRDIR)/GlobalArea.h $(SVRDIR)/prototype.h
\
$(SVRDIR)/tpccflags.h
$(SVRDIR)/tpcc_info.h $(SVRDIR)/TrnCntrlInfo.h
$(SVRDIR)/tpcc_info.h \
$(COMDIR)/log.h $(COMDIR)/sema.h
$(COMDIR)/forlinux.h

# target object
TIER_OBJS = pldel.o plnew.o plord.o
plpay.o plsto.o tpccpl.o
TIER_ARCH_LIB = libtier.a

$(TIER_ARCH_LIB) : $(TIER_OBJS)
$(INCFILE)
$(AR) $(ARFLAGS) $(TIER_ARCH_LIB)
$(TIER_OBJS)

.SUFFIXES: .o .c
.c.o:
$(CC) -o $@ -c $(CFLAGS) $(INCLUDE)
$(DMACRO) <

$(TIER_OBJS) : $(INCFILE)
$(TIER_OBJS) : Makefile

clean:
rm $(TIER_ARCH_LIB) $(TIER_OBJS)

.....
svrapl/3tier/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 "forlinux.h"
#include "log.h"
#include "tpcc.h"
#include "GlobalArea.h"
#include "prototype.h"

#define SQLTXT "BEGIN inittpc.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 = :c_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

#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);
}
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

#ifdef
int tkvcddinit (int plsqliflag)
    {
        text stmbuff[SQL_BUF_SIZE];

        if (plsqliflag)
        {
            pldctx = (pldelctx *) malloc (sizeof(pldelctx));
            DISCARD
            memset(pldctx, (char)0, (ub4)sizeof(pldelctx));
            /* Initialize */
            DISCARD OCCHandleAlloc(tpcenv,
(dvoid **)&pldctx->curp1, OCI_HTYPE_STMT, 0,
(dvoid **)0);
            DISCARD sprintf ((char *) stmbuff, SQLTXT);

```

```

            DISCARD OCISmtPrepare(pldctx->curp1,
errhp, stmbuff,
                (ub4) strlen((char *)stmbuff),
                OCI_NTV_SYNTAX,
                OCI_DEFAULT);
            DISCARD OCIERROR(errhp,
                OCISmtExecute(tpcscv, pldctx-
>curp1, errhp, 1, 0, NULLP(OCISnapshot),
                NULLP(OCISnapshot),
                OCI_DEFAULT));

            DISCARD OCCHandleAlloc(tpcenv, (dvoid **)
&pldctx->curp2, OCI_HTYPE_STMT,
                0, (dvoid **)0);
#ifdef (ISO5) || defined(ISO6) ||
defined(ISO8)
            #if defined(ISO5)
                sqlfile("../blocks/tkvcpdel_iso5.sql", stmbuff);
            #endif
            #if defined(ISO6)
                sqlfile("../blocks/tkvcpdel_iso6.sql", stmbuff);
            #endif
            #if defined(ISO8)
                sqlfile("../blocks/tkvcpdel_iso8.sql", stmbuff);
            #endif
            #else
                /* Replaced 04.01.20 TUXEDO Client */
            #if 0
                ! sqlfile("../blocks/tkvcpdel.sql", stmbuff);
            #endif

                sqlfile("../home/tpc/blocks/tkvcpdel.sql", stmbuff);
                /* Replaced end */
            #endif
            DISCARD OCISmtPrepare(pldctx->curp2,
errhp, stmbuff,
                (ub4)strlen((char *)stmbuff),
                OCI_NTV_SYNTAX, OCI_DEFAULT);
            OCIBNDPL(pldctx->curp2, pldctx->w_id_bp ,
errhp, "w_id",
                ADR(w_id), SIZ(int),
                SOLT_INT, &pldctx->w_id_len);
            OCIBNDPL(pldctx->curp2, pldctx->ordcnt_bp ,
errhp, "ordcnt",
                ADR(pldctx->ordcnt), SIZ(int),
                SOLT_INT, &pldctx->ordcnt_len);

            /* Replaced T.kato 03.07.18 New Oracle10i tool
kit */
            #if 0
            ! OCIBNDPL(pldctx->curp2, pldctx-
>del_date_bp, errhp, "now",
            ! dctx->del_date, SIZ(OCIDate),
            SOLT_ODT, &pldctx->del_date_len);
            #endif

#ifdef TSL
            OCIBNDPL(pldctx->curp2, pldctx-
>del_date_bp, errhp, "now",
                dctx->del_date, SIZ(OCIDate),
                SOLT_ODT, &pldctx->del_date_len);
            #else
            OCIBNDPL(pldctx->curp2, pldctx-
>del_date_bp, errhp, "now",
                ADR(pldctx->del_date), SIZ(OCIDate),
                SOLT_ODT, &pldctx->del_date_len);
            #endif
            /* Replaced end */

            OCIBNDPL(pldctx->curp2, pldctx-
>carrier_id_bp , errhp,
                "carrier_id", ADR(o_carrier_id), SIZ(int),

```

```

                SOLT_INT, &pldctx->carrier_id_len);
            OCIBNDPL(pldctx->curp2, pldctx->d_id_bp,
errhp, "d_id",
                pldctx->del_d_id, SIZ(int), SOLT_INT,
                pldctx->del_d_id_len,
                NDISTS, &pldctx->del_d_id_rcnt);
            OCIBNDPL(pldctx->curp2, pldctx->o_id_bp,
errhp, "order_id",
                pldctx->del_o_id, SIZ(int), SOLT_INT,
                pldctx->del_o_id_len, NDISTS,
                &pldctx->del_o_id_rcnt);

            /* Replaced T.kato 03.09.09 Oracle10g tool kit */
            #if 0
            /* Replaced T.kato 03.07.18 New Oracle10i tool
kit */
            !#if 0
            ! OCIBNDPL(pldctx->curp2, pldctx-
>sums_bp, errhp, "sums",
            ! pldctx->sums, SIZ(int), SOLT_INT,
            pldctx->sums_len, NDISTS,
            ! &pldctx->sums_rcnt);
            !#endif
            !
            !#ifdef TSL
            ! OCIBNDPL(pldctx->curp2, pldctx-
>sums_bp, errhp, "sums",
            ! pldctx->sums, SIZ(int), SOLT_INT,
            pldctx->sums_len, NDISTS,
            ! &pldctx->sums_rcnt);
            !#else
            ! OCIBNDPL(pldctx->curp2, pldctx-
>sums_bp, errhp, "sums",
            ! pldctx-
>sums, SIZ(float), SOLT_BFLOAT, pldctx-
>sums_len, NDISTS,
            ! &pldctx->sums_rcnt);
            !#endif
            /* Replaced end */
            #endif

            #ifdef USE_IEEE_NUMBER
            OCIBNDPL(pldctx->curp2, pldctx-
>sums_bp, errhp, "sums",
                pldctx-
>sums, SIZ(float), SOLT_BFLOAT, pldctx-
>sums_len, NDISTS,
                &pldctx->sums_rcnt);
            #else
            OCIBNDPL(pldctx->curp2, pldctx-
>sums_bp, errhp, "sums",
                pldctx->sums, SIZ(int), SOLT_INT,
            pldctx->sums_len, NDISTS,
                &pldctx->sums_rcnt);
            #endif
            /* Replaced end */

            OCIBNDPL(pldctx->curp2, pldctx-
>o_c_id_bp, errhp, "o_c_id",
                pldctx->o_c_id, SIZ(int), SOLT_INT,
            pldctx->o_c_id_len, NDISTS,
                &pldctx->o_c_id_rcnt);
            OCIBNDPL(pldctx->curp2, pldctx->retry_bp ,
errhp, "retry",
                ADR(pldctx->retry), SIZ(int), SOLT_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 OCISmtPrepare(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 OCISmtPrepare(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, OCI_HTYPE_STMT, 0,
(dvoid**)0);
DISCARD sprintf ((char *) stmbuf, SQLTXT4);
DISCARD OCISmtPrepare(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, ":o_l_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 OCISmtPrepare(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(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];
}
}

int tkvcd (int plsqflag)
{
/*int i, j;
int i;
/*int rpc,rcount,count;
int rpc,rcount;
int invalid;

if (plsqflag)
{

```

```

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,
OCISmtExecute(tpcsvc,pldctx-
>curp2,errhp,1,0,NULLP(CONST OCISnapshot),
NULLP(OCISnapshot),OCI_DEFAULT));
for (i = 0; i < NDISTS; i++)
{
del_o_id[i] = 0;
}
for (i = 0; (unsigned int)i < pldctx-
>del_o_id_rcnt; i++)
del_o_id[pldctx->del_o_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=OCISmtExecute(tpcsvc,dctx-
>curd1,errhp,NDISTS,0,
NULLP(CONST
OCISnapshot),NULLP(OCISnapshot),OCI_DEF
AULT);

```

```

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_DEF
AULT);
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)
{
TpccUserLog (LOG_FILE_INF, "Error in
TPC-C server %d: %d rows selected, %d ords
updated\n",
proc_no, rpc, rcount);
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_DEF
AULT);
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) {
TpccUserLog
(LOG_FILE_INF, "Error in TPC-C
server %d: %d ordnrs updated, %d ordl
updated\n",
proc_no, rpc, rcount);
}
#endif

```

```

/* 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) {
TpccUserLog(LOG_FILE_INF, "Error in
TPC-C server %d: %d rows selected, %d cust
updated\n",
proc_no, rpc, rcount);

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 plsqflag)
{
if (plsqflag)
{
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);
  }
}

.....
svrapl/3tier/plnew.c
.....

#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 "forlinux.h"
#include "log.h"

#ifdef ORA_TPCC
# define ORA_TPCC
# include "tpcc.h"
#endif

#include "GlobalArea.h"
#include "prototype.h"

#define SQLTXT2 "BEGIN
inittpcc.init_no(idx1arr); END;"

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

```

```

int tkvcninit ()
{
  /* for warning */
  /* int i;*/

  /* Replaced T.Kato 03.03.19 Repaleced Oracle
  10i tool kit */
  /* text stmbuf[16*1024];*/
  text stmbuf[32*1024];
  /* Replaced end */

  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);
  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,(dvoi
d**))(&nctx->cur1),
    OCI_HTYPE_STMT, 0, (dvoid**)0);
  /* Replaced T.kato 03.03.19 Replaced Oracle
  10i tool kit */
  /* sqlfile("../blocks/tkvcnew.sql",stmbuf);*/
  #if defined(ISO)
  sqlfile("../blocks/tkvcnew_iso.sql",stmbuf);
  #else
  #if defined(ISO7)
  sqlfile("../blocks/tkvcnew_iso7.sql",stmbuf);
  #else
  /* Replaced 04.01.20 TUXEDO Client */
  #if 0
  ! sqlfile("../blocks/tkvcnew.sql",stmbuf);
  #endif
  sqlfile("/home/tpc/blocks/tkvcnew.sql",stmbuf);
  /* Replaced end */
  #endif
  #endif
  /* Replaced end */

  DISCARD
  OCIERROR(errhp,OCIStmtPrepare(nctx->cur1,
errhp, stmbuf,
    strlen((char *)stmbuf),
  OCI_NTV_SYNTAX, OCI_DEFAULT));

  /* bind variables */

  OCIBNDPL(nctx->cur1, nctx->w_id_bp, errhp,
"w_id",ADR(w_id),SIZ(w_id),
  SOLT_INT, &nctx->w_id_len);
  OCIBNDPL(nctx->cur1, nctx->d_id_bp, errhp,
"d_id",ADR(d_id),SIZ(d_id),
  SOLT_INT, &nctx->d_id_len);
  OCIBNDPL(nctx->cur1, nctx->c_id_bp, errhp,
"c_id",ADR(c_id),SIZ(c_id),
  SOLT_INT, &nctx->c_id_len);
  OCIBNDPL(nctx->cur1, nctx->o_all_local_bp,
errhp, "o_all_local",

```

```

ADR(o_all_local),
SIZ(o_all_local),SOLT_INT, &nctx-
>o_all_local_len);
  OCIBNDPL(nctx->cur1, nctx->o_ol_cnt_bp,
errhp, "o_ol_cnt",ADR(o_ol_cnt),
  SIZ(o_ol_cnt),SOLT_INT, &nctx-
>o_ol_cnt_len);
  OCIBNDPL(nctx->cur1, nctx->w_tax_bp,
errhp, "w_tax",ADR(w_tax),SIZ(w_tax),
  SOLT_FLT, &nctx->w_tax_len);
  OCIBNDPL(nctx->cur1, nctx->d_tax_bp, errhp,
"d_tax",ADR(d_tax),SIZ(d_tax),
  SOLT_FLT, &nctx->d_tax_len);
  OCIBNDPL(nctx->cur1, nctx->o_id_bp, errhp,
"o_id",ADR(o_id),SIZ(o_id),
  SOLT_INT, &nctx->o_id_len);
  OCIBNDPL(nctx->cur1, nctx->c_discount_bp,
errhp, "c_discount",
  ADR(c_discount),
  SIZ(c_discount),SOLT_FLT, &nctx-
>c_discount_len);
  OCIBNDPL(nctx->cur1, nctx->c_credit_bp,
errhp, "c_credit",c_credit,
  SIZ(c_credit),SOLT_CHR, &nctx-
>c_credit_len);
  OCIBNDPL(nctx->cur1, nctx->c_last_bp,
errhp, "c_last",c_last,SIZ(c_last),
  SOLT_STR, &nctx->c_last_len);
  OCIBNDPL(nctx->cur1, nctx->retries_bp,
errhp, "retry",ADR(retries),
  SIZ(retries),SOLT_INT, &nctx-
>retries_len);
  OCIBNDPL(nctx->cur1, nctx->cr_date_bp,
errhp, "cr_date",&cr_date,
  SIZ(OCIDate), SOLT_ODT, &nctx-
>cr_date_len);

  OCIBNDPLA(nctx->cur1, nctx-
>ol_i_id_bp,errhp,"ol_i_id",nol_i_id,
  SIZ(int), SOLT_INT, nctx-
>nol_i_id_len,NITEMS,&nctx->nol_i_count);
  OCIBNDPLA(nctx->cur1, nctx-
>ol_supply_w_id_bp, errhp, "ol_supply_w_id",
  nol_supply_w_id,SIZ(int),SOLT_INT,
  nctx->nol_supply_w_id_len,
  NITEMS, &nctx->nol_s_count);

  /* Replaced T.kato 03.09.09 Oracle10g tool kit */
  #if 0
  ! OCIBNDPLA(nctx->cur1, nctx-
>ol_quantity_bp,errhp,"ol_quantity",
  ! nol_quantity, SIZ(int),SOLT_INT,nctx-
>nol_quantity_len,
  ! NITEMS,&nctx->nol_q_count);
  ! OCIBNDPLA(nctx->cur1, nctx-
>i_price_bp,errhp,"i_price",i_price,SIZ(int),
  ! SOLT_INT, nctx->i_price_len, NITEMS,
  &nctx->nol_item_count);
  #endif

#ifdef USE_IEEE_NUMBER
  OCIBNDPLA(nctx->cur1, nctx-
>ol_quantity_bp,errhp,"ol_quantity",
  nol_quantity,
  SIZ(float),SOLT_BFLOAT,nctx-
>nol_quantity_len,
  NITEMS,&nctx->nol_q_count);

  OCIBNDPLA(nctx->cur1, nctx-
>i_price_bp,errhp,"i_price",i_price,SIZ(float),
  SOLT_BFLOAT, nctx->i_price_len,
  NITEMS, &nctx->nol_item_count);
#else

```

```

OCIBNDPLA(nctx->curm1, nctx-
>o_l_quantity_bp,errhp,":o_l_quantity",
    nol_quantity, SIZ(int),SOLT_INT,nctx-
>nol_quantity_len,
    NITEMS,&nctx->nol_q_count);

OCIBNDPLA(nctx->curm1, nctx-
>i_price_bp,errhp,":i_price",i_price,SIZ(int),
    SOLT_INT, nctx->i_price_len, NITEMS,
&nctx->nol_item_count);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

OCIBNDPLA(nctx->curm1, nctx-
>i_name_bp,errhp,":i_name",i_name,
    SIZ(i_name[0]),SOLT_STR, nctx-
>i_name_len,NITEMS,
    &nctx->nol_name_count);

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! OCIBNDPLA(nctx->curm1, nctx-
>s_quantity_bp,errhp,":s_quantity",s_quantity,
!     SIZ(int), SOLT_INT,nctx-
>s_quant_len,NITEMS,&nctx->nol_qty_count);
#endif

#ifdef USE_IEEE_NUMBER
OCIBNDPLA(nctx->curm1, nctx-
>s_quantity_bp,errhp,":s_quantity",s_quantity,
    SIZ(float), SOLT_BFLOAT,nctx-
>s_quant_len,NITEMS,&nctx->nol_qty_count);
#else
OCIBNDPLA(nctx->curm1, nctx-
>s_quantity_bp,errhp,":s_quantity",s_quantity,
    SIZ(int), SOLT_INT,nctx-
>s_quant_len,NITEMS,&nctx->nol_qty_count);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

OCIBNDPLA(nctx->curm1, nctx-
>s_bg_bp,errhp,":brand_generic",brand_generic,
    SIZ(char), SOLT_CHR,nctx-
>s_bg_len,NITEMS,&nctx->nol_bg_count);

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! OCIBNDPLA(nctx->curm1, nctx-
>o_l_amount_bp,errhp,":o_l_amount",nol_amount,
!     SIZ(int),SOLT_INT, nctx-
>nol_amount_len,NITEMS,&nctx-
>nol_am_count);
! OCIBNDPLA(nctx->curm1, nctx-
>s_remote_bp,errhp,":s_remote",nctx-
>s_remote,
!     SIZ(int),SOLT_INT, nctx-
>s_remote_len,NITEMS,&nctx-
>s_remote_count);
#endif

#ifdef USE_IEEE_NUMBER
OCIBNDPLA(nctx->curm1, nctx-
>o_l_amount_bp,errhp,":o_l_amount",nol_amount,
    SIZ(float),SOLT_BFLOAT, nctx-
>nol_amount_len,NITEMS,&nctx-
>nol_am_count);

OCIBNDPLA(nctx->curm1, nctx-
>s_remote_bp,errhp,":s_remote",nctx-
>s_remote,
    SIZ(float),SOLT_BFLOAT, nctx-
>s_remote_len,NITEMS,&nctx-
>s_remote_count);
#else
#endif

```

```

OCIBNDPLA(nctx->curm1, nctx-
>o_l_amount_bp,errhp,":o_l_amount",nol_amount,
    SIZ(int),SOLT_INT, nctx-
>nol_amount_len,NITEMS,&nctx-
>nol_am_count);

OCIBNDPLA(nctx->curm1, nctx-
>s_remote_bp,errhp,":s_remote",nctx-
>s_remote,
    SIZ(int),SOLT_INT, nctx-
>s_remote_len,NITEMS,&nctx-
>s_remote_count);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

/* open second cursor */
DISCARD
OCIERROR(errhp,OCIHandleAlloc(tpcenv,
(dvoid**>(&nctx->curm2),
    OCI_HTYPE_STMT, 0, (dvoid**)0));
DISCARD sprintf((char *) stmbuf, SOLTXT2);
DISCARD
OCIERROR(errhp,OCIStmtPrepare(nctx->curm2,
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];
/* for Warning */
/* 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_l_cnt = NITEMS;

    /* Bind array */
    OCIBNDPLA(nctx->curm2,
idx1arr_bp,errhp,":idx1arr",idx1arr,
        SIZ(int), SOLT_INT, idx1arr_len,
NITEMS,&idx1arr_count);

    DBGLOG("NEW:[1]Start",0);
    execstatus = OCIStmtExecute(tpcsvc,nctx-
>curm2,errhp,1,0,
        NULLP(CONST
OCI_Snapshot),NULLP(OCI_Snapshot),OCI_DEF
AULT);
    DBGLOG("NEW:[1]End >%d",execstatus);
    if(execstatus != OCI_SUCCESS) {

OCITransRollback(tpcsvc,errhp,OCI_DEFAULT)
;
        errcode = OCIERROR(errhp,execstatus);
        return -1;
    }
}

return (0);
}

```

```

int tkvcn ()
{
    int i;
    int rcount;

retry:

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

    /* get number of order lines, and check if all
are local */

    o_l_cnt = NITEMS;
    o_all_local = 1;
    for (i = 0; i < NITEMS; i++) {
        if (nol_i_id[i] == 0) {
            o_l_cnt = i;
            break;
        }
        if (nol_supply_w_id[i] != w_id) {

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
!     nctx->s_remote[i] = 1;
#endif

#ifdef USE_IEEE_NUMBER
        nctx->s_remote[i] = 1.0;
#else
        nctx->s_remote[i] = 1;
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

        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_l_cnt_len = sizeof(o_l_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_l_cnt;
    nctx->nol_i_count = o_l_cnt;
    nctx->nol_q_count = o_l_cnt;
    nctx->nol_s_count = o_l_cnt;
    nctx->s_remote_count = o_l_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_l_cnt; i++) {
    nctx->o_l_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->no_l_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);
nctx->i_name_len[i]=0;
nctx->s_bg_len[i] = 0;
}
for (i = o_ol_cnt; i < NITEMS; i++) {

nctx->no_l_i_id_len[i] = 0;
nctx->no_l_supply_w_id_len[i] = 0;
nctx->no_l_quantity_len[i] = 0;
nctx->no_l_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;
}

DBGLOG("NEW:[2]Start",0);
execstatus = OCISmtExecute(tpcsvc,nctx-
>curr1,errhp,1,0,0,0,
OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
DBGLOG("NEW:[2]End >%d",execstatus);

if(execstatus != OCI_SUCCESS) {

OCITransRollback(tpcsvc,errhp,OCI_DEFAULT)
;
errcode = OCIERROR(errhp,execstatus);
if(errcode == NOT_SERIALIZABLE) {
retries++;
goto retry;
} else if (errcode == RECOVER) {
retries++;
goto retry;
}
/* Deleted T.Kato 02.10.25 */
#if 0
! ) else if (errcode ==
SNAPSHOT_TOO_OLD) {
! retries++;
! goto retry;
#endif
/* Deleted end */
} else {
return -1;
}
}

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

#ifdef DEBUG
printf("w_id = %d, d_id = %d, c_id
= %d\n",w_id, d_id, c_id);
#endif

return (0);
}

```

```

void tkvcndone ()
{
/* for warning */
/* int i;*/

if (nctx)
{
DISCARD OCIHandleFree((dvoid *)nctx-
>curr1,OCI_HTYPE_STMT);
DISCARD OCIHandleFree((dvoid *)nctx-
>curr2,OCI_HTYPE_STMT);
free (nctx);
}
}

:-----:
svrapl/3tier/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_tpcitrc
lwang 07/23/01 - fix include
lwang 07/23/01 - Creation

*/

#include "forlinux.h"
#include "log.h"
#include "tpcc.h"
#include "GlobalArea.h"
#include "prototype.h"

:-----:
PRIVATE TYPES AND
CONSTANTS
:-----:
*/

```

```

/*-----:
:-----:
STATIC FUNCTION
DECLARATIONS
:-----:
*/

/* Replaced T.Kato 2004.12.21 New Oracle10g
tool kit */
#if 0
#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 "
#endif

#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 \
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_of_cnt \
        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 /*+ INDEX(ordl) */
\
        ol_i_id, ol_supply_w_id, ol_quantity,
ol_amount, ol_delivery_d \
        FROM ordl \
        WHERE ol_o_id = :o_id AND ol_d_id
= :d_id AND ol_w_id = :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"

/* Replaced end */

int tkvcoin ()
{
    int i;
    text stmbuff[SQL_BUF_SIZE];

    octx = (ordctx *) malloc (sizeof(ordctx));
    DISCARD memset(octx, (char)0, sizeof(ordctx));
    octx->cs = 1;
    octx->norow = 0;
    octx->somerows = 10;

    /* Deleted T.Kato 2004.12.21 New Oracle10g
    tool kit */
    #if 0
    ! /* get the rowid handles */
    ! OCIERROR(errhp, OCIDescriptorAlloc((dvoid
    *)tpcenv, (dvoid **)&octx->o_rowid,
    ! (ub4)OCI_DTYPE_ROWID,
    (size_t) 0, (dvoid **)0));
    #endif
    /* Deleted end */

    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));

    /* Deleted T.Kato 2004.12.21 New Oracle10g
    tool kit */
    #if 0
    ! DISCARD OCIERROR(errhp,
    ! OCIHandleAlloc(tpcenv, (dvoid **)&octx-
    >curo0, OCI_HTYPE_STMT, 0, (dvoid **)0));

```

```

#endif
/* Deleted end */

    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 *) stmbuff, SQLCUR0);
    DISCARD OCIERROR(errhp,
    OCIStmtPrepare(octx-
    >curo0, errhp, stmbuff, (ub4)strlen((char *)stmbuff),

    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 *) stmbuff, SQLCUR1);
    DISCARD OCIERROR(errhp,
    OCIStmtPrepare(octx-
    >curo1, errhp, stmbuff, (ub4)strlen((char *)stmbuff),

    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 *) stmbuff, SQLCUR2);
    DISCARD OCIERROR(errhp,
    OCIStmtPrepare(octx-
    >curo2, errhp, stmbuff, (ub4)strlen((char *)stmbuff),

    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 *) stmbuff, SQLCUR3);
    DISCARD OCIERROR(errhp,
    OCIStmtPrepare(octx-
    >curo3, errhp, stmbuff, (ub4)strlen((char *)stmbuff),

    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 *) stmbuff, SQLCUR4);
    DISCARD OCIERROR(errhp,
    OCIStmtPrepare(octx-
    >curo4, errhp, stmbuff, (ub4)strlen((char *)stmbuff),

    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), SOLT_INT);
    OCIBND(octx->curo0, octx-
    >d_id_bp[0], errhp, "d_id", ADR(d_id),
    SIZ(int), SOLT_INT);
    OCIBND(octx->curo0, octx-
    >c_last_bp[0], errhp, "c_last", c_last,
    SIZ(c_last), SOLT_STR);
    OCIDFNRA(octx->curo0, octx-
    >c_rowid_dp, errhp, 1, octx->c_rowid_ptr,
    SIZ(OCIRowid*), SOLT_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]), SOLT_RDD);
    OCIDEF(octx->curo1, octx-
    >c_id_dp, errhp, 1, ADR(c_id), SIZ(int), SOLT_INT);

    /* Replaced T.kato 03.09.09 Oracle10g tool kit */
    #if 0
    ! OCIDEF(octx->curo1, octx-
    >c_balance_dp[0], errhp, 2, ADR(c_balance),
    ! SIZ(double), SOLT_FLT);
    #endif

    #ifdef USE_IEEE_NUMBER
    OCIDEF(octx->curo1, octx-
    >c_balance_dp[0], errhp, 2, ADR(c_balance),
    SIZ(double), SOLT_BDOUBLE);
    #else
    OCIDEF(octx->curo1, octx-
    >c_balance_dp[0], errhp, 2, ADR(c_balance),
    SIZ(double), SOLT_FLT);
    #endif /* USE_IEEE_NUMBER */
    /* Replaced end */

    OCIDEF(octx->curo1, octx-
    >c_first_dp[0], errhp, 3, c_first, SIZ(c_first)-1,
    SOLT_CHR);
    OCIDEF(octx->curo1, octx-
    >c_middle_dp[0], errhp, 4, c_middle,
    SIZ(c_middle)-1, SOLT_AFC);

```

```

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_IN
T);
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);

/* Deleted T.Kato 2004.12.21 New Oracle10g
tool kit */
#if 0
! OCIDEF(octx->curo1,octx-
>o_rowid_dp[0],errhp,10,ADR(octx->o_rowid),
! SIZ(OCIRowid*),SQLT_RDD);
#endif
/* deleted end */

/* 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);

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! OCIDEF(octx->curo2,octx-
>c_balance_dp[1],errhp,1,ADR(c_balance),
! SIZ(double),SQLT_FLT);
#endif

#ifdef USE_IEEE_NUMBER
OCIDEF(octx->curo2,octx-
>c_balance_dp[1],errhp,1,ADR(c_balance),
SIZ(double),SQLT_BDOUBLE);
#else
OCIDEF(octx->curo2,octx-
>c_balance_dp[1],errhp,1,ADR(c_balance),
SIZ(double),SQLT_FLT);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

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_IN
T);
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);

/* Deleted T.Kato 2004.12.21 New Oracle10g
tool kit */
#if 0
! OCIDEF(octx->curo2,octx-
>o_rowid_dp[1],errhp,9,ADR(octx->o_rowid),
! SIZ(OCIRowid*),SQLT_RDD);
#endif
/* Deleted end */

/* Bind for last cursor */

/* Replaced T.Kato 2004.12.21 New Oracle10g
tool kit */
#if 0
! 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);
! */
#endif

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);
/* Replaced end */

/* Deleted T.Kato 2004.12.21 New Oracle10g
tool kit */
#if 0
! OCIBND(octx->curo3,octx-
>o_rowid_bp,errhp,:"ordr_rowid",
! &octx->o_rowid,
SIZ(OCIRowid*),SQLT_RDD);
#endif
/* Deleted end */

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);

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! OCIDFNRA(octx->curo3, octx-
>ol_quantity_dp,errhp,3, ol_quantity,SIZ(int),
! SQLT_INT, NULL,octx->ol_quantity_len,
NULL);
! OCIDFNRA(octx->curo3,octx-
>ol_amount_dp,errhp,4,ol_amount, SIZ(int),
! SQLT_INT,NULL, octx->ol_amount_len,
NULL);
#endif

#ifdef USE_IEEE_NUMBER

```

```

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);
#else
OCIDFNRA(octx->curo3, octx-
>ol_quantity_dp,errhp,3, ol_quantity,SIZ(int),
SQLT_INT, NULL,octx->ol_quantity_len,
NULL);
OCIDFNRA(octx->curo3,octx-
>ol_amount_dp,errhp,4,ol_amount, SIZ(int),
SQLT_INT,NULL, octx->ol_amount_len,
NULL);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

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);
}

int tkvco ()
{
int i;
int rcount;

#ifdef ISO9
int secondread = 0;
char sdate[30];
ub4 datelen;
sysdate(sdate);
printf("Order Status started at: %s\n", sdate);
#endif

int oci_stat;

/* Deleted T.Kato 2004.12.21 "o_rowid" was
deleted by New Oracle10g tool kit */
#if 0
!!int f_w_id = w_id;
!!int f_d_id = d_id;
!!int f_c_id = c_id;
!!
!!int c2_w_id = -1;
!!int c2_d_id = -1;
!!int c2_c_id = -1;
!!unsigned char b_row_id[512];
!!unsigned char a_row_id[512];
!!
!!ub2 buf_len = sizeof(b_row_id) - 1;
!!

```

```

!!memset(b_row_id, 0x00, sizeof(b_row_id));
!!memset(a_row_id, 0x00, sizeof(a_row_id));
#endif
/* Deleted end */

/* Added T.Kato 2004.12.21 New Oracle10g tool
kit */
#ifdef DEBUG
if (bylastname) tkvc_trace_on();
#endif

#ifdef BLANK_PAD_C_LAST
for (i = strlen(c_last); i < sizeof(c_last)-1; i++)
{
c_last[i] = ' ';
}
c_last[i] = '\0';
#endif
/* Added end */

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)
{
/* Replaced T.Kato 2004.12.21 New Oracle tool
kit */
/* cbctx.reexec = FALSE;*/

ordcount++;
cbctx.reexec = FALSE;
errcode = 0;
/*#define STRIP_BLANKS_C_LAST Always
no blanks */
#ifdef STRIP_BLANKS_C_LAST
for (i = strlen(c_last)-1; i >= 0 && (c_last[i] == '
'); i--)
{
c_last[i] = '\0';
}
#endif
/* Replaced end */

DBGLOG("ORD:[1]Start",0);
execstatus=OCIStmtExecute(tpcsvc,octx-
>curo0,errhp,100,0,
NULLP(CONST
OCI_Snapshot),NULLP(OCI_Snapshot),OCI_DEF
AULT);
DBGLOG("ORD:[1]End >%d",execstatus);
/* 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 == RECOVER))
{
DISCARD
OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
retries++;
goto retry;
} else {

```

```

/* Added T.Kato 2004.12.21 New Oracle10g tool
kit */
#ifdef DEBUG
tkvc_trace_off();
#endif
/* Added end */
return -1;
}
}
if (execstatus == OCI_NO_DATA) /* there are
no more rows */
{
/* get rowcount, find middle one */
/* Replaced T.Kato 03.10.14 Add error check */
/* DISCARD OCIAttrGet(octx-
>curo0,OCI_HTYPE_STMT,&rcount,NULL, */
/* OCI_ATTR_ROW_COUNT,errhp); */

oci_stat = OCIAttrGet(octx-
>curo0,OCI_HTYPE_STMT,&rcount,NULL,
OCI_ATTR_ROW_COUNT,errhp);
DISCARD OCIERROR(errhp, oci_stat);

/* Deleted T.Kato 04.06.22 for Linux */
#if 0
! if (oci_stat == OCI_SUCCESS)
! {
! TpcUserLog(LOG_FILE_INF,
"ORDERSTATUS OCI_ATTR_ROW_COUNT
success\n");
! }
#endif
/* Deleted end */

/* Replaced end */

if (rcount < 1)
{
/* Replaced T.Kato 2004.12.21 New Oracle10g
tool kit */
#if 0
! TpcUserLog(LOG_FILE_INF,
"ORDERSTATUS rcount=%d\n",rcount);
! return (-1);
#endif
TpcUserLog(LOG_FILE_INF,
"ORDERSTATUS rcount=%d\n",rcount);
TpcUserLog(LOG_FILE_INF, "
w_id =%d\n",w_id);
TpcUserLog(LOG_FILE_INF, "
d_id =%d\n",d_id);
TpcUserLog(LOG_FILE_INF, "
c_last=%s\n",c_last);
TpcUserLog(LOG_FILE_INF, "
retries=%d\n",retries);
TpcUserLog(LOG_FILE_INF, "
errcode=%d\n",errcode);
TpcUserLog(LOG_FILE_INF, "
execstatus=%d\n",execstatus);
TpcUserLog(LOG_FILE_INF, "
ordcount=%d\n",ordcount);
#ifdef DEBUG
tkvc_trace_off();
#endif
return -1;
/* Replaced end */

}
octx->cust_idx=(rcount)/2 ;

```

```

}
else
{
/* count the number of rows */
DBGLOG("ORD:[2]Start",0);
execstatus=OCIStmtExecute(tpcsvc,octx-
>curo4,errhp,1,0,
NULLP(CONST
OCI_Snapshot),NULLP(OCI_Snapshot),OCI_DEF
AULT);
DBGLOG("ORD:[2]End >%d",execstatus);
if ((execstatus != OCI_NO_DATA) &&
(execstatus != OCI_SUCCESS))
{
errcode=OCIERROR(errhp, execstatus);
if ((errcode == NOT_SERIALIZABLE) ||
(errcode == RECOVER))
{
DISCARD
OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
retries++;
goto retry;
} else {
return -1;
}
}
}

/* Replaced T.Kato 2004.12.21 New Oracle10g
tool kit */
#if 0
! if (octx->rcount+1 < 2*10 )
! octx->cust_idx=(octx->rcount+1)/2 ;
! else /* */
! {
! cbctx.reexec = TRUE;
! cbctx.count = (octx->rcount+1)/2 ;
! DBGLOG("ORD:[3]Start",0);
! execstatus=OCIStmtExecute(tpcsvc,octx-
>curo0,errhp,cbctx.count,
! 0,NULLP(CONST
OCI_Snapshot),
! NULLP(OCI_Snapshot),OCI_DEFAULT);
! DBGLOG("ORD:[3]End
>%d",execstatus);
! /* will get OCI_NO_DATA if <100 found */
! if (cbctx.count > 0)
! {
! TpcUserLog (LOG_FILE_INF, "did not
get all rows ");
! return (-1);
! }
! if ((execstatus != OCI_NO_DATA) &&
(execstatus != OCI_SUCCESS))
! {
! errcode=OCIERROR(errhp, execstatus);
! if ((errcode == NOT_SERIALIZABLE) ||
(errcode == RECOVER))
! {
! DISCARD
OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
! retries++;
! goto retry;
! } else {
! return -1;
! }
! }
! octx->cust_idx=0 ;
! }
#endif

cbctx.reexec = TRUE;
cbctx.count = (octx->rcount+1)/2 ;

```

```

    execstatus=OCISmtExecute(tpcsvc,octx-
>curo0,errhp,cbctx.count,
        0,NULLP(CONST
OCI_Snapshot),
NULLP(OCI_Snapshot),OCI_DEFAULT);

    DISCARD OCIAAttrGet(octx-
>curo0,OCI_HTYPE_STMT,&rcount,NULL,
OCI_ATTR_ROW_COUNT,errhp);

    /* will get OCI_NO_DATA if <100 found */
    if (cbctx.count != (unsigned int)rcount)
    {
        TpcUserLog (LOG_ERR, "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 {
            return -1;
        }
    }

    octx->cust_idx=cbctx.count - 1 ;
/* Replaced end */

}

    octx->c_rowid_cust = octx->c_rowid_ptr[octx-
>cust_idx];
    DBGLOG("ORD:[4]Start",0);
    execstatus=OCISmtExecute(tpcsvc,octx-
>curo1,errhp,1,0,
        NULLP(CONST
OCI_Snapshot),NULLP(OCI_Snapshot),OCI_DEF
AULT);
    DBGLOG("ORD:[4]End >%d",execstatus);
    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 {
            return -1;
        }
    }
} else
{
    DBGLOG("ORD:[5]Start",0);
    execstatus=OCISmtExecute(tpcsvc,octx-
>curo2,errhp,1,0,
        NULLP(CONST
OCI_Snapshot),NULLP(OCI_Snapshot),
OCI_DEFAULT);
    DBGLOG("ORD:[5]End >%d",execstatus);
    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
        {
            return -1;
        }
    }

/* Deleted T.Kato 2004.12.21 "o_rowid" was
deleted by New Oracle10g tool kit */
#ifndef ISO9
    /* Deleted T.Kato 2004.12.21 "o_rowid" was
deleted by New Oracle10g tool kit */
    #if 0
    !c2_w_id = w_id;
    !c2_d_id = d_id;
    !c2_c_id = c_id;
    !!
    !!OCIRowidToChar(octx->o_rowid, b_row_id,
&buf_len, errhp);
    #endif
    /* Deleted end */

    #ifdef ISO9
        sysdate (sdate);
    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_ent
ry_d_base,(text*)FULLDATE,SIZ(FULLDATE),(t
ext*
)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->o_l_w_id_len = sizeof(int);
    octx->o_l_d_id_len = sizeof(int);
    octx->o_l_o_id_len = sizeof(int);

    DBGLOG("ORD:[6]Start",0);

```

```

    execstatus = OCISmtExecute(tpcsvc,octx-
>curo3,errhp,o_ol_cnt,0,
        NULLP(CONST
OCI_Snapshot),NULLP(OCI_Snapshot),
OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
    DBGLOG("ORD:[6]End >%d",execstatus);
    if (execstatus != OCI_SUCCESS )
    {
        errcode=OCIERROR(errhp,execstatus);

/* Deleted T.Kato 2004.12.21 "o_rowid" was
deleted by New Oracle10g tool kit */
    #if 0
    !!OCIRowidToChar(octx->o_rowid, a_row_id,
&buf_len, errhp);
    !!TpcUserLog(LOG_FILE_INF, "DBG_LOG
start : w_id=%d d_id=%d c_id=%d\\n", f_w_id,
f_d_id, f_c_id);
    !!TpcUserLog(LOG_FILE_INF, "DBG_LOG
cur2 : w_id=%d d_id=%d c_id=%d\\n", c2_w_id,
c2_d_id, c2_c_id);
    !!TpcUserLog(LOG_FILE_INF, "DBG_LOG
cur2 : row_id=%s\\n", b_row_id);
    !!TpcUserLog(LOG_FILE_INF, "DBG_LOG
error : row_id=%s\\n", a_row_id);
    #endif
    /* Deleted end */

        DISCARD
OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
        if((errcode == NOT_SERIALIZABLE) ||
(errcode == RECOVERR)
|| (errcode == SNAPSHOT_TOO_OLD))
        {
            retries++;
            goto retry;
        }
        else
        {

/* Added T.Kato 2004.12.21 New Oracle10g tool
kit */
        #ifdef DEBUG
            if (bylastname) tkvc_trace_off();
        #endif
        /* Added end */
        return -1;
    }
}

/* clean up and convert the delivery dates */
for (i = 0; i < o_ol_cnt; i++)
{
    o_l_del_len[i]=sizeof(o_l_delivery_d[i]);
    DISCARD
OCIERROR(errhp,OCIDateToText(errhp,&o_l_d_
base[i],
        (const
text*)SHORTDATE,(ub1)strlen(SHORTDATE),(t
ext*)0,0,
        &o_l_del_len[i], o_l_delivery_d[i]);
}
/*
    cvtdmy(o_l_d_base[i],o_l_delivery_d[i]);
*/
}

/* Added T.Kato 2004.12.21 New Oracle10g tool
kit */
#ifdef DEBUG
    if (bylastname) tkvc_trace_off();
#endif
/* Added end */

```

```

return (0);
}

/* Added T.Kato 2004.12.21 New Oracle10g tool
kit */
#ifdef DEBUG
#define SQLTRCON "alter session set events
'10046 trace name context forever, level 12"
#define SQLTRCOFF "alter session set events
'10046 trace name context off"

/*static trace_on = 0; Moved to Global Area */

tkvc_trace_on()
{
    if (!trace_on)
    {
        char stmbuf[100];
        OCIStmt *curtrc;
        OCIHandleAlloc(tpcenv, (dvoid **)&curtrc,
        OCI_HTYPE_STMT, 0, (dvoid**)0);
        strcpy((char *) stmbuf, SQLTRCON);
        DISCARD OCIERROR(errhp,
        OCIStmtPrepare(curtrc, errhp, stmbuf,
        strlen((char *)stmbuf),
        OCI_NTV_SYNTAX, OCI_DEFAULT));
        OCIERROR(errhp,
        OCIStmtExecute(tpcenv, curtrc,
        errhp, 1, 0, 0, 0, OCI_DEFAULT));
        OCIHandleFree((dvoid *)curtrc,
        OCI_HTYPE_STMT);
        trace_on++;
    }
}

tkvc_trace_off()
{
    if (trace_on)
    {
        char stmbuf[100];
        OCIStmt *curtrc;
        OCIHandleAlloc(tpcenv, (dvoid **)&curtrc,
        OCI_HTYPE_STMT, 0, (dvoid**)0);
        strcpy(stmbuf, SQLTRCOFF);
        DISCARD OCIERROR(errhp,
        OCIStmtPrepare(curtrc, errhp, stmbuf,
        strlen((char *)stmbuf),
        OCI_NTV_SYNTAX, OCI_DEFAULT));
        OCIERROR(errhp,
        OCIStmtExecute(tpcenv, curtrc,
        errhp, 1, 0, 0, 0, OCI_DEFAULT));
        OCIHandleFree((dvoid *)curtrc,
        OCI_HTYPE_STMT);
        trace_on = 0;
    }
}
#endif
/* Added end */

void tkvcodone ()
{
    if (octx)
        free (octx);
}

/* end of file tkvcord.c */

```

```

svrapl/3tier/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 "forlinux.h"
#include "log.h"
#include "tpcc.h"
#include "GlobalArea.h"
#include "prototype.h"

#define SQLTXT_INIT "BEGIN inittpc.init_pay;
END;"

int tkvcpin (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),
    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));
    DBGLOG("PAY:[1]Start", 0);

```

```

DISCARD OCIERROR(errhp,
OCIStmtExecute(tpcenv, pctx->curpi, errhp, 1, 0,
NULLP(CONST
OCI_Snapshot), NULLP(OCI_Snapshot), OCI_DEF
AULT));
DBGLOG("PAY:[1]End ", 0);

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

/* Replaced 04.01.20 TUXEDO Client */
#if 0
! sqlfile("../blocks/paynz.sql", stmbuf);
#endif
sqlfile("/home/tpc/blocks/paynz.sql", stmbuf);
/* Replaced end */
DISCARD
OCIERROR(errhp, OCIStmtPrepare(pctx->curp0,
errhp, stmbuf,
strlen((char *)stmbuf),
OCI_NTV_SYNTAX, OCI_DEFAULT));

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

/* Replaced 04.01.20 TUXEDO Client */
#if 0
! sqlfile("../blocks/payz.sql", stmbuf); /* sqlfile
opens $O/bench/.../blocks/... */
#endif
sqlfile("/home/tpc/blocks/payz.sql", stmbuf); /*
sqlfile opens $O/bench/.../blocks/... */
/* Replaced end */
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);

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! OCIBNDPL(pctx->curp0, pctx-
>h_amount_bp[0],
errhp,"h_amount",ADR(h_amount),
! SIZ(int),SQLT_INT, &pctx-
>h_amount_len);
#endif

#ifdef USE_IEEE_NUMBER
OCIBNDPL(pctx->curp0, pctx-
>h_amount_bp[0],
errhp,"h_amount",ADR(h_amount),
SIZ(float),SQLT_BFLOAT, &pctx-
>h_amount_len);
#else
OCIBNDPL(pctx->curp0, pctx-
>h_amount_bp[0],
errhp,"h_amount",ADR(h_amount),
SIZ(int),SQLT_INT, &pctx-
>h_amount_len);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

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,
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);

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! OCIBNDPL(pctx->curp0, pctx-
>c_balance_bp[0], errhp,"c_balance",
! ADR(c_balance), SIZ(double),SQLT_FLT,
&pctx->c_balance_len);
#endif

#ifdef USE_IEEE_NUMBER
OCIBNDPL(pctx->curp0, pctx-
>c_balance_bp[0], errhp,"c_balance",
ADR(c_balance),
SIZ(double),SQLT_BDOUBLE, &pctx-
>c_balance_len);
#else
OCIBNDPL(pctx->curp0, pctx-
>c_balance_bp[0], errhp,"c_balance",

```

```

ADR(c_balance), SIZ(double),SQLT_FLT,
&pctx->c_balance_len);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

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);

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! OCIBNDPL(pctx->curp1, pctx-
>h_amount_bp[1],
errhp,"h_amount",ADR(h_amount),
! SIZ(int),SQLT_INT, &pctx-
>h_amount_len);
#endif

#ifdef USE_IEEE_NUMBER
OCIBNDPL(pctx->curp1, pctx-
>h_amount_bp[1],
errhp,"h_amount",ADR(h_amount),
SIZ(float),SQLT_BFLOAT, &pctx-
>h_amount_len);
#else
OCIBNDPL(pctx->curp1, pctx-
>h_amount_bp[1],
errhp,"h_amount",ADR(h_amount),
SIZ(int),SQLT_INT, &pctx-
>h_amount_len);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

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);

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! OCIBNDPL(pctx->curp1, pctx-
>c_balance_bp[1], errhp,"c_balance",
! ADR(c_balance), SIZ(double),SQLT_FLT,
&pctx->c_balance_len);
#endif

#ifdef USE_IEEE_NUMBER
OCIBNDPL(pctx->curp1, pctx-
>c_balance_bp[1], errhp,"c_balance",
    ADR(c_balance),
SIZ(double),SQLT_BDOUBLE, &pctx-
>c_balance_len);
#else
OCIBNDPL(pctx->curp1, pctx-
>c_balance_bp[1], errhp,"c_balance",
    ADR(c_balance), SIZ(double),SQLT_FLT,
&pctx->c_balance_len);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

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);
}

int 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) {
DBGLOG("PAY:[2]Start",0);
execstatus=OCIStmtExecute(tpcsvc,pctx-
>curp1,errhp,1,0,
    NULL(CONST
OCISnapshot),NULL(OCISnapshot),

OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS);
DBGLOG("PAY:[2]End >%d",execstatus);
} else {
DBGLOG("PAY:[3]Start",0);
execstatus=OCIStmtExecute(tpcsvc,pctx-
>curp0,errhp,1,0,
    NULL(CONST
OCISnapshot),NULL(OCISnapshot),

OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS);
DBGLOG("PAY:[3]End >%d",execstatus);
}

if(execstatus != OCI_SUCCESS) {
OCITransRollback(tpcsvc,errhp,OCI_DEFAULT)
;
errcode = OCIERROR(errhp,execstatus);
if(errcode == NOT_SERIALIZABLE) {
retries++;
goto retry;
} else if (errcode == RECOVERERR) {
retries++;
goto retry;
} else if (errcode == SNAPSHOT_TOO_OLD)
{
retries++;
goto retry;
} else {
return -1;
}
}
return 0;

void tkvcpdone ()
{
if(pctx) {
free(pctx);
}
}

.....
svrapl/3tier/plsto.c

```



```

.....
#endif 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.
+=====
=====*/

#include "forlinux.h"
#include "log.h"
#include "tpcc.h"
#include "GlobalArea.h"
#include "prototype.h"

#define PLSQLSTO
#define SQLTXT "BEGIN
stocklevel.getstocklevel (:w_id, :d_id, :threshold,
\
:low_stock); END;"
#else
/* Replaced Hayashi 06.02.20 New Oracle10g
tool kit */
#if 0
/* Replaced Hayashi 06.01.12 New Oracle10g
tool kit */
/* Replaced T.Kato 03.07.18 New Oracle10i tool
kit */
/*#define SQLTXT "SELECT count (DISTINCT
s_i_id) \ */
/*#define SQLTXT "SELECT /*+ nocache (stok)
*/ count (DISTINCT s_i_id) \ */
/*#define SQLTXT "SELECT /*+ USE_NL(ordl)
nocache (stok) */ count (DISTINCT s_i_id) \ */
#endif
#define SQLTXT "SELECT /*+ USE_NL(ordl) */
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

int tkvcsinit ()
{
text stmbuf[SQL_BUF_SIZE];

```

```

sctx = (stocx *)malloc(sizeof(stocx));
memset(sctx, (char)0, sizeof(stocx));

sctx->norow=0;

OCIERROR(errhp,
OCIHandleAlloc(tpcenv, (dvoid**)&sctx-
>curs, OCI_HTYPE_STMT, 0, (dvoid**)0));
sprintf ((char *) stmbuf, SOLTXT);
OCIERROR(errhp, OCIStmtPrepare(sctx-
>curs, errhp, stmbuf, strlen((char *)stmbuf),
OCI_NTV_SYNTAX, OCI_DEFAULT));
#endif 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);

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! OCIBND(sctx->curs, sctx->threshold_bp, errhp,
":threshold", ADR(threshold),
! sizeof(int), SQLT_INT);
#endif

#ifdef USE_IEEE_NUMBER
OCIBND(sctx->curs, sctx->threshold_bp, errhp,
":threshold", ADR(threshold),
sizeof(float), SQLT_BFLOAT);
#else
OCIBND(sctx->curs, sctx->threshold_bp, errhp,
":threshold", ADR(threshold),
sizeof(int), SQLT_INT);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

#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);
}

int tkvcs ()
{
retry:
DBGLOG("STO:[1]Start", 0);
execstatus= OCIStmtExecute(tpcsvc, sctx-
>curs, errhp, 1, 0, 0, 0,

```

```

OCI_COMMIT_ON_SUCCESS |
OCI_DEFAULT);
DBGLOG("STO:[1]End >%d", execstatus);
if (execstatus != OCI_SUCCESS)
{
errcode=OCIERROR(errhp, execstatus);

OCITransCommit(tpcsvc, errhp, OCI_DEFAULT);
if((errcode == NOT_SERIALIZABLE) ||
(errcode == RECOVER))
|| (errcode == SNAPSHOT_TOO_OLD))
{
retries++;
goto retry;
} else {
return -1;
}
}

return (0);
}

void tkvcsdone ()
{
if(sctx) free(sctx);
}

.....
svrapl/3tier/tpccpl.c
.....

#endif 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 (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 "forlinux.h"
#include <stdio.h>
#include <sys/types.h>
#include <sys/poll.h>
#include <sys/time.h>
#include <unistd.h>
//#include <time.h>
#include "tpcc.h"
/* Added T.Kato 02.10.23 Ajustment interface for
transaction data organization format*/
#include "tpcc_info.h"
/* Added end */
#include "log.h"
#include "log_level.h"

```

```

#include "GlobalArea.h"
#include "prototype.h"

#define SQLTXT "alter session set
isolation_level = serializable"
#define SQLTXTTRC "alter session set
sql_trace = true"
#define SQLTXTTIM "alter session set
timed_statistics = true"

#ifdef ORA_NT
#undef boolean
#include "dpbcore.h"
#define gettime dpbtimef
#else
extern double gettime ();
#endif

/*
extern char oracle_home[256];
*/

/* NewOrder Binding stuff */

/* vmm313 void ocierror(fname, lineno, errhp,
status) */
int ocierror(char *fname, int lineno, OCIError
*errhp, sword status)
{
text errbuf[512];
sb4 errcode;
sb4 lstat;
ub4 recno=2;

switch (status) {
case OCI_SUCCESS:
break;

case OCI_SUCCESS_WITH_INFO:
TpccUserLog(LOG_FILE_INF, "Module %s
Line %d\n", fname, lineno);
TpccUserLog(LOG_FILE_INF, "Error -
OCI_SUCCESS_WITH_INFO\n");

lstat = OCIErrorGet (errhp, recno++, (text *)
NULL, &errcode, errbuf,
(ub4) sizeof(errbuf),
OCI_HTYPE_ERROR);

TpccUserLog(LOG_FILE_INF, "Error - %s\n",
errbuf);
break;

case OCI_NEED_DATA:
TpccUserLog(LOG_FILE_INF, "Module %s
Line %d\n", fname, lineno);
TpccUserLog(LOG_FILE_INF, "Error -
OCI_NEED_DATA\n");
return (IRRECERR);

case OCI_NO_DATA:
TpccUserLog(LOG_FILE_INF, "Module %s
Line %d\n", fname, lineno);
TpccUserLog(LOG_FILE_INF, "Error -
OCI_NO_DATA\n");
return (IRRECERR);

case OCI_ERROR:
/* Replaced T.Kato 03.09.12 */
#if 0
! 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);
#endif

lstat = OCIErrorGet (errhp, (ub4) 1,
(text *) NULL, &errcode, errbuf,
(ub4) sizeof(errbuf),
OCI_HTYPE_ERROR);
if (errcode == NOT_SERIALIZABLE) {
TpccUserLog(LOG_FILE_INF, "Module %s
Line %d\n", fname, lineno);
TpccUserLog(LOG_FILE_INF, "Information
- NOT_SERIALIZABLE (OCI_ERROR)\n");
return (errcode);
}
if (errcode == SNAPSHOT_TOO_OLD) {
TpccUserLog(LOG_FILE_INF, "Module %s
Line %d\n", fname, lineno);
TpccUserLog(LOG_FILE_INF, "Information
- SNAPSHOT_TOO_OLD (OCI_ERROR)\n");
return (errcode);
}

/* Replaced end */
while (lstat != OCI_NO_DATA)
{
TpccUserLog(LOG_FILE_INF, "Module %s
Line %d\n", fname, lineno);
TpccUserLog(LOG_FILE_INF, "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:
TpccUserLog(LOG_FILE_INF, "Module %s
Line %d\n", fname, lineno);
TpccUserLog(LOG_FILE_INF, "Error -
OCI_INVALID_HANDLE\n");
/* Replaced 03.05.15 TPCexit no argument */
// TPCexit(1);
TPCexit();
/* Replaced end */
exit(-1);

case OCI_STILL_EXECUTING:
TpccUserLog(LOG_FILE_INF, "Module %s
Line %d\n", fname, lineno);
TpccUserLog(LOG_FILE_INF, "Error -
OCI_STILL_EXECUTING\n");
return (IRRECERR);

case OCI_CONTINUE:
TpccUserLog(LOG_FILE_INF, "Module %s
Line %d\n", fname, lineno);
TpccUserLog(LOG_FILE_INF, "Error -
OCI_CONTINUE\n");
return (IRRECERR);

default:
TpccUserLog(LOG_FILE_INF, "Module %s
Line %d\n", fname, lineno);
TpccUserLog(LOG_FILE_INF, "Status - %d\n",
status);
return (IRRECERR);

```

```

}
return (RECOVERR);
}

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

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

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

int sqlfile(char *fnam, text *linebuf)
{
FILE *fd;
int nulpt = 0;
char realfile[512];

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

/*
sprintf(realfile, "%s/bench/tpc/tpcc/blocks/%s", ora
cle_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 *orad)
{
struct tm *loctime;
time_t int_time;

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

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

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

/* Convert the current date and time into local
time */

```

```

loctime = localtime( &int_time);

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

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

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

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

    int day,month,year;
    int hour,min,sec;

    memcpy(&Date,oracle,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)
{
    TpcUserLog(LOG_INF, "Server Apl end
procedure execute (TPCexit)\n");

    if (new_init) {
        tkvcndone();
        new_init = 0;
    }

    if (pay_init) {
        tkvcpdone();
        pay_init = 0;
    }

    if (ord_init) {
        tkvcodone();
        ord_init = 0;
    }

#ifdef DEL_ORA8I
    if (del_init) {
        tkvcddone();
        del_init = 0;
    }
#else
    if (del_init_oci) {
        tkvcddone(0);
        del_init_oci = 0;
    }

    if (del_init_plsql) {
        tkvcddone(1);
        del_init_plsql = 0;
    }
#endif

    if (sto_init) {
        tkvcsdone();
        sto_init = 0;
    }
}

/* Deleted T.Kato 040120 Shutdown can
disconnect server normally without the following
logic for TUXEDO. */
/* But You must be valid the
following logic for COM+. */
#if 0
!
OCIERROR(errhp,OCISessionEnd(tpcsvc,errhp,
tpcusr, OCI_DEFAULT));
! OCIERROR(errhp,OCIServerDetach(tpcsrv,
errhp, OCI_DEFAULT));
#endif
/* Deleted end */

    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);

/* Close Derivery log */
if (lfp) {
    fclose (lfp);
    lfp = NULL;
}
TpcUserLog(LOG_INF, "TPCexit all
finished\n");
}

int TPCinit (int id, char *uid, char *pwd)
{
/* Deleted T.Kato 02.10.24 Deleted derivery log
open
! char filename[40];
Deleted end */

    text stmbuf[100];

/* Added T.Kato 02.10.24 */
    sword rval;
/* Added End */

/* Replaced T.kato 02.10.24 Moved delivery log
open */
#if 0
! proc_no = id;
! sprintf (filename, "tpcc_%d.del", proc_no);
! if ((lfp = fopen (filename, "w")) == NULL) {
#ifdef TUX
! TpcUserLog ("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
! return (-1);
! }
#endif
// Init delevery flag
iflg = 0;
/* replaced end */

```

```

/* Replaced T.Kato 04.03.14 For Tuxedo
process */
#if 0
/* Replaced 03.05.19 For Thread */
!#if 0
!!
OCIInitialize(OCI_DEFAULT|OCI_OBJECT,(dvoi
d *)0,0,0,0);
!#endif
!
OCIInitialize(OCI_THREADED|OCI_OBJECT,(d
void *)0,0,0,0);
/* Replaced end */
#endif

OCIInitialize(OCI_DEFAULT|OCI_OBJECT,(dvoi
d *)0,0,0,0);
/* Replaced end */

OCIEnvInit(&tpcenv, OCI_DEFAULT, 0, (dvoid
**));
OCIHandleAlloc((dvoid *)tpcenv, (dvoid
**) &tpcsrv, OCI_HTYPE_SERVER, 0, (dvoid
**));
OCIHandleAlloc((dvoid *)tpcenv, (dvoid
**) &errhp, OCI_HTYPE_ERROR, 0, (dvoid
**));
OCIHandleAlloc((dvoid *)tpcenv, (dvoid
**) &tpscv, OCI_HTYPE_SVCCTX, 0, (dvoid
**));

/* Replaced T.Kato 02.10.24 Retry until
successfully
! OCIAttach(tpcsrv, errhp, (text
*)0,0,OCI_DEFAULT);
*/
for (;;) {
    rval = OCIAttach(tpcsrv, errhp, (text
*)0,0,OCI_DEFAULT);
    if (rval == OCI_SUCCESS || rval ==
OCI_SUCCESS_WITH_INFO)
        break;
    OCIERROR(errhp, rval);
    sleep(1);
}
/* Replaced end */

OCIAttrSet((dvoid *)tpcsrv,
OCI_HTYPE_SVCCTX, (dvoid *)tpcsrv,
(ub4)0,OCI_ATTR_SERVER, errhp);
OCIHandleAlloc((dvoid *)tpcenv, (dvoid
**) &tpcusr, OCI_HTYPE_SESSION, 0, (dvoid
**));
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(tpcsrv,
errhp, tpcusr, OCI_CRED_RDBMS,
OCI_DEFAULT));

OCIAttrSet(tpcsrv, 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);
DBGLOG("INI:[1]Start",0);
OCIERROR(errhp,OCIStmtExecute(tpcsrv,
curi, errhp,1,0,0,0,OCI_DEFAULT));
DBGLOG("INI:[1]End ",0);
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(tpcsrv,
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);
    DBGLOG("INI:[2]Start",0);
    OCIERROR(errhp, OCIStmtExecute(tpcsrv,
curi, errhp,1,0,0,0,OCI_DEFAULT));
    DBGLOG("INI:[2]End ",0);
    OCIHandleFree((dvoid *)curi,
OCI_HTYPE_STMT);
}

logon = 1;

OCIERROR(errhp,OCIDateSysDate(errhp,&cr_d
ate));

if (tkvcninit (0)) { /* new order */
    TPCexit ();
    return (-1);
}
else
    new_init = 1;

if (tkvcpinit (0)) { /* payment */
    TPCexit ();
    return (-1);
}
else
    pay_init = 1;

if (tkvcoint (0)) { /* order status */
    TPCexit ();
    return (-1);
}
else
    ord_init = 1;

#ifdef DEL_ORA8I
if (tkvcdinit (0)) { /* delivery */
    TPCexit ();
    return (-1);
}
else

```

```

    del_init = 1;
#else
if (tkvcdinit (0)) { /* delivery */
    TPCexit ();
    return (-1);
}
else
    del_init_oci = 1;

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

if (tkvcsinit (0)) { /* stock level */
    TPCexit ();
    return (-1);
}
else
    sto_init = 1;

return (0);
}

int TPCnew (struct newstruct *str)
{
/* Added T.Kato 02.11.25 */
#ifdef AVOID_DEADLOCK
    static int
    init_value_index[NITEMS]=(0,1,2,3,4,5,6,7,8,9,
0,11,12,13,14);
#endif
/* Added end */
    int i;

    w_id = str->newin.w_id;
    d_id = str->newin.d_id;
    c_id = str->newin.c_id;

/* Added T.Kato 02.10.24 */
for (i = 0; i < 15; i++) {
    nol_i_id[i] = 0;
    nol_supply_w_id[i] = 0;
    nol_quantity[i] = 0;
}
/* Added end */

for (i = 0; i < 15; i++) {
/* Added T.Kato 02.10.24 */
    if((str->newin.ol_i_id[i] == 0) && (str-
>newin.ol_supply_w_id[i] == 0) && (str-
>newin.ol_quantity[i] == 0))
        break;
/* Added end */
    nol_i_id[i] = str->newin.ol_i_id[i];
    nol_supply_w_id[i] = str-
>newin.ol_supply_w_id[i];
/* Replaced T.kato 03.09.09 Oracle10g tool kit */
/* nol_quantity[i] = str->newin.ol_quantity[i];*/

#ifdef USE_IEEE_NUMBER
    nol_quantity[i] = (float)str-
>newin.ol_quantity[i];
#else
    nol_quantity[i] = str->newin.ol_quantity[i];
#endif
/* Replaced end */
}
}

```

```

retries = 0;

#ifndef AVOID_DEADLOCK

for (i = NITEMS; i > 0; i--) {
    if (no_l_i_id[i-1] > 0) {
        ordl_cnt = i;
        break;
    }
}

/* Replaced T.Kato 02.11.22 */
// for (i = 0; i < NITEMS; i++) indx[i] = i;
memcpy( indx, init_value_index, sizeof(indx) );
/* Replaced End */

q_sort_item(no_l_i_id, str, 0, ordl_cnt-1);

#endif

/*
vgetdate(cr_date); */

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

if ((str->newout.terror = tkvcn ()) != 0) {
    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);
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);
/* Replaced T.Kato 02.11.13 */
#if 0
! str->newout.total_amount = total_amount;
#endif
str->newout.total_amount = 0.0;
/* Replaced end */
for (i = 0; i < o_ol_cnt; i++) {
    strncpy (str->newout.i_name[i], i_name[i],
25);
    str->newout.brand_generic[i] =
brand_generic[i][0];

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! str->newout.s_quantity[i] = s_quantity[i];
! str->newout.i_price[i] = (float)(i_price[i])/100;
! str->newout.ol_amount[i] =
(float)(no_l_amount[i])/100;
#endif

```

```

#ifdef USE_IEEE_NUMBER
str->newout.s_quantity[i] = (int) s_quantity[i];
str->newout.i_price[i] = i_price[i]/100;
str->newout.ol_amount[i] =
no_l_amount[i]/100;
#else
str->newout.s_quantity[i] = s_quantity[i];
str->newout.i_price[i] = (float)(i_price[i])/100;
str->newout.ol_amount[i] =
(float)(no_l_amount[i])/100;
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

/* Added T.Kato 02.11.13 */
str->newout.total_amount += str-
>newout.ol_amount[i];
/* Added end */

}

/* Added T.Kato 03.08.15 */
str->newout.total_amount =
(float)(str->newout.total_amount * (1.0 -
c_discount) * (1.0 + w_tax + d_tax));
/* Added End */
#ifndef AVOID_DEADLOCK
q_sort(indx, str, 0, ordl_cnt-1);
#endif

if (status)
    strcpy (str->newout.status, "Item number is
not valid");
else
    str->newout.status[0] = '0';
str->newout.retry = retries;
return (0);

}

int TPCpay (struct paystruct *str)
{

long double long64bit;

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;

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! h_amount = str->payin.h_amount;
#endif

#ifdef USE_IEEE_NUMBER
h_amount = (float) str->payin.h_amount;
#else
h_amount = str->payin.h_amount;
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

bylastname = str->payin.bylastname;

/* Added T.Kato 03.08.15 */
memset(c_data, 0x00, sizeof(c_data));
/* Added end */

/*
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 = tkvcn ()) != 0) {
    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,(ub1)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,(ub1)strlen(SHORTDATE),(t
ext*)0,0,&sincelen,c_since_d));

str->payout.terror = NOERR;
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);

```

```

/* Replaced T.Kato 03.08.15 */
/*str->payout.c_credit_lim =
(float)(c_credit_lim)/100;*/

    long64bit = (long double)((c_credit_lim / 100.0
+ 0.005555) * 100.0);
    str->payout.c_credit_lim =
(float)((double)long64bit / 100.0);
/* replaced end */

    str->payout.c_discount = c_discount;
/* Replaced T.Kato 03.08.15 */
/*str->payout.c_balance =
(float)(c_balance)/100;*/
    long64bit = (long double)((c_balance / 100.0 +
0.005555) * 100.0);
    str->payout.c_balance =
(float)((double)long64bit / 100.0);
/* Replaced end */
    strncpy(str->payout.c_data, c_data, 201);
    strncpy(str->payout.h_date, (char*)h_date,
20);
    str->payout.retry = retries;
    return (0);
}

int TPCord (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 ()) != 0) {
        if (str->ordout.terror != RECOVER)
            str->ordout.terror = IRRECERR;
        return (-1);
    }

    datelen = sizeof(o_entry_d);
    OCIERROR(errhp,
    OCIDateToText(errhp,&o_entry_d_base,(text*)F
ULLDATE,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";

```

```

/* Replaced by TSL -- BEGIN -- 2006.03.17
adjust data on DB. */
/* if ( !strcmp((char*)ol_delivery_d[i],"15-09-
1911") ) */

    if ( !strcmp((char*)ol_delivery_d[i],"01-01-
1811") )
/* Replaced by TSL -- END -- 2006.03.17 adjust
data on DB. */

        strncpy((char*)ol_delivery_d[i],"NOT
DELIVR",10);
        str->ordout.ol_supply_w_id[i] =
ol_supply_w_id[i];
        str->ordout.ol_i_id[i] = ol_i_id[i];

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! str->ordout.ol_quantity[i] = ol_quantity[i];
! str->ordout.ol_amount[i] =
(float)(ol_amount[i])/100;
#endif

#ifdef USE_IEEE_NUMBER
    str->ordout.ol_quantity[i] = (int) ol_quantity[i];
    str->ordout.ol_amount[i] = ol_amount[i]/100;
#else
    str->ordout.ol_quantity[i] = ol_quantity[i];
    str->ordout.ol_amount[i] =
(float)(ol_amount[i])/100;
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

    strncpy(str->ordout.ol_delivery_d[i],
(char*)ol_delivery_d[i], 11);
}
str->ordout.retry = retries;
return (0);
}

int TPCdel (struct delstruct *str)
{
/* Replaced T.kato 02.10.24 Change the delivery
log writing method */
#if 0
! double tr_end;
! int i;
#endif

    int i;

/* Replaced T.kato 03.12.22 Convert to linux
time. */
#if 0
! SYSTEMTIME systemTime;
! struct tm times;
#else
    struct timeval times;
    int msec;
#endif
/* Replaced end */

    char filename[40];
    //int svrcnt;
/* Replaced end */

/* Added T.Kato 02.10.24 Open the delivery log
file */
    if (iflg == 0)

```

```

{
    // Execute first delivery transaction
    sprintf (filename,
"/home/tpc/dellog/tpcc_%08d.del", (int)getpid());

    if ((lfp = fopen (filename, "w")) == NULL) {
        TpcUserLog (LOG_FILE_INF,
"DELIVERY: Error in TPC-C server %d: Failed to
open %s\n",
            proc_no, filename);
        return (-1);
    }

    // Set first execution indicator
    iflg = 1;
}
/* Added end */

    w_id = str->delin.w_id;
    o_carrier_id = str->delin.o_carrier_id;
    retries = 0;
/*
    vgetdate(cr_date); */

OCIERROR(errhp,OCIDateSysDate(errhp,&cr_d
ate));
#ifdef DEL_ORA8I
    if ((str->delout.terror = tkvcd ()) != 0) {
#else
    if ((str->delout.terror = tkvcd (PLSQLFLAG)) !=
0) { // "PLSQLFLAG" are supplied from
Compile option!!
#endif
        if (str->delout.terror == DEL_ERROR)
            return DEL_ERROR;
        if (str->delout.terror != RECOVER)
            str->delout.terror = IRRECERR;
        return (-1);
    }

/* Replaced T.Kato 02.10.24 Changed time
stamp method */
#if 0
! tr_end = gettime ();
! 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);
#endif

/* Replaced T.Kato 03.12.22 Convert to linux
time. */
#if 0
! GetLocalTime(&systemTime);
! times.tm_year = (int)systemTime.wYear -
1900;
! times.tm_mon = (int)systemTime.wMonth - 1;
! times.tm_mday = (int)systemTime.wDay;
! times.tm_hour = (int)systemTime.wHour;
! times.tm_min = (int)systemTime.wMinute;
! times.tm_sec = (int)systemTime.wSecond;
!
! fprintf(lfp,"%09d%03d%09d%03d %d %d",str-
>delin.startsec,
! str->delin.startusec,((long)mktime
(&times)),(long)systemTime.wMilliseconds,w_id,
o_carrier_id);
/* Replaced end */
#else
/* get system time */
gettimeofday(&times, 0);
msec = times.tv_usec / 1000;

```

```

fprintf(lfp,"%010d%03d %010d%03d %d %d", (int)
)str->delin.startsec,
(int)str->delin.startusec, (int)times.tv_sec,
msec, w_id,o_carrier_id);
#endif
/* Replaced end T.Kato */

for (i = 0; i < 10; i++) {
    fprintf (lfp, " %d %d", i + 1, del_o_id[i]);
    if (del_o_id[i] <= 0) {
        TpcUserLog (LOG_FILE_INF,
"DELIVERY: no new order for w_id: %d,
d_id %d\n",
        w_id, i + 1);
    }
}
fprintf (lfp, " %d\n", retries);
str->delout.terror = NOERR;
str->delout.retry = retries;
return (0);
}

int TPCsto (struct stostruct *str)
{
    w_id = str->stoin.w_id;
    d_id = str->stoin.d_id;

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! threshold = str->stoin.threshold;
#endif

#ifdef USE_IEEE_NUMBER
    threshold = (float) str->stoin.threshold;
#else
    threshold = str->stoin.threshold;
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

    retries = 0;

    if ((str->stoout.terror = tkvcs ()) != 0) {
        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;
    return (0);
}

#ifdef AVOID_DEADLOCK

/* Added T.Kato 02.11.22 */
void q_sort_item(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);
}

```

```

swap_item(str,left,last);
q_sort_item(arr,str,left,last-1);
q_sort_item(arr,str,last+1,right);
}

void swap_item(struct newstruct *str, int i, int j)
{
    int temp;

/* Added T.kato 03.09.09 Oracle10g tool kit */
#ifdef USE_IEEE_NUMBER
    float temp_float;
#endif
/* Added end */

    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;

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! temp = nol_quantity[i];
! nol_quantity[i] = nol_quantity[j];
! nol_quantity[j] = temp;
#endif

#ifdef USE_IEEE_NUMBER
    temp_float = nol_quantity[i];
    nol_quantity[i] = nol_quantity[j];
    nol_quantity[j] = temp_float;
#else
    temp = nol_quantity[i];
    nol_quantity[i] = nol_quantity[j];

    nol_quantity[j] = temp;
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

/* Added T.Kato 03.08.15 */
temp = str->newin.ol_quantity[i];
str->newin.ol_quantity[i] = str-
>newin.ol_quantity[j];
str->newin.ol_quantity[j] = temp;
/* Added End */
}
/* Added end */

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)
{
    int temp;
    char tmpstr[25];
    char tmpch;

/* Added T.Kato 02.11.13 */
    float tmpflt;
/* Added end */

    temp = indx[i];
    indx[i] = indx[j];
    indx[j] = temp;

/* Deleted T.Kato 02.11.22 */
#if 0
! 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 = nol_quantity[i];
! nol_quantity[i] = nol_quantity[j];
! nol_quantity[j] = temp;
#endif
/* Deleted End */

/* Replaced T.Kato 03.03.19 Chenged Oracle
10i tool kit */
#if 0
! strcpy(tmpstr,str->newout.i_name[i]);
! strcpy(str->newout.i_name[i],str-
>newout.i_name[j]);
! strcpy(str->newout.i_name[j],tmpstr);
#endif
    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);
/* Replaced end */

/* Added T.Kato 03.08.15 */
temp = str->newin.ol_quantity[i];
str->newin.ol_quantity[i] = str-
>newin.ol_quantity[j];
str->newin.ol_quantity[j] = temp;
/* Added End */

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;

/* Replaced T.Kato 02.11.13 (int)temp =>
(float)tmpflt */
#if 0
! temp = str->newout.i_price[i];
! str->newout.i_price[i] = str->newout.i_price[j];
! str->newout.i_price[j] = temp;
!
! temp = str->newout.ol_amount[i];
! str->newout.ol_amount[i] = str-
>newout.ol_amount[j];
! str->newout.ol_amount[j] = temp;

```

```
#endif

tmpflt = str->newout.i_price[i];
str->newout.i_price[i] = str->newout.i_price[j];
str->newout.i_price[j] = tmpflt;

tmpflt = str->newout.ol_amount[i];
str->newout.ol_amount[i] = str-
>newout.ol_amount[j];
str->newout.ol_amount[j] = tmpflt;
/* Replaced end */

}

#endif
```


Appendix C: RTE Scripts

```

.....
rte01.conf
.....

#
# rte01.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl001a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl001a
  SUTPORT = 80
  SUTTERM = 1750
  BASENO = 4001
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl001b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 5751
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl001b
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 8001
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl002a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 218401
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl002a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 218901
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl002a
  SUTPORT = 80
  SUTTERM = 1250
  BASENO = 222901
  LOGPATH = /w04

```

```

  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl002b
  SUTPORT = 80
  SUTTERM = 2500
  BASENO = 224151
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl002b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 226651
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl003a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 436801
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl003a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 437301
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl003a
  SUTPORT = 80
  SUTTERM = 1500
  BASENO = 441051
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl003b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 442551
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl003b
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 444801
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl004a
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 655201
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl004a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 655451
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT

```

```

  SUTHOST = cl004a
  SUTPORT = 80
  SUTTERM = 1750
  BASENO = 659201
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl004b
  SUTPORT = 80
  SUTTERM = 2000
  BASENO = 660951
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl004b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 662951
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte02.conf
.....

#
# rte02.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl005a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 873601
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl005a
  SUTPORT = 80
  SUTTERM = 1750
  BASENO = 877601

```

```

LOGPATH = /w01
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl005b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 879351
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl005b
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 881601
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl006a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1092001
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl006a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1092501
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl006a
  SUTPORT = 80
  SUTTERM = 1250
  BASENO = 1096501
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl006b
  SUTPORT = 80
  SUTTERM = 2500
  BASENO = 1097751
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl006b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 1100251
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl007a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1310401
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl007a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1310901
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT

```

```

STARTSUT
  SUTHOST = cl007a
  SUTPORT = 80
  SUTTERM = 1500
  BASENO = 1314651
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl007b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 1316151
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl007b
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1318401
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl008a
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1528801
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl008a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1529051
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl008a
  SUTPORT = 80
  SUTTERM = 1750
  BASENO = 1532801
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl008b
  SUTPORT = 80
  SUTTERM = 2000
  BASENO = 1534551
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl008b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1536551
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010

```

```

DEL-KEYING = 2010
STK-KEYING = 2010
NEW-THINK = 12020
PAY-THINK = 12020
ORD-THINK = 10020
DEL-THINK = 5020
STK-THINK = 5020
CONST-CLAST = 111
CONST-CID = 1023
CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte03.conf
.....

#
# rte03.conf :configuration file for TPC-C
# Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl009a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 11501
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl009a
  SUTPORT = 80
  SUTTERM = 1750
  BASENO = 15501
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl009b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 17251
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl009b
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 19501
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl010a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 229901
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl010a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 230401
  LOGPATH = /w03
  LOGLEVEL = 0

```

```

ENDSUT
STARTSUT
  SUTHOST = cl010a
  SUTPORT = 80
  SUTTERM = 1250
  BASENO = 234401
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl010b
  SUTPORT = 80
  SUTTERM = 2500
  BASENO = 235651
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl010b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 238151
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl011a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 448301
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl011a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 448801
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl011a
  SUTPORT = 80
  SUTTERM = 1500
  BASENO = 452551
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl011b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 454051
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl011b
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 456301
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl012a
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 666701
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl012a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 666951
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl012a
  SUTPORT = 80
  SUTTERM = 1750
  BASENO = 670701
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl012b
  SUTPORT = 80
  SUTTERM = 2000
  BASENO = 672451
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl012b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 674451
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte04.conf
.....

#
# rte04.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl013a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 885101
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl013a
  SUTPORT = 80
  SUTTERM = 1750
  BASENO = 889101
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl013b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 890851
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl013b
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 893101
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl014a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1103501
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl014a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1104001
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl014a
  SUTPORT = 80
  SUTTERM = 1250
  BASENO = 1108001
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl014b
  SUTPORT = 80
  SUTTERM = 2500
  BASENO = 1109251
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl014b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 1111751
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl015a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1321901
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl012a

```

```

SUTHOST = cl015a
SUTPORT = 80
SUTTERM = 3750
BASENO = 1322401
LOGPATH = /w06
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl015a
  SUTPORT = 80
  SUTTERM = 1500
  BASENO = 1326151
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl015b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 1327651
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl015b
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1329901
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl016a
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1540301
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl016a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1540551
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl016a
  SUTPORT = 80
  SUTTERM = 1750
  BASENO = 1544301
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl016b
  SUTPORT = 80
  SUTTERM = 2000
  BASENO = 1546051
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl016b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1548051
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
WAREHOUSE = 174720
MEASUREMENT = 18000

```

```

PAY-MIX = 4302
ORD-MIX = 402
DEL-MIX = 402
STK-MIX = 402
NEW-KEYING = 18010
PAY-KEYING = 3010
ORD-KEYING = 2010
DEL-KEYING = 2010
STK-KEYING = 2010
NEW-THINK = 12020
PAY-THINK = 12020
ORD-THINK = 10020
DEL-THINK = 5020
STK-THINK = 5020
CONST-CLAST = 111
CONST-CID = 1023
CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte05.conf
.....

#
# rte05.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
  STARTSUT
    SUTHOST = cl017a
    SUTPORT = 80
    SUTTERM = 4000
    BASENO = 23001
    LOGPATH = /w00
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl017a
    SUTPORT = 80
    SUTTERM = 1750
    BASENO = 27001
    LOGPATH = /w01
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl017b
    SUTPORT = 80
    SUTTERM = 2250
    BASENO = 28751
    LOGPATH = /w01
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl017b
    SUTPORT = 80
    SUTTERM = 3500
    BASENO = 31001
    LOGPATH = /w02
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl018a
    SUTPORT = 80
    SUTTERM = 500
    BASENO = 241401
    LOGPATH = /w02
    LOGLEVEL = 0
  ENDSUT

```

```

STARTSUT
  SUTHOST = cl018a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 241901
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl018a
  SUTPORT = 80
  SUTTERM = 1250
  BASENO = 245901
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl018b
  SUTPORT = 80
  SUTTERM = 2500
  BASENO = 247151
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl018b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 249651
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl019a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 459801
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl019a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 460301
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl019a
  SUTPORT = 80
  SUTTERM = 1500
  BASENO = 464051
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl019b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 465551
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl019b
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 467801
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl020a
  SUTPORT = 80

```

```

SUTTERM = 250
BASENO = 678201
LOGPATH = /w08
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl020a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 678451
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl020a
  SUTPORT = 80
  SUTTERM = 1750
  BASENO = 682201
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl020b
  SUTPORT = 80
  SUTTERM = 2000
  BASENO = 683951
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl020b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 685951
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....:
rte06.conf
.....:

#
# rte06.conf :configuration file for TPC-C
# Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP

```

```

STARTRTE
STARTSUT
  SUTHOST = cl021a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 896601
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl021a
  SUTPORT = 80
  SUTTERM = 1750
  BASENO = 900601
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl021b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 902351
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl021b
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 904601
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl022a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1115001
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl022a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1115501
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl022a
  SUTPORT = 80
  SUTTERM = 1250
  BASENO = 1119501
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl022b
  SUTPORT = 80
  SUTTERM = 2500
  BASENO = 1120751
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl022b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 1123251
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl023a

```

```

  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1333401
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl023a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1333901
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl023a
  SUTPORT = 80
  SUTTERM = 1500
  BASENO = 1337651
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl023b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 1339151
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl023b
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1341401
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl024a
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1551801
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl024a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1552051
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl024a
  SUTPORT = 80
  SUTTERM = 1750
  BASENO = 1555801
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl024b
  SUTPORT = 80
  SUTTERM = 2000
  BASENO = 1557551
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl024b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1559551

```

```

LOGPATH = /w11
LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
WAREHOUSE = 174720
MEASUREMENT = 18000
PAY-MIX = 4302
ORD-MIX = 402
DEL-MIX = 402
STK-MIX = 402
NEW-KEYING = 18010
PAY-KEYING = 3010
ORD-KEYING = 2010
DEL-KEYING = 2010
STK-KEYING = 2010
NEW-THINK = 12020
PAY-THINK = 12020
ORD-THINK = 10020
DEL-THINK = 5020
STK-THINK = 5020
CONST-CLAST = 111
CONST-CID = 1023
CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte07.conf
.....

#
# rte07.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
SUTHOST = cl025a
SUTPORT = 80
SUTTERM = 4000
BASENO = 34501
LOGPATH = /w00
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl025a
SUTPORT = 80
SUTTERM = 1750
BASENO = 38501
LOGPATH = /w01
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl025b
SUTPORT = 80
SUTTERM = 2250
BASENO = 40251
LOGPATH = /w01
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl025b
SUTPORT = 80
SUTTERM = 3500
BASENO = 42501
LOGPATH = /w02
LOGLEVEL = 0
ENDSUT
STARTSUT

```

```

SUTHOST = cl026a
SUTPORT = 80
SUTTERM = 500
BASENO = 252901
LOGPATH = /w02
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl026a
SUTPORT = 80
SUTTERM = 4000
BASENO = 253401
LOGPATH = /w03
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl026a
SUTPORT = 80
SUTTERM = 1250
BASENO = 257401
LOGPATH = /w04
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl026b
SUTPORT = 80
SUTTERM = 2500
BASENO = 258651
LOGPATH = /w04
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl026b
SUTPORT = 80
SUTTERM = 3250
BASENO = 261151
LOGPATH = /w05
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl027a
SUTPORT = 80
SUTTERM = 500
BASENO = 471301
LOGPATH = /w05
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl027a
SUTPORT = 80
SUTTERM = 3750
BASENO = 471801
LOGPATH = /w06
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl027a
SUTPORT = 80
SUTTERM = 1500
BASENO = 475551
LOGPATH = /w07
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl027b
SUTPORT = 80
SUTTERM = 2250
BASENO = 477051
LOGPATH = /w07
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl027b
SUTPORT = 80
SUTTERM = 3500

```

```

BASENO = 479301
LOGPATH = /w08
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl028a
SUTPORT = 80
SUTTERM = 250
BASENO = 689701
LOGPATH = /w08
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl028a
SUTPORT = 80
SUTTERM = 3750
BASENO = 689951
LOGPATH = /w09
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl028a
SUTPORT = 80
SUTTERM = 1750
BASENO = 693701
LOGPATH = /w10
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl028b
SUTPORT = 80
SUTTERM = 2000
BASENO = 695451
LOGPATH = /w10
LOGLEVEL = 0
ENDSUT
STARTSUT
SUTHOST = cl028b
SUTPORT = 80
SUTTERM = 3750
BASENO = 697451
LOGPATH = /w11
LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
WAREHOUSE = 174720
MEASUREMENT = 18000
PAY-MIX = 4302
ORD-MIX = 402
DEL-MIX = 402
STK-MIX = 402
NEW-KEYING = 18010
PAY-KEYING = 3010
ORD-KEYING = 2010
DEL-KEYING = 2010
STK-KEYING = 2010
NEW-THINK = 12020
PAY-THINK = 12020
ORD-THINK = 10020
DEL-THINK = 5020
STK-THINK = 5020
CONST-CLAST = 111
CONST-CID = 1023
CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte08.conf
.....

```

```
#
# rte08.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#
```

```
STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl029a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 908101
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl029a
  SUTPORT = 80
  SUTTERM = 1750
  BASENO = 912101
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl029b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 913851
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl029b
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 916101
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl030a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1126501
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl030a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1127001
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl030a
  SUTPORT = 80
  SUTTERM = 1250
  BASENO = 1131001
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl030b
  SUTPORT = 80
  SUTTERM = 2500
  BASENO = 1132251
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl030b
  SUTPORT = 80
```

```
SUTTERM = 3250
BASENO = 1134751
LOGPATH = /w05
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl031a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1344901
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl031a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1345401
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl031a
  SUTPORT = 80
  SUTTERM = 1500
  BASENO = 1349151
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl031b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 1350651
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl031b
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1352901
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl032a
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1563301
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl032a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1563551
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl032a
  SUTPORT = 80
  SUTTERM = 1750
  BASENO = 1567301
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl032b
  SUTPORT = 80
  SUTTERM = 2000
  BASENO = 1569051
  LOGPATH = /w10
```

```
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl032b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1571051
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP
```

```
.....
rte09.conf
.....
```

```
#
# rte09.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#
```

```
STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl065a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 954001
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl065a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 958001
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl065a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 962001
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl065b
```

```

SUTPORT = 80
SUTTERM = 500
BASENO = 965501
LOGPATH = /w02
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl065b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 966001
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl065b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 970001
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl065b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 973751
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl066a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1172401
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl066a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1172901
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl066a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1176651
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl066a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1180401
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl066b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1183901
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl066b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1184151

```

```

LOGPATH = /w09
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl066b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1187901
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl066b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1191651
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte10.conf
.....

#
# rte10.conf :configuration file for TPC-C
# Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
  STARTRTE
    STARTSUT
      SUTHOST = cl067a
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 1390801
      LOGPATH = /w00
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl067a
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 1394801
      LOGPATH = /w01
      LOGLEVEL = 0
    ENSUT
  STARTSUT

```

```

SUTHOST = cl067a
SUTPORT = 80
SUTTERM = 3500
BASENO = 1398801
LOGPATH = /w02
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl067b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1402301
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl067b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1402801
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl067b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1406801
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl067b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 1410551
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl069a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1609201
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl069a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1609701
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl069a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1613451
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl069a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1617201
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl069b
  SUTPORT = 80
  SUTTERM = 250

```



```

BASENO = 1620701
LOGPATH = /w08
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl069b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1620951
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl069b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1624701
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl069b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1628451
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte11.conf
.....

#
# rte11.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl033a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 46001
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT

```

```

STARTSUT
  SUTHOST = cl033a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 50001
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl033a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 54001
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl033b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 57501
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl033b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 58001
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl033b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 62001
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl033b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 65751
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl034a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 264401
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl034a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 264901
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl034a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 268651
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl034a
  SUTPORT = 80

```

```

SUTTERM = 3500
BASENO = 272401
LOGPATH = /w08
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl034b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 275901
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl034b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 276151
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl034b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 279901
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl034b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 283651
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte12.conf
.....

#
# rte12.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP

```

```

STARTRTE
STARTSUT
  SUTHOST = cl035a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 482801
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl035a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 486801
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl035a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 490801
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl035b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 494301
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl035b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 494801
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl035b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 498801
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl035b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 502551
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl036a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 701201
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl036a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 701701
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl036a

```

```

SUTPORT = 80
SUTTERM = 3750
BASENO = 705451
LOGPATH = /w07
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl036a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 709201
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl036b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 712701
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl036b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 712951
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl036b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 716701
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl036b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 720451
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

```

```

.....
rte13.conf

```

```

.....
#
# rte13.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl037a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 919601
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl037a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 923601
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl037a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 927601
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl037b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 931101
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl037b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 931601
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl037b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 935601
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl037b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 939351
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1138001
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT

```

```

SUTHOST = cl038a
SUTPORT = 80
SUTTERM = 3750
BASENO = 1138501
LOGPATH = /w06
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1142251
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1146001
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl038b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1149501
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl038b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1149751
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl038b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1153501
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl038b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1157251
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023

```

```

CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0
ENDVARIABLE
ENDGROUP

.....:
rte14.conf
.....:

#
# rte14.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
  STARTRTE
    STARTSUT
      SUTHOST = cl039a
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 1356401
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl039a
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 1360401
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl039a
      SUTPORT = 80
      SUTTERM = 3500
      BASENO = 1364401
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl039b
      SUTPORT = 80
      SUTTERM = 500
      BASENO = 1367901
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl039b
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 1368401
      LOGPATH = /w03
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl039b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 1372401
      LOGPATH = /w04
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl039b
      SUTPORT = 80
      SUTTERM = 3250
      BASENO = 1376151
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT

```

```

STARTSUT
  SUTHOST = cl040a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1574801
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl040a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1575301
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl040a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1579051
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl040a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1582801
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl040b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1586301
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl040b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1586551
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl040b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1590301
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl040b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1594051
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010

```

```

DEL-KEYING = 2010
STK-KEYING = 2010
NEW-THINK = 12020
PAY-THINK = 12020
ORD-THINK = 10020
DEL-THINK = 5020
STK-THINK = 5020
CONST-CLAST = 111
CONST-CID = 1023
CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte15.conf
.....

#
# rte15.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl041a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 69001
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl041a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 73001
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl041a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 77001
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl041b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 80501
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl041b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 81001
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl041b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 85001
  LOGPATH = /w04
  LOGLEVEL = 0

```

```

ENDSUT
STARTSUT
  SUTHOST = cl041b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 88751
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl042a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 287401
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl042a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 287901
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl042a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 291651
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl042a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 295401
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl042b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 298901
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl042b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 299151
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl042b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 302901
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl042b
  SUTPORT = 80
  SUTTERM = 80
  BASENO = 306651
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE

```

```

WAREHOUSE = 174720
MEASUREMENT = 18000
PAY-MIX = 4302
ORD-MIX = 402
DEL-MIX = 402
STK-MIX = 402
NEW-KEYING = 18010
PAY-KEYING = 3010
ORD-KEYING = 2010
DEL-KEYING = 2010
STK-KEYING = 2010
NEW-THINK = 12020
PAY-THINK = 12020
ORD-THINK = 10020
DEL-THINK = 5020
STK-THINK = 5020
CONST-CLAST = 111
CONST-CID = 1023
CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte16.conf
.....

#
# rte16.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl043a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 505801
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl043a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 509801
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl043a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 513801
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl043b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 517301
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl043b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 517801
  LOGPATH = /w03

```

```

LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl043b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 521801
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl043b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 525551
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl044a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 724201
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl044a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 724701
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl044a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 728451
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl044a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 732201
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl044b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 735701
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl044b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 735951
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl044b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 739701
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT

```

```

  SUTHOST = cl044b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 743451
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte18.conf
.....

#
# rte18.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
  STARTSUT
    SUTHOST = cl049a
    SUTPORT = 80
    SUTTERM = 1000
    BASENO = 92001
    LOGPATH = /w00
    LOGLEVEL = 0
  ENSUT
  STARTSUT
    SUTHOST = cl049a
    SUTPORT = 80
    SUTTERM = 1000
    BASENO = 93001
    LOGPATH = /w01
    LOGLEVEL = 0
  ENSUT
  STARTSUT
    SUTHOST = cl049a
    SUTPORT = 80
    SUTTERM = 1000
    BASENO = 94001
    LOGPATH = /w02
    LOGLEVEL = 0
  ENSUT
  STARTSUT
    SUTHOST = cl049a
    SUTPORT = 80
    SUTTERM = 1000
    BASENO = 95001

```

```

  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl049a
  SUTPORT = 80
  SUTTERM = 1000
  BASENO = 96001
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl049a
  SUTPORT = 80
  SUTTERM = 700
  BASENO = 97001
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl049b
  SUTPORT = 80
  SUTTERM = 300
  BASENO = 97701
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl049b
  SUTPORT = 80
  SUTTERM = 1000
  BASENO = 98001
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl049b
  SUTPORT = 80
  SUTTERM = 1000
  BASENO = 99001
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl049b
  SUTPORT = 80
  SUTTERM = 1000
  BASENO = 100001
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl049b
  SUTPORT = 80
  SUTTERM = 1000
  BASENO = 101001
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl049b
  SUTPORT = 80
  SUTTERM = 750
  BASENO = 102001
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl049b
  SUTPORT = 80
  SUTTERM = 650
  BASENO = 102751
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT

```

```

ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

```

```

.....
rte19.conf
.....

```

```

#
# rte19.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

```

```

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl050a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 310401
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl050a
  SUTPORT = 80
  SUTTERM = 1700
  BASENO = 314401
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl050b
  SUTPORT = 80
  SUTTERM = 2300
  BASENO = 316101
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl050b
  SUTPORT = 80
  SUTTERM = 3400
  BASENO = 318401
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl051a
  SUTPORT = 80
  SUTTERM = 600

```

```

BASENO = 528801
LOGPATH = /w02
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl051a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 529401
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl051a
  SUTPORT = 80
  SUTTERM = 1350
  BASENO = 533151
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl051b
  SUTPORT = 80
  SUTTERM = 2400
  BASENO = 534501
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl051b
  SUTPORT = 80
  SUTTERM = 3300
  BASENO = 536901
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl052a
  SUTPORT = 80
  SUTTERM = 450
  BASENO = 747201
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl052a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 747651
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl052a
  SUTPORT = 80
  SUTTERM = 1500
  BASENO = 751401
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl052b
  SUTPORT = 80
  SUTTERM = 2250
  BASENO = 752901
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl052b
  SUTPORT = 80
  SUTTERM = 3450
  BASENO = 755151
  LOGPATH = /w08
  LOGLEVEL = 0

```

```

ENDSUT
STARTSUT
  SUTHOST = cl053a
  SUTPORT = 80
  SUTTERM = 300
  BASENO = 942601
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl053a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 942901
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl053a
  SUTPORT = 80
  SUTTERM = 1650
  BASENO = 946651
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl053b
  SUTPORT = 80
  SUTTERM = 2100
  BASENO = 948301
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl053b
  SUTPORT = 80
  SUTTERM = 3600
  BASENO = 950401
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

```

```

.....
rte20.conf
.....

```

```

#
# rte20.conf :configuration file for TPC-C
Rev3.0

```

```
# Author : mkdef -Auto Configurator for R3-
#
```

```
STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl054a
  SUTPORT = 80
  SUTTERM = 3000
  BASENO = 1161001
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl054a
  SUTPORT = 80
  SUTTERM = 2700
  BASENO = 1164001
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl054b
  SUTPORT = 80
  SUTTERM = 300
  BASENO = 1166701
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl054b
  SUTPORT = 80
  SUTTERM = 3000
  BASENO = 1167001
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl054b
  SUTPORT = 80
  SUTTERM = 2400
  BASENO = 1170001
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl055a
  SUTPORT = 80
  SUTTERM = 600
  BASENO = 1379401
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl055a
  SUTPORT = 80
  SUTTERM = 3000
  BASENO = 1380001
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl055a
  SUTPORT = 80
  SUTTERM = 2100
  BASENO = 1383001
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl055b
  SUTPORT = 80
  SUTTERM = 650
  BASENO = 1385101
  LOGPATH = /w05
```

```
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl055b
  SUTPORT = 80
  SUTTERM = 2750
  BASENO = 1385751
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl055b
  SUTPORT = 80
  SUTTERM = 2300
  BASENO = 1388501
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl056a
  SUTPORT = 80
  SUTTERM = 450
  BASENO = 1597801
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl056a
  SUTPORT = 80
  SUTTERM = 2750
  BASENO = 1598251
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl056a
  SUTPORT = 80
  SUTTERM = 2500
  BASENO = 1601001
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl056b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1603501
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl056b
  SUTPORT = 80
  SUTTERM = 2750
  BASENO = 1603751
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl056b
  SUTPORT = 80
  SUTTERM = 2700
  BASENO = 1606501
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
WAREHOUSE = 174720
MEASUREMENT = 18000
PAY-MIX = 4302
ORD-MIX = 402
DEL-MIX = 402
STK-MIX = 402
NEW-KEYING = 18010
```

```
PAY-KEYING = 3010
ORD-KEYING = 2010
DEL-KEYING = 2010
STK-KEYING = 2010
NEW-THINK = 12020
PAY-THINK = 12020
ORD-THINK = 10020
DEL-THINK = 5020
STK-THINK = 5020
CONST-CLAST = 111
CONST-CID = 1023
CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0
ENDVARIABLE
ENDGROUP
```

```
.....:
rte21.conf
.....:
```

```
#
# rte21.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#
```

```
STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl105a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 977001
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl105a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 981001
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl105a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 985001
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl105b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 988501
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl105b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 989001
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl105b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 993001
```

```

LOGPATH = /w04
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl105b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 996751
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl106a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1195401
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl106a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1195901
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl106a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1199651
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl106a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1203401
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl106b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1206901
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl106b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1207151
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl106b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1210901
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl106b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1214651
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT

```

```

ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte23.conf
.....

#
# rte23.conf :configuration file for TPC-C
# Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl107a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1413801
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl107a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1417801
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl107a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1421801
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl107a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1425301
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl107b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1425301
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl107b
  SUTPORT = 80
  SUTTERM = 4000

```

```

BASENO = 1425801
LOGPATH = /w03
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl107b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1429801
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl107b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 1433551
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl108a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1632201
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl108a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1632701
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl108a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1636451
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl108a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1640201
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl108b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1643701
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl108b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1643951
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl108b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1647701
  LOGPATH = /w10
  LOGLEVEL = 0

```



```

ENDSUT
STARTSUT
  SUTHOST = cl108b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1651451
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte24.conf
.....

#
# rte24.conf :configuration file for TPC-C
# Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl109a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 126401
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl109a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 130401
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl109a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 134401
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl109b
  SUTPORT = 80

```

```

SUTTERM = 500
BASENO = 137901
LOGPATH = /w02
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl109b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 138401
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl109b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 142401
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl109b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 146151
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl110a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 344801
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl110a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 345301
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl110a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 349051
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl110a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 352801
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl110b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 356301
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl110b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 356551
  LOGPATH = /w09

```

```

LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl110b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 360301
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl110b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 364051
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte26.conf
.....

#
# rte26.conf :configuration file for TPC-C
# Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl061a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 103401
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl061a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 107401
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl061a

```

```

SUTPORT = 80
SUTTERM = 3500
BASENO = 111401
LOGPATH = /w02
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl061b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 114901
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl061b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 115401
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl061b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 119401
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl061b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 123151
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl062a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 321801
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl062a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 322301
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl062a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 326051
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl062a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 329801
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl062b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 333301

```

```

LOGPATH = /w08
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl062b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 333551
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl062b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 337301
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl062b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 341051
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte27.conf
.....

#
# rte27.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
  STARTSUT
    SUTHOST = cl063a
    SUTPORT = 80
    SUTTERM = 4000
    BASENO = 540201
    LOGPATH = /w00
    LOGLEVEL = 0
  ENSUT
  STARTSUT

```

```

SUTHOST = cl063a
SUTPORT = 80
SUTTERM = 4000
BASENO = 544201
LOGPATH = /w01
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl063a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 548201
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl063b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 551701
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl063b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 552201
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl063b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 556201
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl063b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 559951
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl064a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 758601
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl064a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 759101
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl064a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 762851
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl064a
  SUTPORT = 80
  SUTTERM = 3500

```

```

BASENO = 766601
LOGPATH = /w08
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl064b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 770101
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl064b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 770351
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl064b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 774101
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl064b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 777851
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte32.conf
.....

#
# rte32.conf :configuration file for TPC-C
# Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE

```

```

STARTSUT
  SUTHOST = cl121a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1023001
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl121a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1027001
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl121a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1031001
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl121b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1034501
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl121b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1035001
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl121b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1039001
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl121b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 1042751
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl122a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1241401
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl122a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1241901
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl122a
  SUTPORT = 80

```

```

  SUTTERM = 3750
  BASENO = 1245651
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl122a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1249401
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl122b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1252901
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl122b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1253151
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl122b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1256901
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl122b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1260651
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte34.conf
.....

```

```

#
# rte34.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl111a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 563201
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl111a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 567201
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl111a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 571201
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl111b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 574701
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl111b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 575201
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl111b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 579201
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl111b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 582951
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl112a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 781601
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl112a

```

```

  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 782101
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl112a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 785851
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl112a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 789601
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl112b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 793101
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl112b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 793351
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl112b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 797101
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl112b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 800851
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191

```

```

  THR-PER-PROC = 250
  SYNC = 0
  ENDVARIABLE
  ENDDGROUP

.....
rte35.conf
.....

#
# rte35.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl113a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1000001
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl113a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1004001
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl113a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1008001
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl113b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1011501
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl113b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1012001
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl113b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1016001
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl113b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 1019751
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT

```

```

SUTHOST = cl114a
SUTPORT = 80
SUTTERM = 500
BASENO = 1218401
LOGPATH = /w05
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl114a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1218901
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl114a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1222651
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl114a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1226401
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl114b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1229901
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl114b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1230151
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl114b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1233901
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl114b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1237651
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010

```

```

STK-KEYING = 2010
NEW-THINK = 12020
PAY-THINK = 12020
ORD-THINK = 10020
DEL-THINK = 5020
STK-THINK = 5020
CONST-CLAST = 111
CONST-CID = 1023
CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte36.conf
.....

#
# rte36.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
  STARTRTE
    STARTSUT
      SUTHOST = cl115a
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 1436801
      LOGPATH = /w00
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl115a
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 1440801
      LOGPATH = /w01
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl115a
      SUTPORT = 80
      SUTTERM = 3500
      BASENO = 1444801
      LOGPATH = /w02
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl115b
      SUTPORT = 80
      SUTTERM = 500
      BASENO = 1448301
      LOGPATH = /w02
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl115b
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 1448801
      LOGPATH = /w03
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl115b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 1452801
      LOGPATH = /w04
      LOGLEVEL = 0
    ENSUT

```

```

STARTSUT
  SUTHOST = cl115b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 1456551
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl116a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1655201
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl116a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1655701
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl116a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1659451
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl116a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1663201
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl116b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1666701
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl116b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1666951
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl116b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1670701
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl116b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1674451
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720

```

```

MEASUREMENT = 18000
PAY-MIX = 4302
ORD-MIX = 402
DEL-MIX = 402
STK-MIX = 402
NEW-KEYING = 18010
PAY-KEYING = 3010
ORD-KEYING = 2010
DEL-KEYING = 2010
STK-KEYING = 2010
NEW-THINK = 12020
PAY-THINK = 12020
ORD-THINK = 10020
DEL-THINK = 5020
STK-THINK = 5020
CONST-CLAST = 111
CONST-CID = 1023
CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte37.conf
.....

#
# rte37.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl117a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 149401
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl117a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 153401
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl117a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 157401
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl117b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 160901
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl117b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 161401
  LOGPATH = /w03
  LOGLEVEL = 0

```

```

ENDSUT
STARTSUT
  SUTHOST = cl117b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 165401
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl117b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 169151
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl118a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 367801
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl118a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 368301
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl118a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 372051
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl118a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 375801
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl118b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 379301
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl118b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 379551
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl118b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 383301
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl118b

```

```

SUTPORT = 80
SUTTERM = 3750
BASENO = 387051
LOGPATH = /w11
LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
WAREHOUSE = 174720
MEASUREMENT = 18000
PAY-MIX = 4302
ORD-MIX = 402
DEL-MIX = 402
STK-MIX = 402
NEW-KEYING = 18010
PAY-KEYING = 3010
ORD-KEYING = 2010
DEL-KEYING = 2010
STK-KEYING = 2010
NEW-THINK = 12020
PAY-THINK = 12020
ORD-THINK = 10020
DEL-THINK = 5020
STK-THINK = 5020
CONST-CLAST = 111
CONST-CID = 1023
CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte38.conf
.....

#
# rte38.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl119a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 586201
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl119a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 590201
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl119a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 594201
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl119b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 597701
  LOGPATH = /w02

```

```

LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl119b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 598201
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl119b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 602201
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl119b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 605951
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl120a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 804601
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl120a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 805101
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl120a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 808851
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl120a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 812601
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl120b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 816101
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl120b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 816351
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT

```

```

SUTHOST = cl120b
SUTPORT = 80
SUTTERM = 3750
BASENO = 820101
LOGPATH = /w10
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl120b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 823851
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte40.conf
.....

#
# rte40.conf :configuration file for TPC-C
# Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl123a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1459801
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl123a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1463801
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl123a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1467801

```

```

LOGPATH = /w02
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl123b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1471301
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl123b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1471801
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl123b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1475801
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl123b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 1479551
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl124a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1678201
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl124a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1678701
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl124a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1682451
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl124a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1686201
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl124b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1689701
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT

```

```

STARTSUT
  SUTHOST = cl124b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1689951
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl124b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1693701
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl124b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1697451
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte41.conf
.....

#
# rte41.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl125a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 172401
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl125a
  SUTPORT = 80
  SUTTERM = 4000

```

```

  BASENO = 176401
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl125a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 180401
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl125b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 183901
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl125b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 184401
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl125b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 188401
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl125b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 192151
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl126a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 390801
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl126a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 391301
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl126a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 395051
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl126a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 398801
  LOGPATH = /w08
  LOGLEVEL = 0

```

```

ENDSUT
STARTSUT
  SUTHOST = cl126b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 402301
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl126b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 402551
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl126b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 406301
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl126b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 410051
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte42.conf
.....

#
# rte42.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl127a
  SUTPORT = 80

```



```

SUTTERM = 4000
BASENO = 609201
LOGPATH = /w00
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl127a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 613201
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl127a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 617201
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl127b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 620701
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl127b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 621201
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl127b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 625201
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl127b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 628951
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl128a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 827601
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl128a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 828101
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl128a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 831851
  LOGPATH = /w07

```

```

LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl128a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 835601
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl128b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 839101
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl128b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 839351
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl128b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 843101
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl128b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 846851
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

```

```

.....
rte43.conf
.....

```

```
#
```

```

# rte43.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
  STARTRTE
    STARTSUT
      SUTHOST = cl129a
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 1046001
      LOGPATH = /w00
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl129a
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 1050001
      LOGPATH = /w01
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl129a
      SUTPORT = 80
      SUTTERM = 3500
      BASENO = 1054001
      LOGPATH = /w02
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl129b
      SUTPORT = 80
      SUTTERM = 500
      BASENO = 1057501
      LOGPATH = /w02
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl129b
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 1058001
      LOGPATH = /w03
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl129b
      SUTPORT = 80
      SUTTERM = 80
      BASENO = 1062001
      LOGPATH = /w04
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl129b
      SUTPORT = 80
      SUTTERM = 3250
      BASENO = 1065751
      LOGPATH = /w05
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl130a
      SUTPORT = 80
      SUTTERM = 500
      BASENO = 1264401
      LOGPATH = /w05
      LOGLEVEL = 0
    ENSUT
    STARTSUT
      SUTHOST = cl130a
      SUTPORT = 80
      SUTTERM = 3750

```

```

BASENO = 1264901
LOGPATH = /w06
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl130a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1268651
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl130a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1272401
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl130b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1275901
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl130b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1276151
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl130b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1279901
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl130b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1283651
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
WAREHOUSE = 174720
MEASUREMENT = 18000
PAY-MIX = 4302
ORD-MIX = 402
DEL-MIX = 402
STK-MIX = 402
NEW-KEYING = 18010
PAY-KEYING = 3010
ORD-KEYING = 2010
DEL-KEYING = 2010
STK-KEYING = 2010
NEW-THINK = 12020
PAY-THINK = 12020
ORD-THINK = 10020
DEL-THINK = 5020
STK-THINK = 5020
CONST-CLAST = 111
CONST-CID = 1023
CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0

```

```

ENDVARIABLE
ENDGROUP
.....
rte44.conf
.....
#
# rte44.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl131a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1482801
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl131a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1486801
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl131a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1490801
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl131b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1494301
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl131b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1494801
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl131b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1498801
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl131b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 1502551
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl132a
  SUTPORT = 80

```

```

SUTTERM = 500
BASENO = 1701201
LOGPATH = /w05
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl132a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1701701
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl132a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1705451
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl132a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1709201
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl132b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1712701
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl132b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1712951
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl132b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1716701
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl132b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1720451
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
WAREHOUSE = 174720
MEASUREMENT = 18000
PAY-MIX = 4302
ORD-MIX = 402
DEL-MIX = 402
STK-MIX = 402
NEW-KEYING = 18010
PAY-KEYING = 3010
ORD-KEYING = 2010
DEL-KEYING = 2010
STK-KEYING = 2010
NEW-THINK = 12020

```

```

PAY-THINK = 12020
ORD-THINK = 10020
DEL-THINK = 5020
STK-THINK = 5020
CONST-CLAST = 111
CONST-CID = 1023
CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte45.conf
.....

#
# rte45.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl133a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 195401
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl133a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 199401
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl133a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 203401
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl133b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 206901
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl133b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 207401
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl133b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 211401
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl133b

```

```

SUTPORT = 80
SUTTERM = 3250
BASENO = 215151
LOGPATH = /w05
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl134a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 413801
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl134a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 414301
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl134a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 418051
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl134a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 421801
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl134b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 425301
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl134b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 425551
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl134b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 429301
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl134b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 433051
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
WAREHOUSE = 174720
MEASUREMENT = 18000
PAY-MIX = 4302

```

```

ORD-MIX = 402
DEL-MIX = 402
STK-MIX = 402
NEW-KEYING = 18010
PAY-KEYING = 3010
ORD-KEYING = 2010
DEL-KEYING = 2010
STK-KEYING = 2010
NEW-THINK = 12020
PAY-THINK = 12020
ORD-THINK = 10020
DEL-THINK = 5020
STK-THINK = 5020
CONST-CLAST = 111
CONST-CID = 1023
CONST-IID = 8191
THR-PER-PROC = 250
SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte46.conf
.....

#
# rte46.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl135a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 632201
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl135a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 636201
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl135a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 640201
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl135b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 643701
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl135b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 644201
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT

```

```

SUTHOST = cl135b
SUTPORT = 80
SUTTERM = 3750
BASENO = 648201
LOGPATH = /w04
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl135b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 651951
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl136a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 850601
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl136a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 851101
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl136a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 854851
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl136a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 858601
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl136b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 862101
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl136b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 862351
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl136b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 866101
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl136b
  SUTPORT = 80
  SUTTERM = 3750

```

```

BASENO = 869851
LOGPATH = /w11
LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

.....
rte47.conf
.....

#
# rte47.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
  STARTRTE
    STARTSUT
      SUTHOST = cl137a
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 1069001
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl137a
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 1073001
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl137a
      SUTPORT = 80
      SUTTERM = 3500
      BASENO = 1077001
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl137b
      SUTPORT = 80
      SUTTERM = 500
      BASENO = 1080501
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT

```

```

STARTSUT
  SUTHOST = cl137b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1081001
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl137b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1085001
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl137b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 1088751
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl138a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1287401
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl138a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1287901
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl138a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1291651
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl138a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1295401
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl138b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1298901
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl138b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1299151
  LOGPATH = /w09
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl138b
  SUTPORT = 80

```

```

SUTTERM = 3750
BASENO = 1302901
LOGPATH = /w10
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl138b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1306651
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

```

.....
rte48.conf
.....

```

#
# rte48.conf :configuration file for TPC-C
# Rev3.0
# Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
STARTRTE
STARTSUT
  SUTHOST = cl139a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1505801
  LOGPATH = /w00
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl139a
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1509801
  LOGPATH = /w01
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl139a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1513801
  LOGPATH = /w02
  LOGLEVEL = 0

```

```

ENDSUT
STARTSUT
  SUTHOST = cl139b
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1517301
  LOGPATH = /w02
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl139b
  SUTPORT = 80
  SUTTERM = 4000
  BASENO = 1517801
  LOGPATH = /w03
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl139b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1521801
  LOGPATH = /w04
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl139b
  SUTPORT = 80
  SUTTERM = 3250
  BASENO = 1525551
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl140a
  SUTPORT = 80
  SUTTERM = 500
  BASENO = 1724201
  LOGPATH = /w05
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl140a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1724701
  LOGPATH = /w06
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl140a
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1728451
  LOGPATH = /w07
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl140a
  SUTPORT = 80
  SUTTERM = 3500
  BASENO = 1732201
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl140b
  SUTPORT = 80
  SUTTERM = 250
  BASENO = 1735701
  LOGPATH = /w08
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl140b

```

```

SUTPORT = 80
SUTTERM = 3750
BASENO = 1735951
LOGPATH = /w09
LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl140b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1739701
  LOGPATH = /w10
  LOGLEVEL = 0
ENDSUT
STARTSUT
  SUTHOST = cl140b
  SUTPORT = 80
  SUTTERM = 3750
  BASENO = 1743451
  LOGPATH = /w11
  LOGLEVEL = 0
ENDSUT
ENDRTE
STARTVARIABLE
  WAREHOUSE = 174720
  MEASUREMENT = 18000
  PAY-MIX = 4302
  ORD-MIX = 402
  DEL-MIX = 402
  STK-MIX = 402
  NEW-KEYING = 18010
  PAY-KEYING = 3010
  ORD-KEYING = 2010
  DEL-KEYING = 2010
  STK-KEYING = 2010
  NEW-THINK = 12020
  PAY-THINK = 12020
  ORD-THINK = 10020
  DEL-THINK = 5020
  STK-THINK = 5020
  CONST-CLAST = 111
  CONST-CID = 1023
  CONST-IID = 8191
  THR-PER-PROC = 250
  SYNC = 0
ENDVARIABLE
ENDGROUP

```

Appendix D: System Tunables

```

=====
(PRIMEQUEST configuration)
=====
[OS tunables]
-----
[/etc/modprobe.conf]
-----
alias eth2 tg3
alias eth3 tg3
alias eth4 tg3
alias eth5 tg3
alias scsi_hostadapter mptbase
alias eth6 tg3
alias eth7 tg3
alias eth8 tg3
alias eth10 tg3
alias eth12 tg3
alias eth14 tg3
alias eth16 tg3
alias eth18 tg3
alias eth20 tg3
alias eth22 tg3
alias scsi_hostadapter1 mptscsih
options lpfc lpfc_lun_queue_depth=30
lpfc_cr_delay=1 lpfc_cr_count=2
alias scsi_hostadapter2 lpfc
alias usb-controller ehci-hcd
alias usb-controller1 uhci-hcd
alias eth35 e1000
alias eth36 e1000

[/etc/sysctl.conf]
-----
# Kernel sysctl configuration file for Red Hat
Linux
#
# For binary values, 0 is disabled, 1 is enabled.
See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging
functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the
PID to the core filename.
# Useful for debugging multi-threaded
applications.
kernel.core_uses_pid = 1
kernel.sem = 100 100000 120 512

kernel.shmmax = 0x400000000
kernel.shmall = 0x20000000
fs.aio-max-nr = 5242880

```

```

vm.nr_hugepages = 7970

[/etc/security/limits.conf]
-----
# /etc/security/limits.conf
#
#Each line describes a limit for a user in the
form:
#
#<domain> <type> <item> <value>
#
#Where:
#<domain> can be:
# - an user name
# - a group name, with @group syntax
# - the wildcard *, for default entry
# - the wildcard %, can be also used
with %group syntax,
# for maxlogin limit
#
#<type> can have the two values:
# - "soft" for enforcing the soft limits
# - "hard" for enforcing hard limits
#
#<item> can be one of the following:
# - core - limits the core file size (KB)
# - data - max data size (KB)
# - fsize - maximum filesize (KB)
# - memlock - max locked-in-memory
address space (KB)
# - nfile - max number of open files
# - rss - max resident set size (KB)
# - stack - max stack size (KB)
# - cpu - max CPU time (MIN)
# - nproc - max number of processes
# - as - address space limit
# - maxlogins - max number of logins for this
user
# - priority - the priority to run user process
with
# - locks - max number of file locks the user
can hold
#
#<domain> <type> <item> <value>
#
#* soft core 0
#* hard rss 10000
#@student hard nproc 20
#@faculty soft nproc 20
#@faculty hard nproc 50
#ftp hard nproc 0
#@student - maxlogins 4

#oracle soft memlock 268435456
#oracle hard memlock 268435456
#oracle soft memlock 1073741824
#oracle hard memlock 1073741824
oracle soft memlock 2147483648
oracle hard memlock 2147483648
oracle soft nfile 4096
oracle hard nfile 65536
#oracle soft nproc 2047
oracle soft nproc 4095
oracle hard nproc 16384

# End of file

[/etc/rc.d/rc.local]
-----
#!/bin/sh

```

```

#
# This script will be executed *after* all the other
init scripts.
# You can put your own initialization stuff in here
if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

echo "100 100000 120 512" >
/proc/sys/kernel/sem
echo 0x20000000 > /proc/sys/kernel/shmall
echo 0xc000000000 > /proc/sys/kernel/shmmax
echo 5242880 > /proc/sys/fs/aio-max-nr

# needed for text and RO data in huge pages
mount none /mnt/hlib -t hugetlbfs
chown -R oracle:dba /mnt/*

/sbin/route add -host cl107 dev eth0
/sbin/route add -host cl108 dev eth10
/sbin/route add -host cl109 dev eth12
/sbin/route add -host cl110 dev eth14

/sbin/route add -host cl111 dev eth16
/sbin/route add -host cl112 dev eth18
/sbin/route add -host cl113 dev eth20
/sbin/route add -host cl114 dev eth22

/sbin/route add -host cl105 dev eth0
/sbin/route add -host cl106 dev eth10
/sbin/route add -host cl115 dev eth12
/sbin/route add -host cl116 dev eth14

/sbin/route add -host cl117 dev eth16
/sbin/route add -host cl118 dev eth18
/sbin/route add -host cl119 dev eth20
/sbin/route add -host cl120 dev eth22

#/sbin/route add -host cl110 dev eth16

#/sbin/route add -host cl111 dev eth20
#/sbin/route add -host cl112 dev eth24
#/sbin/route add -host cl113 dev eth28
#/sbin/route add -host cl114 dev eth32

/usr/sbin/ntpdate fign

[/etc/elilo.conf]
-----
prompt
timeout=20
#default=2.6.9-27-New2.EL
#default=2.6.9-40.EL
#default=2.6.9-40.EL.largesmp
#default=2.6.9-42.EL.largesmp
### Intel kernel for profiling 07/08/08 ###
#default=2.6.9-42-vpa-srat-
jun14_scalable_del_timer-qttools
### Test 32s ###
#default=2.6.9-55.EL.largesmp-vpa-srat-
scalable_del_timer
### 0808 by Ikarashi
default=2.6.9-42.EL.oralargesmp
#default=2.6.9-42.EL.oralargesmp.lpfc
### Mote 32S Audit
#default=2.6.9-42.EL.ora
### Mote 16S Audit
#default=2.6.9-42.EL.ora.lpfc
relocatable

image=vmlinuz-2.6.9-42.EL.oralargesmp
label=2.6.9-42.EL.oralargesmp
initrd=initrd-2.6.9-42.EL.oralargesmp.img

```

```

read-only
append="rhgb root=/dev/sda2
log_buf_len=2M console=ttyS0,19200n8r
elevator=noop ide=nodma
ihash_entries=1000000 dhash_entries=1000000
rhash_entries=500000 thash_entries=100000
ro"

image=vmlinuz-2.6.9-42-vpa-srat-
jun14_scalable_del_timer-qtools
label=2.6.9-42-vpa-srat-
jun14_scalable_del_timer-qtools
initrd=initrd-2.6.9-42-vpa-srat-
jun14_scalable_del_timer-qtools.img
read-only
append="rhgb root=/dev/sda2
log_buf_len=2M console=ttyS0,19200n8r
elevator=noop ide=nodma
ihash_entries=1000000 dhash_entries=1000000
rhash_entries=500000 thash_entries=100000
ro"

image=vmlinuz-2.6.9-55.ELlargesmp-vpa-srat-
scalable_del_timer
label=2.6.9-55.ELlargesmp-vpa-srat-
scalable_del_timer
initrd=initrd-2.6.9-55.ELlargesmp-vpa-srat-
scalable_del_timer.img
read-only
append="rhgb root=/dev/sda2
log_buf_len=2M console=ttyS0,19200n8r
elevator=noop ide=nodma
ihash_entries=1000000 dhash_entries=1000000
rhash_entries=500000 thash_entries=100000
ro"

image=vmlinuz-2.6.9-42.EL.ora
label=2.6.9-42.EL.ora.lpfc
initrd=initrd-2.6.9-42.EL.ora.lpfc.img
read-only
append="rhgb root=/dev/sda2
log_buf_len=2M console=ttyS0,19200n8r
elevator=noop ide=nodma
ihash_entries=1000000 dhash_entries=1000000
rhash_entries=500000 thash_entries=100000
ro"

image=vmlinuz-2.6.9-42.EL.ora
label=2.6.9-42.EL.ora
initrd=initrd-2.6.9-42.EL.ora.img
read-only
append="rhgb root=/dev/sda2
log_buf_len=2M console=ttyS0,19200n8r
elevator=noop ide=nodma
ihash_entries=1000000 dhash_entries=1000000
rhash_entries=500000 thash_entries=100000
ro"

image=vmlinuz-2.6.9-42.ELlargesmp
label=2.6.9-42.ELlargesmp
initrd=initrd-2.6.9-42.ELlargesmp.img
read-only
append="rhgb quiet root=/dev/sda2
log_buf_len=2M console=ttyS0,19200n8r
noirqdebug elevator=noop ide=nodma
ihash_entries=1000000 dhash_entries=1000000
rhash_entries=500000 thash_entries=100000
ro"

image=vmlinuz-2.6.9-40.ELlargesmp
label=2.6.9-40.ELlargesmp
initrd=initrd-2.6.9-40.ELlargesmp.img
read-only
append="rhgb quiet root=/dev/sda2
console=ttyS0,19200n8r noirqdebug

```

```

elevator=noop ide=nodma
ihash_entries=1000000 dhash_entries=1000000
rhash_entries=500000 thash_entries=100000
ro"

image=vmlinuz-2.6.9-40.EL
label=2.6.9-40.EL
initrd=initrd-2.6.9-40.EL.img
read-only
append="rhgb root=/dev/sda2
console=ttyS0,19200n8r elevator=noop
ide=nodma ihash_entries=1000000
dhash_entries=1000000 rhash_entries=500000
thash_entries=100000 ro"

image=vmlinuz-2.6.9-27.EL-vpa-sl
label=RHEL4-u3-vpa-sl
initrd=initrd-2.6.9-27.EL-vpa-sl
read-only
append="rhgb quiet root=/dev/sda2
console=ttyS0,19200n8r noirqdebug
elevator=noop ide=nodma
ihash_entries=1000000 dhash_entries=1000000
rhash_entries=500000 thash_entries=100000"

image=vmlinuz-2.6.9-27.EL
label=2.6.9-27-bcm-30-1-2.EL
initrd=initrd-2.6.9-27-bcm-30-1-2.EL.img
read-only
append="rhgb quiet root=/dev/sda2
console=ttyS0,19200n8r"

image=vmlinuz-2.6.9-27.EL
label=2.6.9-27-New2.EL
initrd=initrd-2.6.9-27.ELtg3.img
read-only
append="rhgb quiet root=/dev/sda2
console=ttyS0,19200n8r noirqdebug
elevator=noop ide=nodma
ihash_entries=1000000 dhash_entries=1000000
rhash_entries=500000 thash_entries=100000
ro"

image=vmlinuz-2.6.9-27.EL
label=2.6.9-27-New.EL
initrd=initrd-2.6.9-27.ELbcm5700.img
read-only
append="rhgb quiet root=/dev/sda2
console=ttyS0,19200n8r noirqdebug
elevator=noop decay=15 ide=nodma
ihash_entries=1000000 dhash_entries=1000000
rhash_entries=500000 thash_entries=100000
ro"

image=vmlinuz-2.6.9-27.EL
label=2.6.9-27-bcm-intel.EL
initrd=initrd-2.6.9-27.ELbcm5700.img
read-only
append="rhgb root=LABEL=/ noirqdebug
elevator=noop decay=15 ide=nodma
ihash_entries=1000000 dhash_entries=1000000
rhash_entries=500000 thash_entries=100000
ro"

image=vmlinuz-2.6.9-27.EL
label=2.6.9-27-bcm.EL
initrd=initrd-2.6.9-27.ELbcm5700.img
read-only
append="rhgb quiet root=/dev/sda2
console=ttyS0 elevator=deadline
thash_entries=4500000
rhash_entries=4500000"

image=vmlinuz-2.6.9-27.EL
label=2.6.9-27-2.EL

```

```

initrd=initrd-2.6.9-27-2.EL.img
read-only
append="rhgb quiet root=/dev/sda2
console=ttyS0,19200n8r"

image=vmlinuz-2.6.9-27.EL
label=2.6.9-27.EL
initrd=initrd-2.6.9-27.EL.img
read-only
append="rhgb quiet root=/dev/sda2
console=ttyS0,19200n8r"

image=vmlinuz-2.6.9-22.EL
label=linux
initrd=initrd-2.6.9-22.EL.img
read-only
append="rhgb quiet root=/dev/sda2
console=ttyS0,19200n8r"

image=vmlinuz-2.6.9-22.EL
label=linux-bcm
initrd=initrd-2.6.9-22.ELbcm5700.img
read-only
append="rhgb quiet root=/dev/sda2
console=ttyS0,19200n8r"

image=vmlinuz-2.6.9-22.EL
label=linux-tg
initrd=initrd-2.6.9-22.ELtg.img
read-only
append="rhgb quiet root=/dev/sda2
console=ttyS0,19200n8r"

```

[Database tunables]

```

-----
:.....:
:.....:
:.....:
p_run.ora
:.....:
:.....:

trace_enabled           = false
control_files           =
(ora_dev/control_001,ora_dev/control_002)
processes               = 3100
sessions                = 4600
transactions            = 5000
db_name                 = tpcc
db_files                = 3806
compatible              = 10.1.0.0.0
dml_locks               = 500
db_block_size           = 2048
remote_login_passwordfile = shared
aq_tm_processes        = 0
db_cache_size           = 15360M
db_keep_cache_size     = 1189016M
db_recycle_cache_size  = 275000M
db_16k_cache_size      = 500096M
db_8k_cache_size       = 2048M
db_4k_cache_size       = 2048M
shared_pool_size       = 50096M
java_pool_size         = 0
disk_asynch_io         = true
db_block_checking      = false
db_block_checksum     = false
undo_management        = auto
undo_retention         = 1
undo_tablespace        = undo_1
transactions_per_rollback_segment = 1
cursor_space_for_time  = true
plsql_optimize_level   = 2
replication_dependency_tracking = false
db_file_multiblock_read_count = 32
fast_start_mttr_target = 0

```

```

parallel_max_servers      = 16
log_buffer                = 33554432
log_checkpoint_interval  = 0
log_checkpoint_timeout    = 1660
log_checkpoints_to_alert = true
timed_statistics         = false
statistics_level          = basic
query_rewrite_enabled     = false

[PRIMERGY tunables]
=====
(cI001 configuration)
=====

[OS tunables]
-----

:~::~:
limits.conf
:~::~:

# /etc/security/limits.conf
#
# Each line describes a limit for a user in the
# form:
#
# <domain> <type> <item> <value>
#
# Where:
# <domain> can be:
#   - an user name
#   - a group name, with @group syntax
#   - the wildcard *, for default entry
#
# <type> can have the two values:
#   - "soft" for enforcing the soft limits
#   - "hard" for enforcing hard limits
#
# <item> can be one of the following:
#   - core - limits the core file size (KB)
#   - data - max data size (KB)
#   - fsize - maximum filesize (KB)
#   - memlock - max locked-in-memory
#         address space (KB)
#   - nofile - max number of open files
#   - rss - max resident set size (KB)
#   - stack - max stack size (KB)
#   - cpu - max CPU time (MIN)
#   - nproc - max number of processes
#   - as - address space limit
#   - maxlogins - max number of logins for this
#         user
#   - priority - the priority to run user process
#         with
#   - locks - max number of file locks the user
#         can hold
#
# <domain> <type> <item> <value>
#
#*          soft core      0
#*          hard  rss      10000
#@student   hard  nproc     20
#@faculty   soft  nproc     20
#@faculty   hard  nproc     50
#ftp        hard  nproc     0
#@student   -    maxlogins   4
tpc - nofile 30000
tpc - nproc  30000

# End of file

:~::~:

```

```

sysctl.conf
:~::~:

# Kernel sysctl configuration file for Red Hat
Linux
#
# For binary values, 0 is disabled, 1 is enabled.
See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Controls the System Request debugging
# functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the
# PID to the core filename.
# Useful for debugging multi-threaded
# applications.
kernel.core_uses_pid = 1

# Change filedescriptor
fs.file-max = 30000

# Change Message queue
kernel.msgmni = 30000
kernel.msgmnb = 1536000

# Change Max process
kernel.threads-max = 30000

# Change Semaphore
kernel.sem = 3000 384000 32 128

# Change TCP/IP backlog
net.ipv4.tcp_max_syn_backlog = 4096

[HTTP server tunables]
-----

:~::~:
apache_cl_start.sh
:~::~:

#!/bin/sh
export
LD_LIBRARY_PATH=${ORACLE_HOME/srvm/li
b:${ORACLE_HOME/lib64}:${ORACLE_HOME/lib}:
/usr/lib:${ORACLE_HOME/rdbms/lib}:${ORACLE_
HOME/network/lib}:${STUXDIR/lib}

ulimit -u 30000
ulimit -s 1536

# /sbin/swaponoff -a

# For 3tier tune
SVRAPL= ps -e | grep tpcfmlw | awk '{print $1}'
/usr/bin/renice -20 -p ${SVRAPL}

rm -f /home/tpc/sar.tmp
/home/tpc/sar.`hostname`
/usr/lib/sa/sadc 5 > /home/tpc/sar.tmp &
# For 3tier tune

apachectl start

```

```

:~::~:
httpd.conf
:~::~:

#
# Based upon the NCSA server configuration
# files originally by Rob McCool.
#
# This is the main Apache server configuration
# file. It contains the
# configuration directives that give the server its
# instructions.
# See <URL:http://httpd.apache.org/docs-2.0/>
# for detailed information about
# the directives.
#
# Do NOT simply read the instructions in here
# without understanding
# what they do. They're here only as hints or
# reminders. If you are unsure
# consult the online docs. You have been
# warned.
#
# The configuration directives are grouped into
# three basic sections:
# 1. Directives that control the operation of the
# Apache server process as a
# whole (the 'global environment').
# 2. Directives that define the parameters of the
# 'main' or 'default' server,
# which responds to requests that aren't
# handled by a virtual host.
# These directives also provide default values
# for the settings
# of all virtual hosts.
# 3. Settings for virtual hosts, which allow Web
# requests to be sent to
# different IP addresses or hostnames and
# have them handled by the
# same Apache server process.
#
# Configuration and log file names: If the
# filenames you specify for many
# of the server's control files begin with "/" (or
# "drive:/" for Win32), the
# server will use that explicit path. If the
# filenames do "not" begin
# with "/", the value of ServerRoot is prepended -
# so "logs/foo.log"
# with ServerRoot set to "/etc/httpd" will be
# interpreted by the
# server as "/etc/httpd/logs/foo.log".
#
### Section 1: Global Environment
#
# The directives in this section affect the overall
# operation of Apache,
# such as the number of concurrent requests it
# can handle or where it
# can find its configuration files.
#
#
# Don't give away too much information about all
# the subcomponents
# we are running. Comment out this line if you
# don't mind remote sites
# finding out what major optional modules you
# are running
#ServerTokens OS
ServerTokens Productly

```



```
#
# ServerRoot: The top of the directory tree under
# which the server's
# configuration, error, and log files are kept.
#
# NOTE! If you intend to place this on an NFS
# (or otherwise network)
# mounted filesystem then please read the
# LockFile documentation
# (available at
# <URL:http://httpd.apache.org/docs-
# 2.0/mod/core.html#lockfile>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory
# path.
#
ServerRoot "/etc/httpd"

#
# ScoreBoardFile: File used to store internal
# server process information.
# If unspecified (the default), the scoreboard will
# be stored in an
# anonymous shared memory segment, and will
# be unavailable to third-party
# applications.
# If specified, ensure that no two invocations of
# Apache share the same
# scoreboard file. The scoreboard file MUST BE
# STORED ON A LOCAL DISK.
#
#ScoreBoardFile run/httpd.scoreboard

#
# PidFile: The file in which the server should
# record its process
# identification number when it starts.
#
PidFile run/httpd.pid

#
# Timeout: The number of seconds before
# receives and sends time out.
#
#Timeout 300
Timeout 999

#
# KeepAlive: Whether or not to allow persistent
# connections (more than
# one request per connection). Set to "Off" to
# deactivate.
#
#KeepAlive Off
KeepAlive On

#
# MaxKeepAliveRequests: The maximum
# number of requests to allow
# during a persistent connection. Set to 0 to
# allow an unlimited amount.
# We recommend you leave this number high,
# for maximum performance.
#
#MaxKeepAliveRequests 100
MaxKeepAliveRequests 0

#
# KeepAliveTimeout: Number of seconds to wait
# for the next request from the
# same client on the same connection.
#
#KeepAliveTimeout 15
```

```
KeepAliveTimeout 999

##
## Server-Pool Size Regulation (MPM specific)
##

# prefork MPM
# StartServers: number of server processes to
# start
# MinSpareServers: minimum number of server
# processes which are kept spare
# MaxSpareServers: maximum number of server
# processes which are kept spare
# MaxClients: maximum number of server
# processes allowed to start
# MaxRequestsPerChild: maximum number of
# requests a server process serves
<IfModule prefork.c>
StartServers 8
MinSpareServers 5
MaxSpareServers 20
MaxClients 150
MaxRequestsPerChild 1000
</IfModule>

# 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 24
ServerLimit 24
ThreadLimit 500
MaxClients 12000
MinSpareThreads 1
MaxSpareThreads 12000
ThreadsPerChild 500
MaxRequestsPerChild 0

#
#
# To reduce memory usage in the worker MPM,
# the thread guard page
#
# To reduce memory usage in the worker MPM,
# the thread guard page
# can be disabled, at the expense of some
# protection against stack
# overflow.
#
#ThreadGuardArea off

</IfModule>

#
# Listen: Allows you to bind Apache to specific
# IP addresses and/or
# ports, in addition to the default. See also the
# <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses
# as shown below to
# prevent Apache from glomming onto all bound
# IP addresses (0.0.0.0)
```

```
# e.g. "Listen 12.34.56.78:80"
#
# To allow connections to IPv6 addresses add
# "Listen [::]:80"
#
Listen 0.0.0.0:80

#
# Dynamic Shared Object (DSO) Support
#

# To be able to use the functionality of a module
# which was built as a DSO you
# have to place corresponding 'LoadModule'
# lines at this location so the
# directives contained in it are actually available
# _before_ they are used.
# Statically compiled modules (those listed by
# `httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule tpmpl_module
modules/mod_tpmpl.so
LoadModule access_module
modules/mod_access.so
LoadModule status_module
modules/mod_status.so
LoadModule alias_module
modules/mod_alias.so
LoadModule cgi_module modules/mod_cgi.so

#
# Load config files from the config directory
# "/etc/httpd/conf.d".
#
#Include conf.d/*.conf

#
# ExtendedStatus controls whether Apache will
# generate "full" status
# information (ExtendedStatus On) or just basic
# information (ExtendedStatus
# Off) when the "server-status" handler is called.
# The default is Off.
#
#ExtendedStatus On

### Section 2: 'Main' server configuration
#
# The directives in this section set up the values
# used by the 'main'
# server, which responds to any requests that
# aren't handled by a
# <VirtualHost> definition. These values also
# provide defaults for
# any <VirtualHost> containers you may define
# later in the file.
#
# All of these directives may appear inside
# <VirtualHost> containers,
# in which case these default settings will be
# overridden for the
# virtual host being defined.
#
#
# 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 apache
#Group apache
User tpc
Group tpc

#
# ServerAdmin: Your address, where problems
with the server should be
# e-mailed. This address appears on some
server-generated pages, such
# as error documents. e.g. admin@your-
domain.com
#
ServerAdmin root@localhost

#
# ServerName gives the name and port that the
server uses to identify itself.
# This can often be determined automatically,
but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If this is not set to valid DNS name for your
host, server-generated
# redirections will not work. See also the
UseCanonicalName directive.
#
# If your host doesn't have a registered DNS
name, enter its IP address here.
# You will have to access it by its address
anyway, and this will make
# redirections work in a sensible way.
#
#ServerName new.host.name:80
ServerName tpccserver:80

#
# UseCanonicalName: Determines how Apache
constructs self-referencing
# URLs and the SERVER_NAME and
SERVER_PORT variables.
# When set "Off", Apache will use the Hostname
and Port supplied
# by the client. When set "On", Apache will use
the value of the
# ServerName directive.
#
UseCanonicalName Off

#
# DocumentRoot: The directory out of which you
will serve your
# documents. By default, all requests are taken
from this directory, but
# symbolic links and aliases may be used to
point to other locations.
#
#DocumentRoot "/var/www/html"

#
# Each directory to which Apache has access

```

```

can be configured with respect
# to which services and features are allowed
and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very
restrictive set of
# features.
#
#<Directory />
# Options FollowSymLinks
# AllowOverride None
#</Directory>

#
# Note that from this point forward you must
specifically allow
# particular features to be enabled - so if
something's not working as
# you might expect, make sure that you have
specifically enabled it
# below.
#
#
# UserDir: The name of the directory that is
appended onto a user's home
# directory if a -user request is received.
#
# The path to the end user account 'public_html'
directory must be
# accessible to the webserver userid. This
usually means that ~userid
# must have permissions of 711,
~userid/public_html must have permissions
# of 755, and documents contained therein must
be world-readable.
# Otherwise, the client will only receive a "403
Forbidden" message.
#
# See also:
http://httpd.apache.org/docs/misc/FAQ.html#forbidden
#
#<IfModule mod_userdir.c>
#
# UserDir is disabled by default since it can
confirm the presence
# of a username on the system (depending on
home directory
# permissions).
#
# UserDir disable

#
# To enable requests to /~user/ to serve the
user's public_html
# directory, remove the "UserDir disable" line
above, and uncomment
# the following line instead:
#
#UserDir public_html

#</IfModule>

#
# Control access to UserDir directories. The
following is an example
# for a site where these directories are restricted
to read-only.
#
#<Directory /home/"public_html">
# AllowOverride FileInfo AuthConfig Limit
# Options MultiViews Indexes
SymLinksIfOwnerMatch IncludesNoExec

```

```

# <Limit GET POST OPTIONS>
# Order allow,deny
# Allow from all
# </Limit>
# <LimitExcept GET POST OPTIONS>
# Order deny,allow
# Deny from all
# </LimitExcept>
#</Directory>

#
# DirectoryIndex: sets the file that Apache will
serve if a directory
# is requested.
#
# The index.html.var file (a type-map) is used to
deliver content-
# negotiated documents. The MultiViews Option
can be used for the
# same purpose, but it is much slower.
#
#
# AccessFileName: The name of the file to look
for in each directory
# for additional configuration directives. See
also the AllowOverride
# directive.
#
AccessFileName .htaccess

#
# The following lines prevent .htaccess
and .htpasswd files from being
# viewed by Web clients.
#
#
# TypesConfig describes where the mime.types
file (or equivalent) is
# to be found.
#
#
# DefaultType is the default MIME type the
server will use for a document
# if it cannot otherwise determine one, such as
from filename extensions.
# If your server contains mostly text or HTML
documents, "text/plain" is
# a good value. If most of your content is binary,
such as applications
# or images, you may want to use
"application/octet-stream" instead to
# keep browsers from trying to display binary
files as though they are
# text.
#
DefaultType text/plain

#
# The mod_mime_magic module allows the
server to use various hints from the
# contents of the file itself to determine its type.
The MIMEMagicFile
# directive tells the module where the hint
definitions are located.
#
#<IfModule mod_mime_magic.c>
## MIMEMagicFile /usr/share/magic.mime
# MIMEMagicFile conf/magic
#</IfModule>

#
# HostnameLookups: Log the names of clients

```

```

or just their IP addresses
# e.g., www.apache.org (on) or 204.62.129.132
(off).
# The default is off because it'd be overall better
for the net if people
# had to knowingly turn this feature on, since
enabling it means that
# each client request will result in AT LEAST one
lookup request to the
# nameserver.
#
HostnameLookups Off

#
# EnableMMAP: Control whether memory-
mapping is used to deliver
# files (assuming that the underlying OS
supports it).
# The default is on; turn this off if you serve from
NFS-mounted
# filesystems. On some systems, turning it off
(regardless of
# filesystem) can improve performance; for
details, please see
# http://httpd.apache.org/docs-
2.0/mod/core.html#enablemmap
#
#EnableMMAP off

#
# EnableSendfile: Control whether the sendfile
kernel support is
# used to deliver files (assuming that the OS
supports it).
# The default is on; turn this off if you serve from
NFS-mounted
# filesystems. Please see
# http://httpd.apache.org/docs-
2.0/mod/core.html#enablesendfile
#
#EnableSendfile off
#

#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive
within a <VirtualHost>
# container, error messages relating to that
virtual host will be
# logged here. If you *do* define an error logfile
for a <VirtualHost>
# container, that host's errors will be logged
there and not here.
#
ErrorLog logs/error_log

#
# LogLevel: Control the number of messages
logged to the error_log.
# Possible values include: debug, info, notice,
warn, error, crit,
# alert, emerg.
#
LogLevel warn

#
# The following directives define some format
nicknames for use with
# a CustomLog directive (see below).
#

#
# The location and format of the access logfile
(Common Logfile Format).
# If you do not define any access logfiles within

```

```

a <VirtualHost>
# container, they will be logged here.
Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles,
transactions will be
# logged therein and *not* in this file.
#
# CustomLog logs/access_log common
#CustomLog logs/access_log combined

#
# If you would like to have agent and referer
logfiles, uncomment the
# following directives.
#
#CustomLog logs/referer_log referer
#CustomLog logs/agent_log agent

#
# If you prefer a single logfile with access, agent,
and referer information
# (Combined Logfile Format) you can use the
following directive.
#
#CustomLog logs/access_log combined

#
# Optionally add a line containing the server
version and virtual host
# name to server-generated pages (error
documents, FTP directory listings,
# mod_status and mod_info output etc., but not
CGI generated documents).
# Set to "EMail" to also include a mailto: link to
the ServerAdmin.
# Set to one of: On | Off | EMail
#
#ServerSignature On
ServerSignature Off

#
# Aliases: Add here as many aliases as you
need (with no limit). The format is
# Alias fakename realname
#
# Note that if you include a trailing / on fakename
then the server will
# require it to be present in the URL. So "/icons"
isn't aliased in this
# example, only "/icons/". If the fakename is
slash-terminated, then the
# realname must also be slash terminated, and if
the fakename omits the
# trailing slash, the realname must also omit it.
#
# We include the /icons/ alias for FancyIndexed
directory listings. If you
# do not use FancyIndexing, you may comment
this out.
#

#
# This should be changed to the
ServerRoot/manual/. The alias provides
# the manual, even if you choose to move your
DocumentRoot. You may comment
# this out if you do not care for the
documentation.
#
#<IfModule mod_dav_fs.c>
# # Location of the WebDAV lock database.
# DAVLockDB /var/lib/dav/lockdb
#</IfModule>

#

```

```

# ScriptAlias: This controls which directories
contain server scripts.
# ScriptAliases are essentially the same as
Aliases, except that
# documents in the realname directory are
treated as applications and
# run by the server when requested rather than
as documents sent to the client.
# The same rules about trailing "/" apply to
ScriptAlias directives as to
# Alias.
#
#ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
ScriptAlias /cgi-bin/ "/home/tpc/tool/bin/"

#
# "/var/www/cgi-bin/" should be changed to
whatever your ScriptAliased
# CGI directory exists, if you have that
configured.
#
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>

#
# Redirect allows you to tell clients about
documents which used to exist in
# your server's namespace, but do not anymore.
This allows you to tell the
# clients where to look for the relocated
document.
# Example:
# Redirect permanent /foo
http://www.example.com/bar

#
# Directives controlling the display of server-
generated directory listings.
#

#
# FancyIndexing is whether you want fancy
directory indexing or standard.
# VersionSort is whether files containing version
numbers should be
# compared in the natural way, so that `apache-
1.3.9.tar' is placed before
# `apache-1.3.12.tar'.
#

#
# AddIcon* directives tell the server which icon
to show for different
# files or filename extensions. These are only
displayed for
# FancyIndexed directories.
#

#
# DefaultIcon is which icon to show for files
which do not have an icon
# explicitly set.
#

#
# AddDescription allows you to place a short
description after a file in
# server-generated indexes. These are only
displayed for FancyIndexed
# directories.

```

```
# Format: AddDescription "description" filename
#
#AddDescription "GZIP compressed
document" .gz
#AddDescription "tar archive" .tar
#AddDescription "GZIP compressed tar
archive" .tgz

#
# ReadmeName is the name of the README file
the server will look for by
# default, and append to directory listings.
#
# HeaderName is the name of a file which
should be prepended to
# directory indexes.

#
# IndexIgnore is a set of filenames which
directory indexing should ignore
# and not include in the listing. Shell-style
wildcarding is permitted.
#
#
# AddEncoding allows you to have certain
browsers (Mosaic/X 2.1+) uncompress
# information on the fly. Note: Not all browsers
support this.
# Despite the name similarity, the following Add*
directives have nothing
# to do with the FancyIndexing customization
directives above.
#
#
# DefaultLanguage and AddLanguage allows
you to specify the language of
# a document. You can then use content
negotiation to give a browser a
# file in a language the user can understand.
#
# Specify a default language. This means that all
data
# going out without a specific language tag (see
below) will
# be marked with this one. You probably do NOT
want to set
# this unless you are sure it is correct for all
cases.
#
# * It is generally better to not mark a page as
# * being a certain language than marking it with
the wrong
# * language!
#
# DefaultLanguage nl
#
# Note 1: The suffix does not have to be the
same as the language
# keyword --- those with documents in Polish
(whose net-standard
# language code is pl) may wish to use
"AddLanguage pl .po" to
# avoid the ambiguity with the common suffix for
perl scripts.
#
# Note 2: The example entries below illustrate
that in some cases
# the two character 'Language' abbreviation is
not identical to
# the two character 'Country' code for its country,
# E.g. 'Danmark/dk' versus 'Danish/da'.
#
# Note 3: In the case of 'ltz' we violate the RFC
```

```
by using a three char
# specifier. There is 'work in progress' to fix this
and get
# the reference data for rfc1766 cleaned up.
#
# Danish (da) - Dutch (nl) - English (en) -
Estonian (et)
# French (fr) - German (de) - Greek-Modern (el)
# Italian (it) - Norwegian (no) - Norwegian
Nynorsk (nn) - Korean (ko)
# Portugese (pt) - Luxembourgish* (ltz)
# Spanish (es) - Swedish (sv) - Catalan (ca) -
Czech(cs)
# Polish (pl) - Brazilian Portuguese (pt-br) -
Japanese (ja)
# Russian (ru) - Croatian (hr)
#
#
# LanguagePriority allows you to give
precedence to some languages
# in case of a tie during content negotiation.
#
# Just list the languages in decreasing order of
preference. We have
# more or less alphabetized them here. You
probably want to change this.
#
#
# ForceLanguagePriority allows you to serve a
result page rather than
# MULTIPLE CHOICES (Prefer) [in case of a tie]
or NOT ACCEPTABLE (Fallback)
# [in case no accepted languages matched the
available variants]
#
#
# Specify a default charset for all pages sent out.
This is
# always a good idea and opens the door for
future internationalisation
# of your web site, should you ever want it.
Specifying it as
# a default does little harm; as the standard
dictates that a page
# is in iso-8859-1 (latin1) unless specified
otherwise i.e. you
# are merely stating the obvious. There are also
some security
# reasons in browsers, related to javascript and
URL parsing
# which encourage you to always set a default
charset.
#
AddDefaultCharset UTF-8
#
#
# Commonly used filename extensions to
character sets. You probably
# want to avoid clashes with the language
extensions, unless you
# are good at carefully testing your setup after
each change.
# See
http://www.iana.org/assignments/character-sets
for the
# official list of charset names and their
respective RFCs
#
#
# AddType allows you to add to or override the
MIME configuration
```

```
# file mime.types for specific file types.
#
#
# AddHandler allows you to map certain file
extensions to "handlers":
# actions unrelated to filetype. These can be
either built into the server
# or added with the Action directive (see below)
#
# To use CGI scripts outside of ScriptAliased
directories:
# (You will also need to add "ExecCGI" to the
"Options" directive.)
#
AddHandler cgi-script .cgi
#
#
# For files that include their own HTTP headers:
#
AddHandler send-as-is asis
#
#
# For server-parsed imagemap files:
#
#
# For type maps (negotiated resources):
# (This is enabled by default to allow the Apache
"It Worked" page
# to be distributed in multiple languages.)
#
#
# Filters allow you to process content before it is
sent to the client.
#
# To parse .shtml files for server-side includes
(SSl):
# (You will also need to add "Includes" to the
"Options" directive.)
#
#
# Action lets you define media types that will
execute a script whenever
# a matching file is called. This eliminates the
need for repeated URL
# pathnames for oft-used CGI file processors.
# Format: Action media/type /cgi-script/location
# Format: Action handler-name /cgi-
script/location
#
#
# Customizable error responses come in three
flavors:
# 1) plain text 2) local redirects 3) external
redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo
boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-
bin/missing_handler.pl"
#ErrorDocument 402
http://www.example.com/subscription\_info.html
#
#
# Putting this all together, we can
Internationalize error responses.
#
# We use Alias to redirect any
/error/HTTP_<error>.html var response to
```

```
# our collection of by-error message multi-
language collections. We use
# includes to substitute the appropriate text.
#
# You can modify the messages' appearance
without changing any of the
# default HTTP_<error>.html.var files by adding
the line;
#
# Alias /error/include/ "/your/include/path/"
#
# which allows you to create your own set of files
by starting with the
# /var/www/error/include/ files and
# copying them to /your/include/path/, even on a
per-VirtualHost basis.
#
```

```
Alias /error/ "/var/www/error/"
```

```
# ErrorDocument 400
/error/HTTP_BAD_REQUEST.html.var
# ErrorDocument 401
/error/HTTP_UNAUTHORIZED.html.var
# ErrorDocument 403
/error/HTTP_FORBIDDEN.html.var
# ErrorDocument 404
/error/HTTP_NOT_FOUND.html.var
# ErrorDocument 405
/error/HTTP_METHOD_NOT_ALLOWED.html.v
ar
# ErrorDocument 408
/error/HTTP_REQUEST_TIME_OUT.html.var
# ErrorDocument 410
/error/HTTP_GONE.html.var
# ErrorDocument 411
/error/HTTP_LENGTH_REQUIRED.html.var
# ErrorDocument 412
/error/HTTP_PRECONDITION_FAILED.html.var
# ErrorDocument 413
/error/HTTP_REQUEST_ENTITY_TOO_LARGE
.html.var
# ErrorDocument 414
/error/HTTP_REQUEST_URI_TOO_LARGE.htm
l.var
# ErrorDocument 415
/error/HTTP_SERVICE_UNAVAILABLE.html.var
# ErrorDocument 500
/error/HTTP_INTERNAL_SERVER_ERROR.htm
l.var
# ErrorDocument 501
/error/HTTP_NOT_IMPLEMENTED.html.var
# ErrorDocument 502
/error/HTTP_BAD_GATEWAY.html.var
# ErrorDocument 503
/error/HTTP_SERVICE_UNAVAILABLE.html.var
# ErrorDocument 506
/error/HTTP_VARIANT_ALSO_VARIES.html.var
```

```
#
# The following directives modify normal HTTP
response behavior to
# handle known problems with browser
implementations.
#
```

```
#
# The following directive disables redirects on
non-GET requests for
# a directory that does not include the trailing
slash. This fixes a
# problem with Microsoft WebFolders which
does not appropriately handle
```

```
# redirects for folders with DAV methods.
# Same deal with Apple's DAV filesystem and
Gnome VFS support for DAV.
#
# Allow server status reports, with the URL of
http://servername/server-status
# Change the ".your-domain.com" to match your
domain to enable.
#
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 192.168.
</Location>
```

```
#
# Allow remote server configuration reports, with
the URL of
# http://servername/server-info (requires that
mod_info.c be loaded).
# Change the ".example.com" to match your
domain to enable.
#
#<Location /server-info>
# SetHandler server-info
# Order deny,allow
# Deny from all
# Allow from .example.com
#</Location>
```

```
#
# Proxy Server directives. Uncomment the
following lines to
# enable the proxy server:
#
#<IfModule mod_proxy.c>
#ProxyRequests On
#
#<Proxy *>
# Order deny,allow
# Deny from all
# Allow from .example.com
#</Proxy>
```

```
#
# Enable/disable the handling of HTTP/1.1 "Via:"
headers.
# ("Full" adds the server version; "Block"
removes all outgoing Via: headers)
# Set to one of: Off | On | Full | Block
#
#ProxyVia On
```

```
#
# To enable a cache of proxied content,
uncomment the following lines.
# See http://httpd.apache.org/docs-
2.0/mod/mod_cache.html for more details.
#
#<IfModule mod_disk_cache.c>
# CacheEnable disk /
# CacheRoot "/var/cache/mod_proxy"
#</IfModule>
#
```

```
#</IfModule>
# End of proxy directives.
```

```
### Section 3: Virtual Hosts
#
# VirtualHost: If you want to maintain multiple
domains/hostnames on your
# machine you can setup VirtualHost containers
for them. Most configurations
```

```
# use only name-based virtual hosts so the
server doesn't need to worry about
# IP addresses. This is indicated by the asterisks
in the directives below.
#
```

```
# Please see the documentation at
# <URL:http://httpd.apache.org/docs-
2.0/vhosts/>
# for further details before you try to setup virtual
hosts.
#
# You may use the command line option '-S' to
verify your virtual host
# configuration.
```

```
#
# Use name-based virtual hosting.
#
#NameVirtualHost *:80
```

```
#
# VirtualHost example:
# Almost any Apache directive may go into a
VirtualHost container.
# The first VirtualHost section is used for
requests without a known
# server name.
#
#<VirtualHost *>
# ServerAdmin webmaster@dummy-
host.example.com
# DocumentRoot /www/docs/dummy-
host.example.com
# ServerName dummy-host.example.com
# ErrorLog logs/dummy-host.example.com-
error_log
# CustomLog logs/dummy-host.example.com-
access_log common
#</VirtualHost>
```

```
#
# For TPAPL
#
<Location /tpapl>
    SetHandler tpapl
    TpaPlConf /home/tpc/conf/tpapl.conf
</Location>
```

```
[Front-end application tunables]
```

```
.....
tpapl.conf
.....
```

```
[TPAPL_INFO]
Term_Base="1"
NumWarehouses="174720"
MaxUsers="1747200"
MaxTerm of Client="11500"
CONTROL_Flag="0"
LogPath="/home/tpc/log/userlog.log"
```

```
[SVRAPL_INFO]
LogPath="/home/tpc/log/DBDepend_Userlog.log"
"
```

```
<< for Linux Client >>
```

```

.....
tnsnames.ora
.....

#
# Installation Generated Net8 Configuration
# Version Date: Oct-27-97
# Filename: Tnsnames.ora
#
extproc_connection_data =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = IPC)(KEY =
tpcc))
    (SDU=14600)
    (CONNECT_DATA = (SERVICE_NAME =
tpcc))
  )

tpcc =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
    (SDU=14600)
    (CONNECT_DATA = (SERVICE_NAME =
tpcc))
  )

[TP monitor tunables]
-----
.....
ubbcnfig
.....

#
# ubbcnfig : TUXEDO configuration file-
@(WAREHOUSE BINED)
#

*RESOURCES
IPCKEY      211940
MASTER     SITE1
UID         500
GID         500
PERM        0660
MAXACCESSERS 1000
MAXSERVERS 100
MAXSERVICES 100
MAXGTT      0
MODEL       SHM
LDBAL       Y
OPTIONS     NO_AA,NO_XA

*MACHINES
cl001 LMID=SITE1
  APPDIR="/home/tpc/bin"
  TUXCONFIG="/home/tpc/conf/tuxconfig"
  TUXDIR="/usr/local/BEA/tuxedo8.1"
  ULOGPFX="/home/tpc/log/tuxedo.log"
  SICACHEENTRIESMAX="0"

*GROUPS
group1 LMID=SITE1 GRPNO=1

*SERVERS
DEFAULT:  RESTART=Y MAXGEN=5
REPLYQ=N RQPERM=0660

tpccfm1w SRVGRP=group1 RQADDR=ware01
SRVID=1 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"

```

```

tpccfm1w SRVGRP=group1 RQADDR=ware02
SRVID=2 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfm1w SRVGRP=group1 RQADDR=ware03
SRVID=3 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfm1w SRVGRP=group1 RQADDR=ware04
SRVID=4 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfm1w SRVGRP=group1 RQADDR=ware05
SRVID=5 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfm1w SRVGRP=group1 RQADDR=ware06
SRVID=6 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfm1w SRVGRP=group1 RQADDR=ware07
SRVID=7 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfm1w SRVGRP=group1 RQADDR=ware08
SRVID=8 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfm1w SRVGRP=group1 RQADDR=ware09
SRVID=9 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"

*SERVICES
"OPSTUXSERVER" TRANTIME=0
SRVGRP=group1

*ROUTING

=====
(diff between cl001 and cl002 configuration)
=====
980c980
< Term_Base="1"
---
> Term_Base="218401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
1048c1048
< cl001 LMID=SITE1
---
> cl002 LMID=SITE1

=====
(diff between cl002 and cl003 configuration)
=====
980c980
< Term_Base="218401"
---
> Term_Base="436801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
1048c1048
< cl002 LMID=SITE1
---
> cl003 LMID=SITE1

=====
(diff between cl003 and cl004 configuration)
=====
980c980
< Term_Base="436801"
---

```

```

> Term_Base="655201"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
1048c1048
< cl003 LMID=SITE1
---
> cl004 LMID=SITE1

=====
(diff between cl004 and cl005 configuration)
=====
980c980
< Term_Base="655201"
---
> Term_Base="873601"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
1048c1048
< cl004 LMID=SITE1
---
> cl005 LMID=SITE1

=====
(diff between cl005 and cl006 configuration)
=====
980c980
< Term_Base="873601"
---
> Term_Base="1092001"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
1048c1048
< cl005 LMID=SITE1
---
> cl006 LMID=SITE1

=====
(diff between cl006 and cl007 configuration)
=====
980c980
< Term_Base="1092001"
---
> Term_Base="1310401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
1048c1048
< cl006 LMID=SITE1
---
> cl007 LMID=SITE1

=====
(diff between cl007 and cl008 configuration)
=====
980c980
< Term_Base="1310401"
---
> Term_Base="1528801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=

```

```

pqtpc_g)(Port= 1527))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
1048c1048
< cl007 LMID=SITE1
---
> cl008 LMID=SITE1

=====
(diff between cl008 and cl009 configuration)
=====
980c980
< Term_Base="1528801"
---
> Term_Base="11501"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
1048c1048
< cl008 LMID=SITE1
---
> cl009 LMID=SITE1

=====
(diff between cl009 and cl010 configuration)
=====
980c980
< Term_Base="11501"
---
> Term_Base="229901"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
1048c1048
< cl009 LMID=SITE1
---
> cl010 LMID=SITE1

=====
(diff between cl010 and cl011 configuration)
=====
980c980
< Term_Base="229901"
---
> Term_Base="448301"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
1048c1048
< cl010 LMID=SITE1
---
> cl011 LMID=SITE1

=====
(diff between cl011 and cl012 configuration)
=====
980c980
< Term_Base="448301"
---
> Term_Base="666701"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=

```

```

pqtpc_d)(Port= 1524))
1048c1048
< cl011 LMID=SITE1
---
> cl012 LMID=SITE1

=====
(diff between cl012 and cl013 configuration)
=====
980c980
< Term_Base="666701"
---
> Term_Base="885101"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
1048c1048
< cl012 LMID=SITE1
---
> cl013 LMID=SITE1

=====
(diff between cl013 and cl014 configuration)
=====
980c980
< Term_Base="885101"
---
> Term_Base="1103501"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
1048c1048
< cl013 LMID=SITE1
---
> cl014 LMID=SITE1

=====
(diff between cl014 and cl015 configuration)
=====
980c980
< Term_Base="1103501"
---
> Term_Base="1321901"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
1048c1048
< cl014 LMID=SITE1
---
> cl015 LMID=SITE1

=====
(diff between cl015 and cl016 configuration)
=====
980c980
< Term_Base="1321901"
---
> Term_Base="1540301"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
1048c1048
< cl015 LMID=SITE1

```

```

---
> cl016 LMID=SITE1

=====
(diff between cl016 and cl017 configuration)
=====
980c980
< Term_Base="1540301"
---
> Term_Base="23001"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
1048c1048
< cl016 LMID=SITE1
---
> cl017 LMID=SITE1

=====
(diff between cl017 and cl018 configuration)
=====
980c980
< Term_Base="23001"
---
> Term_Base="241401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
1048c1048
< cl017 LMID=SITE1
---
> cl018 LMID=SITE1

=====
(diff between cl018 and cl019 configuration)
=====
980c980
< Term_Base="241401"
---
> Term_Base="459801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
1048c1048
< cl018 LMID=SITE1
---
> cl019 LMID=SITE1

=====
(diff between cl019 and cl020 configuration)
=====
980c980
< Term_Base="459801"
---
> Term_Base="678201"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
1048c1048
< cl019 LMID=SITE1
---
> cl020 LMID=SITE1

```

```

=====
(diff between cl020 and cl021 configuration)
=====
980c980
< Term_Base="678201"
...
> Term_Base="896601"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
1048c1048
< cl020 LMID=SITE1
...
> cl021 LMID=SITE1

=====
(diff between cl021 and cl022 configuration)
=====
980c980
< Term_Base="896601"
...
> Term_Base="1115001"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
1048c1048
< cl021 LMID=SITE1
...
> cl022 LMID=SITE1

=====
(diff between cl022 and cl023 configuration)
=====
980c980
< Term_Base="1115001"
...
> Term_Base="1333401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
1048c1048
< cl022 LMID=SITE1
...
> cl023 LMID=SITE1

=====
(diff between cl023 and cl024 configuration)
=====
980c980
< Term_Base="1333401"
...
> Term_Base="1551801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
1048c1048
< cl023 LMID=SITE1
...
> cl024 LMID=SITE1

=====
(diff between cl024 and cl025 configuration)
=====

```

```

980c980
< Term_Base="1551801"
...
> Term_Base="34501"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
1048c1048
< cl024 LMID=SITE1
...
> cl025 LMID=SITE1

=====
(diff between cl025 and cl026 configuration)
=====
980c980
< Term_Base="34501"
...
> Term_Base="252901"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
1048c1048
< cl025 LMID=SITE1
...
> cl026 LMID=SITE1

=====
(diff between cl026 and cl027 configuration)
=====
980c980
< Term_Base="252901"
...
> Term_Base="471301"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
1048c1048
< cl026 LMID=SITE1
...
> cl027 LMID=SITE1

=====
(diff between cl027 and cl028 configuration)
=====
980c980
< Term_Base="471301"
...
> Term_Base="689701"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
1048c1048
< cl027 LMID=SITE1
...
> cl028 LMID=SITE1

=====
(diff between cl028 and cl029 configuration)
=====
980c980
< Term_Base="689701"
...

```

```

> Term_Base="908101"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
1048c1048
< cl028 LMID=SITE1
...
> cl029 LMID=SITE1

=====
(diff between cl029 and cl030 configuration)
=====
980c980
< Term_Base="908101"
...
> Term_Base="1126501"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
1048c1048
< cl029 LMID=SITE1
...
> cl030 LMID=SITE1

=====
(diff between cl030 and cl031 configuration)
=====
980c980
< Term_Base="1126501"
...
> Term_Base="1344901"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
1048c1048
< cl030 LMID=SITE1
...
> cl031 LMID=SITE1

=====
(diff between cl031 and cl032 configuration)
=====
980c980
< Term_Base="1344901"
...
> Term_Base="1563301"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
1048c1048
< cl031 LMID=SITE1
...
> cl032 LMID=SITE1

=====
(diff between cl032 and cl033 configuration)
=====
258,259c258,259
< StartServers 24
< ServerLimit 24
...
> StartServers 47
> ServerLimit 47

```



```

261c261
< MaxClients      12000
...
> MaxClients      23500
263c263
< MaxSpareThreads 12000
...
> MaxSpareThreads 23500
980c980
< Term_Base="1563301"
...
> Term_Base="46001"
983c983
< MaxTerm of Client="11500"
...
> MaxTerm of Client="23000"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
1048c1048
< cl032 LMID=SITE1
...
> cl033 LMID=SITE1
1069a1070,1078
> tpccfmlw SRVGRP=group1
RQADDR=ware10 SRVID=10 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware11 SRVID=11 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware12 SRVID=12 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware13 SRVID=13 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware14 SRVID=14 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware15 SRVID=15 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware16 SRVID=16 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware17 SRVID=17 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware18 SRVID=18 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
=====
(diff between cl033 and cl034 configuration)
=====
980c980
< Term_Base="46001"
...
> Term_Base="264401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
...

```

```

> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
1048c1048
< cl033 LMID=SITE1
...
> cl034 LMID=SITE1
=====
(diff between cl034 and cl035 configuration)
=====
980c980
< Term_Base="264401"
...
> Term_Base="482801"
999d999
<
1014c1013
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
1048c1047
< cl034 LMID=SITE1
...
> cl035 LMID=SITE1
=====
(diff between cl035 and cl036 configuration)
=====
980c980
< Term_Base="482801"
...
> Term_Base="701201"
998a999
>
1013c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
1047c1048
< cl035 LMID=SITE1
...
> cl036 LMID=SITE1
=====
(diff between cl036 and cl037 configuration)
=====
980c980
< Term_Base="701201"
...
> Term_Base="919601"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
1048c1048
< cl036 LMID=SITE1
...
> cl037 LMID=SITE1
=====
(diff between cl037 and cl038 configuration)
=====
980c980
< Term_Base="919601"
...
> Term_Base="1138001"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))

```

```

...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
1048c1048
< cl037 LMID=SITE1
...
> cl038 LMID=SITE1
=====
(diff between cl038 and cl039 configuration)
=====
980c980
< Term_Base="1138001"
...
> Term_Base="1356401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
1048c1048
< cl038 LMID=SITE1
...
> cl039 LMID=SITE1
=====
(diff between cl039 and cl040 configuration)
=====
980c980
< Term_Base="1356401"
...
> Term_Base="1574801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
1048c1048
< cl039 LMID=SITE1
...
> cl040 LMID=SITE1
=====
(diff between cl040 and cl041 configuration)
=====
980c980
< Term_Base="1574801"
...
> Term_Base="69001"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
1048c1048
< cl040 LMID=SITE1
...
> cl041 LMID=SITE1
=====
(diff between cl041 and cl042 configuration)
=====
980c980
< Term_Base="69001"
...
> Term_Base="287401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))

```

```

1048c1048
< cl041 LMID=SITE1
...
> cl042 LMID=SITE1

=====
(diff between cl042 and cl043 configuration)
=====
980c980
< Term_Base="287401"
...
> Term_Base="505801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
1048c1048
< cl042 LMID=SITE1
...
> cl043 LMID=SITE1

=====
(diff between cl043 and cl044 configuration)
=====
980c980
< Term_Base="505801"
...
> Term_Base="724201"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
1048c1048
< cl043 LMID=SITE1
...
> cl044 LMID=SITE1

=====
(diff between cl044 and cl049 configuration)
=====
258,259c258,259
< StartServers 47
< ServerLimit 47
...
> StartServers 23
> ServerLimit 23
261c261
< MaxClients 23500
...
> MaxClients 11500
263c263
< MaxSpareThreads 23500
...
> MaxSpareThreads 11500
980c980
< Term_Base="724201"
...
> Term_Base="92001"
983c983
< MaxTerm of Client="23000"
...
> MaxTerm of Client="11400"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
1048c1048
< cl044 LMID=SITE1
...

```

```

> cl049 LMID=SITE1
1070,1078d1069
< tpccfmlw SRVGRP=group1
ROADDR=ware10 SRVID=10 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
< tpccfmlw SRVGRP=group1
ROADDR=ware11 SRVID=11 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
< tpccfmlw SRVGRP=group1
ROADDR=ware12 SRVID=12 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
< tpccfmlw SRVGRP=group1
ROADDR=ware13 SRVID=13 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
< tpccfmlw SRVGRP=group1
ROADDR=ware14 SRVID=14 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
< tpccfmlw SRVGRP=group1
ROADDR=ware15 SRVID=15 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
< tpccfmlw SRVGRP=group1
ROADDR=ware16 SRVID=16 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
< tpccfmlw SRVGRP=group1
ROADDR=ware17 SRVID=17 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
< tpccfmlw SRVGRP=group1
ROADDR=ware18 SRVID=18 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"

=====
(diff between cl049 and cl050 configuration)
=====
980c980
< Term_Base="92001"
...
> Term_Base="310401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
1048c1048
< cl049 LMID=SITE1
...
> cl050 LMID=SITE1

=====
(diff between cl050 and cl051 configuration)
=====
980c980
< Term_Base="310401"
...
> Term_Base="528801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
1048c1048
< cl050 LMID=SITE1
...
> cl051 LMID=SITE1

```

```

=====
(diff between cl051 and cl052 configuration)
=====
980c980
< Term_Base="528801"
...
> Term_Base="747201"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
1048c1048
< cl051 LMID=SITE1
...
> cl052 LMID=SITE1

=====
(diff between cl052 and cl053 configuration)
=====
980c980
< Term_Base="747201"
...
> Term_Base="942601"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
1048c1048
< cl052 LMID=SITE1
...
> cl053 LMID=SITE1

=====
(diff between cl053 and cl054 configuration)
=====
980c980
< Term_Base="942601"
...
> Term_Base="1161001"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
1048c1048
< cl053 LMID=SITE1
...
> cl054 LMID=SITE1

=====
(diff between cl054 and cl055 configuration)
=====
980c980
< Term_Base="1161001"
...
> Term_Base="1379401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
1048c1048
< cl054 LMID=SITE1
...
> cl055 LMID=SITE1

=====
(diff between cl055 and cl056 configuration)
=====

```

```

980c980
< Term_Base="1379401"
...
> Term_Base="1597801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
1048c1048
< cl055 LMID=SITE1
...
> cl056 LMID=SITE1

=====
(diff between cl056 and cl061 configuration)
=====
258,259c258,259
< StartServers 23
< ServerLimit 23
...
> StartServers 47
> ServerLimit 47
261c261
< MaxClients 11500
...
> MaxClients 23500
263c263
< MaxSpareThreads 11500
...
> MaxSpareThreads 23500
980c980
< Term_Base="1597801"
...
> Term_Base="103401"
983c983
< MaxTerm of Client="11400"
...
> MaxTerm of Client="23000"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
1048c1048
< cl056 LMID=SITE1
...
> cl061 LMID=SITE1
1069a1070,1078
> tpccfmlw SRVGRP=group1
RQADDR=ware10 SRVID=10 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware11 SRVID=11 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware12 SRVID=12 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware13 SRVID=13 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware14 SRVID=14 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware15 SRVID=15 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"

```

```

> tpccfmlw SRVGRP=group1
RQADDR=ware16 SRVID=16 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware17 SRVID=17 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
> tpccfmlw SRVGRP=group1
RQADDR=ware18 SRVID=18 REPLYQ=N
CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"

=====
(diff between cl061 and cl062 configuration)
=====
980c980
< Term_Base="103401"
...
> Term_Base="321801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
1048c1048
< cl061 LMID=SITE1
...
> cl062 LMID=SITE1

=====
(diff between cl062 and cl063 configuration)
=====
980c980
< Term_Base="321801"
...
> Term_Base="540201"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
1048c1048
< cl062 LMID=SITE1
...
> cl063 LMID=SITE1

=====
(diff between cl063 and cl064 configuration)
=====
980c980
< Term_Base="540201"
...
> Term_Base="758601"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
1048c1048
< cl063 LMID=SITE1
...
> cl064 LMID=SITE1

=====
(diff between cl064 and cl065 configuration)
=====
980c980
< Term_Base="758601"
...
> Term_Base="954001"
1014c1014

```

```

< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
1048c1048
< cl064 LMID=SITE1
...
> cl065 LMID=SITE1

=====
(diff between cl065 and cl066 configuration)
=====
980c980
< Term_Base="954001"
...
> Term_Base="1172401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
1048c1048
< cl065 LMID=SITE1
...
> cl066 LMID=SITE1

=====
(diff between cl066 and cl067 configuration)
=====
980c980
< Term_Base="1172401"
...
> Term_Base="1390801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
1048c1048
< cl066 LMID=SITE1
...
> cl067 LMID=SITE1

=====
(diff between cl067 and cl069 configuration)
=====
19a20,21
> # - the wildcard %, can be also used
with %group syntax,
> # for maxlogin limit
36a39
> # - maxsyslogins - max number of logins
on the system
38a42,43
> # - sigpending - max number of pending
signals
> # - msgqueue - max memory used by
POSIX message queues (bytes)
980c985
< Term_Base="1390801"
...
> Term_Base="1609201"
1014c1019
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
1048c1053
< cl067 LMID=SITE1
...
> cl069 LMID=SITE1

```

```

=====
(diff between cl069 and cl105 configuration)
=====
20,21d19
< # - the wildcard %, can be also used
with %group syntax,
< # for maxlogin limit
39d36
< # - maxsyslogins - max number of logins
on the system
42,43d38
< # - sigpending - max number of pending
signals
< # - msgqueue - max memory used by
POSIX message queues (bytes)
985c980
< Term_Base="1609201"
...
> Term_Base="977001"
1019c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
1053c1048
< cl069 LMID=SITE1
...
> cl105 LMID=SITE1

=====
(diff between cl105 and cl106 configuration)
=====
980c980
< Term_Base="977001"
...
> Term_Base="1195401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
1048c1048
< cl105 LMID=SITE1
...
> cl106 LMID=SITE1

=====
(diff between cl106 and cl107 configuration)
=====
980c980
< Term_Base="1195401"
...
> Term_Base="1413801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
1048c1048
< cl106 LMID=SITE1
...
> cl107 LMID=SITE1

=====
(diff between cl107 and cl108 configuration)
=====
980c980
< Term_Base="1413801"
...
> Term_Base="1632201"
1014c1014

```

```

< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
1048c1048
< cl107 LMID=SITE1
...
> cl108 LMID=SITE1

=====
(diff between cl108 and cl109 configuration)
=====
980c980
< Term_Base="1632201"
...
> Term_Base="126401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
1048c1048
< cl108 LMID=SITE1
...
> cl109 LMID=SITE1

=====
(diff between cl109 and cl110 configuration)
=====
980c980
< Term_Base="126401"
...
> Term_Base="344801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
1048c1048
< cl109 LMID=SITE1
...
> cl110 LMID=SITE1

=====
(diff between cl110 and cl111 configuration)
=====
980c980
< Term_Base="344801"
...
> Term_Base="563201"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
1048c1048
< cl110 LMID=SITE1
...
> cl111 LMID=SITE1

=====
(diff between cl111 and cl112 configuration)
=====
980c980
< Term_Base="563201"
...
> Term_Base="781601"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
...

```

```

> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
1048c1048
< cl111 LMID=SITE1
...
> cl112 LMID=SITE1

=====
(diff between cl112 and cl113 configuration)
=====
980c980
< Term_Base="781601"
...
> Term_Base="1000001"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
1048c1048
< cl112 LMID=SITE1
...
> cl113 LMID=SITE1

=====
(diff between cl113 and cl114 configuration)
=====
980c980
< Term_Base="1000001"
...
> Term_Base="1218401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
1048c1048
< cl113 LMID=SITE1
...
> cl114 LMID=SITE1

=====
(diff between cl114 and cl115 configuration)
=====
980c980
< Term_Base="1218401"
...
> Term_Base="1436801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
1048c1048
< cl114 LMID=SITE1
...
> cl115 LMID=SITE1

=====
(diff between cl115 and cl116 configuration)
=====
980c980
< Term_Base="1436801"
...
> Term_Base="1655201"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
1048c1048

```

```

< cl115 LMID=SITE1
...
> cl116 LMID=SITE1

=====
(diff between cl116 and cl117 configuration)
=====
980c980
< Term_Base="1655201"
...
> Term_Base="149401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
1048c1048
< cl116 LMID=SITE1
...
> cl117 LMID=SITE1

=====
(diff between cl117 and cl118 configuration)
=====
980c980
< Term_Base="149401"
...
> Term_Base="367801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
1048c1048
< cl117 LMID=SITE1
...
> cl118 LMID=SITE1

=====
(diff between cl118 and cl119 configuration)
=====
980c980
< Term_Base="367801"
...
> Term_Base="586201"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
1048c1048
< cl118 LMID=SITE1
...
> cl119 LMID=SITE1

=====
(diff between cl119 and cl120 configuration)
=====
980c980
< Term_Base="586201"
...
> Term_Base="804601"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
1048c1048
< cl119 LMID=SITE1
...
> cl120 LMID=SITE1

```

```

=====
(diff between cl120 and cl121 configuration)
=====
980c980
< Term_Base="804601"
...
> Term_Base="1023001"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
1048c1048
< cl120 LMID=SITE1
...
> cl121 LMID=SITE1

=====
(diff between cl121 and cl122 configuration)
=====
980c980
< Term_Base="1023001"
...
> Term_Base="1241401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
1048c1048
< cl121 LMID=SITE1
...
> cl122 LMID=SITE1

=====
(diff between cl122 and cl123 configuration)
=====
980c980
< Term_Base="1241401"
...
> Term_Base="1459801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
1048c1048
< cl122 LMID=SITE1
...
> cl123 LMID=SITE1

=====
(diff between cl123 and cl124 configuration)
=====
980c980
< Term_Base="1459801"
...
> Term_Base="1678201"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
1048c1048
< cl123 LMID=SITE1
...
> cl124 LMID=SITE1

=====
(diff between cl124 and cl125 configuration)

```

```

=====
980c980
< Term_Base="1678201"
...
> Term_Base="172401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
1048c1048
< cl124 LMID=SITE1
...
> cl125 LMID=SITE1

=====
(diff between cl125 and cl126 configuration)
=====
980c980
< Term_Base="172401"
...
> Term_Base="390801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
1048c1048
< cl125 LMID=SITE1
...
> cl126 LMID=SITE1

=====
(diff between cl126 and cl127 configuration)
=====
980c980
< Term_Base="390801"
...
> Term_Base="609201"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
1048c1048
< cl126 LMID=SITE1
...
> cl127 LMID=SITE1

=====
(diff between cl127 and cl128 configuration)
=====
980c980
< Term_Base="609201"
...
> Term_Base="827601"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
...
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
1048c1048
< cl127 LMID=SITE1
...
> cl128 LMID=SITE1

=====
(diff between cl128 and cl129 configuration)
=====
980c980
< Term_Base="827601"

```

```

---
> Term_Base="1046001"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
1048c1048
< cl128 LMID=SITE1
---
> cl129 LMID=SITE1

=====
(diff between cl129 and cl130 configuration)
=====
980c980
< Term_Base="1046001"
---
> Term_Base="1264401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
1048c1048
< cl129 LMID=SITE1
---
> cl130 LMID=SITE1

=====
(diff between cl130 and cl131 configuration)
=====
980c980
< Term_Base="1264401"
---
> Term_Base="1482801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
1048c1048
< cl130 LMID=SITE1
---
> cl131 LMID=SITE1

=====
(diff between cl131 and cl132 configuration)
=====
980c980
< Term_Base="1482801"
---
> Term_Base="1701201"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
1048c1048
< cl131 LMID=SITE1
---
> cl132 LMID=SITE1

=====
(diff between cl132 and cl133 configuration)
=====
980c980
< Term_Base="1701201"
---
> Term_Base="195401"
1014c1014

```

```

< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
1048c1048
< cl132 LMID=SITE1
---
> cl133 LMID=SITE1

=====
(diff between cl133 and cl134 configuration)
=====
980c980
< Term_Base="195401"
---
> Term_Base="413801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
1048c1048
< cl133 LMID=SITE1
---
> cl134 LMID=SITE1

=====
(diff between cl134 and cl135 configuration)
=====
980c980
< Term_Base="413801"
---
> Term_Base="632201"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
1048c1048
< cl134 LMID=SITE1
---
> cl135 LMID=SITE1

=====
(diff between cl135 and cl136 configuration)
=====
980c980
< Term_Base="632201"
---
> Term_Base="850601"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
1048c1048
< cl135 LMID=SITE1
---
> cl136 LMID=SITE1

=====
(diff between cl136 and cl137 configuration)
=====
980c980
< Term_Base="850601"
---
> Term_Base="1069001"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
---

```

```

> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
1048c1048
< cl136 LMID=SITE1
---
> cl137 LMID=SITE1

=====
(diff between cl137 and cl138 configuration)
=====
980c980
< Term_Base="1069001"
---
> Term_Base="1287401"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
1048c1048
< cl137 LMID=SITE1
---
> cl138 LMID=SITE1

=====
(diff between cl138 and cl139 configuration)
=====
980c980
< Term_Base="1287401"
---
> Term_Base="1505801"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
1048c1048
< cl138 LMID=SITE1
---
> cl139 LMID=SITE1

=====
(diff between cl139 and cl140 configuration)
=====
980c980
< Term_Base="1505801"
---
> Term_Base="1724201"
1014c1014
< (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
---
> (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
1048c1048
< cl139 LMID=SITE1
---
> cl140 LMID=SITE1

```

Appendix E: Database Creation Code

```

.....:
createdb.sql
.....:

spool createdb.log

set echo on

shutdown abort

startup pfile=p_create.ora nomount
create database tpcc
  controlfile reuse
  maxinstances 1
  datafile
  'ora_dev/system_1' size 400M reuse
  logfile 'ora_dev/log_1_1' size 340576M reuse
  'ora_dev/log_1_2' size 340576M reuse
  sysaux datafile 'ora_dev/tpccaux' size 120M
  reuse ;

create undo tablespace undo_1 datafile
  'ora_dev/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

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 ;
!

.....:
Makefile.linux
.....:

#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

#include
$(ORACLE_HOME)/bench/buildtools/prefix.mk

#_SYM=-I
#include
$(ORACLE_HOME)/rdbs/lib/env_rdbms.mk

#LINK=/opt/SunProd/SUNW/spro6.1/bin/..WS6U
1/bin/cc
#CC=/opt/SunProd/SUNW/spro6.1/bin/..WS6U1/
bin/cc
LINK=/usr/bin/gcc
CC=/usr/bin/gcc

#LDFLAGS=-L$(ORACLE_HOME)/rdbs/lib/ -
L$(ORACLE_HOME)/lib/ -dy \
#-L$(ORACLE_HOME)/rdbs/lib/ -
L$(ORACLE_HOME)/lib/ \
#$(ORACLE_HOME)/rdbs/lib/defopt.o -lcintsh \
# cat $(ORACLE_HOME)/lib/sysliblist' \
#-o $@

LDFLAGS=-L$(ORACLE_HOME)/rdbs/lib/ -
L$(ORACLE_HOME)/lib/ -dy \
-L$(ORACLE_HOME)/rdbs/lib/ -
L$(ORACLE_HOME)/lib/ \
-lcintsh \
`cat $(ORACLE_HOME)/lib/sysliblist' \
-L$(ORACLE_HOME)/lib -o $@

```

```

#-R$(ORACLE_HOME)/lib -laio -lthread -lposix4
-lkstat -lm -o $@

I_SYM=-I

TARGS=compile cleanup
REMOVE=rm

DPB_LIB_DIR=./lib
DPB_LIB=$(DPB_LIB_DIR)/dpblibunix.o

TPCBIN=./bin
INCLUDE=$(I_SYM).
$(I_SYM)$(ORACLE_HOME)/rdbs/demo \
$(I_SYM)$(ORACLE_HOME)/rdbs/public \
$(I_SYM)$(ORACLE_HOME)/rdbs/include \
$(I_SYM)$(ORACLE_HOME)/plssql/public \
$(I_SYM)$(ORACLE_HOME)/network/public \
$(I_SYM)$(DPB_LIB_DIR)
ITUX=$(I_SYM)$(ROOTDIR)/include

MEMBS=
OBS=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_OBS=plnew.o plpay.o plord.o pldel.o
plsto.o
CTRANPOCI_OBS=plnew.o plpay_oci.o
plord.o pldel.o plsto.o
CTRANTUX_OBS=plnew_tux.o plpay.o plord.o
pldel_tux.o plsto.o
OTHER_OBS=c_drv_val.o test_drv.o
test_sample.o test_tran.o single_txn_ran.o
TUX_OBS=c_drv_tux.o tpccpl_tux.o tpccsvr.o

files:

compile: $(OBS) $(DPB_LIB)
@-$(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) $(OBS) $(CTRAN_OBS)
$(CTRANTUX_OBS) $(OTHER_OBS) \
$(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 \
$(TUX_OBS)
@-$(DOTARGS)

$(DPB_LIB):
( cd $(DPB_LIB_DIR); $(MAKE) -f
Makefile.linux )

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
$(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

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 errprt.o $(DPB_LIB)
$(LINK) $(LDFLAGS) \
c_drv_o7.o c_trans.o tpccpl.o c_dump.o
errprt.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

.....
createindex_icust1.sql
.....

```

```

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 intrans 3
storage ( buffer_pool default )
parallel 512
compute statistics
tablespace icust1_0 ;
set echo off
spool off
exit sql.sqlcode;

.....
createindex_icust2.sql
.....

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 intrans 3
storage ( buffer_pool default )
parallel 512
compute statistics
tablespace icust2_0 ;
set echo off
spool off
exit sql.sqlcode;

.....
createindex_idist.sql
.....

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 intrans 3
storage ( buffer_pool default )
parallel 1
compute statistics
tablespace idist_0 ;
set echo off
spool off
exit sql.sqlcode;

.....
createindex_iitem.sql
.....

set timing on
set sqlblanklines on
spool createindex_iitem.log ;
set echo on ;
drop index iitem ;
create unique index iitem on item ( i_id )
pctfree 5 intrans 4

```



```

storage ( buffer_pool default )

compute statistics
tablespace ittem_0 ;
set echo off
spool off
exit sql.sqlcode;

.....
createindex_inord.sql
.....

set timing on
exit 0;

.....
createindex_iordl.sql
.....

set timing on
exit 0;

.....
createindex_iordr1.sql
.....

set timing on
exit 0;

.....
createindex_iordr2.sql
.....

set timing on
set sqlblanklines on
spool createindex_iordr2.log ;
set echo on ;
drop index iordr2 ;
create unique index iordr2 on ordr (
    o_w_id
    , o_c_id
    , o_d_id
    , o_id )
global partition by range (o_w_id) (
partition iordr2_1 values less than (21840) ,
partition iordr2_2 values less than (43680) ,
partition iordr2_3 values less than (65520) ,
partition iordr2_4 values less than (87360) ,
partition iordr2_5 values less than (109200) ,
partition iordr2_6 values less than (131040) ,
partition iordr2_7 values less than (152880) ,
partition iordr2_8 values less than (MAXVALUE)
)
pctfree 25 initrans 4
storage ( buffer_pool default )
parallel 256
compute statistics
tablespace iordr2_0 ;
alter index iordr2 noprogram;
set echo off
spool off
exit sql.sqlcode;

.....
createindex_istok.sql
.....

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 512
compute statistics
tablespace istok_0 ;
set echo off
spool off
exit sql.sqlcode;

.....
createindex_iware.sql
.....

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 1
compute statistics
tablespace iware_0 ;
set echo off
spool off
exit sql.sqlcode;

.....
createtable_dist.sql
.....

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 1747200
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 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)
)
cluster distcluster (
    d_id
    , d_w_id
);
set echo off
spool off

exit sql.sqlcode;

.....
createtabled_cust.sql
.....

/* created automatically by
/home/nelsond/Desktop/partitioning/or_part_stok
.new/tpcc174720/scripts/buildcreatetable.sh
Wed Jun 20 14:46:58 PDT 2007 */
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 5241600000
hash is ( (c_w_id * 30000 + c_id * 10 + c_d_id -
30011) )
size 360
pctfree 0 initrans 3
storage ( initial 1365002k next 1365000k
maxextents unlimited pctincrease 0 freelist
groups 4 buffer_pool recycle ) parallel ( degree
128 )
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 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 char(500)
)
cluster custcluster (
    c_id
    , c_d_id
    , c_w_id
);
set echo off
spool off
exit sql.sqlcode;

```

```

.....
createtable_hist.sql
.....

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 number
, h_data varchar2(24)
)
pctfree 5 initrans 4
storage ( buffer_pool recycle )
tablespace hist_0
partition by range (h_c_w_id)
(
  partition wh1 values less than (21840) ,
  partition wh2 values less than (43680) ,
  partition wh3 values less than (65520) ,
  partition wh4 values less than (87360) ,
  partition wh5 values less than (109200) ,
  partition wh6 values less than (131040) ,
  partition wh7 values less than (152880) ,
  partition wh8 values less than (MAXVALUE)
);
alter table hist disable table lock;
set echo off
spool off
exit sql.sqlcode;

```

```

.....
createtable_item.sql
.....

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 number
, i_data varchar2(50)
, i_im_id number
)
cluster itemcluster (
  i_id
);
set echo off

```

```

spool off
exit sql.sqlcode;

.....
createtable_nord.sql
.....

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 1747200
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
.....

set timing on
  set sqlblanklines on
  spool createtable_ordr.log
  set echo on
  create table ordr (
    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)
, constraint ordr_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_ordr.sql
.....

```

```

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 1747200
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_stok.sql
.....

/* created automatically by
/home/nelsond/Desktop/partitioning/or_part_stok
.new/tpcc174720/scripts/buildcreatetable.sh Fri
Jun 15 14:40:32 PDT 2007 */
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 17472000000
hash is ((abs(s_i_id - 1) * 21840 +
mod((s_w_id - 1), 21840) + trunc ((s_w_id - 1) /
21840) * 21840 * 100000))
size 256
pctfree 0 initrans 2 maxtrans 2
storage ( initial 1625002k next 1625000k
maxextents unlimited pctincrease 0 freelist
groups 4 buffer_pool keep ) parallel ( degree
32 )
tablespace stok_0;

create table stok (

```

```

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)
)
cluster stokcluster (
  s_i_id
, s_w_id
);
  set echo off
  spool off
  exit sql.sqlcode;

.....
createtable_ware.sql
.....

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 174720
hash is ( (w_id - 1) )
size 1448
initrans 2
storage ( buffer_pool default )
tablespace ware_0;

create table ware (
  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)
)
cluster warecluster (
  w_id
);
  set echo off
  spool off
  exit sql.sqlcode;

.....
assigntemp.sql
.....

spool assigntemp.log;

set echo on;

```

```

alter user tpcc temporary tablespace temp_0;

set echo off;
spool off;

exit ;

.....
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_0 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
perform
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

!

.....
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;

.....
createts_stok.sh
.....

#!/bin/sh

cd /home/oracle/tpcc-kit
. ./stepenv.sh

/home/oracle/tpcc-kit/scripts/addts.sh stok_0
/ora_dev/stok_0_0 6380M 1625000K auto 0 d
wait
/home/oracle/tpcc-kit/scripts/addfile.sh stok_0
/ora_dev/stok_0_1 6380M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh stok_0
/ora_dev/stok_0_2 6380M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh stok_0
/ora_dev/stok_0_3 6380M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh stok_0
/ora_dev/stok_0_4 6380M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh stok_0
/ora_dev/stok_0_5 6380M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh stok_0
/ora_dev/stok_0_6 6380M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh stok_0
/ora_dev/stok_0_7 6380M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh stok_0
/ora_dev/stok_0_8 6380M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh stok_0
/ora_dev/stok_0_9 6380M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh stok_0
/ora_dev/stok_0_10 6380M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh stok_0
/ora_dev/stok_0_11 6380M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh stok_0
/ora_dev/stok_0_12 6380M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh stok_0
/ora_dev/stok_0_13 6380M 0 &

```



```

/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_902 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_903 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_904 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_905 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_906 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_907 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_908 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_909 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_910 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_911 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_912 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_913 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_916 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_917 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_918 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_919 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_920 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_921 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_922 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_923 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_924 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_925 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_926 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_927 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_928 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_929 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_930 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_931 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_932 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_933 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_934 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_935 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_936 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_937 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_938 5955M 0 &

```

```

/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_939 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_940 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_941 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_942 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_943 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_944 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_945 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_946 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_947 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_948 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_949 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_950 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_951 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_952 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_953 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_954 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_955 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_956 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_957 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_958 5955M 0 &
/home/oracle/tpcc-kit/scripts/addfile.sh cust_0
/ora_dev/cust_0_959 5955M 0 &
wait

.....:
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
# drop tablespace $1 including contents;

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
$createsql
set echo off
spool off
exit ;
!

.....:
analyze.sh
.....:

#!/bin/sh
$tpcc_sqlplus $tpcc_user_pass
@$(tpcc_sql_dir)/analyze > $tpcc_log_dir/junk
2>&1

if test $? -ne 0
then
exit 1;
else
exit 0;
fi

.....:
assigntemp.sh
.....:

#!/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

.....:
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
publbd manually
REM catalog and catproc suppose to take care
of it
REM

connect system/manager
REM @$ORACLE_HOME/sqlplus/admin/publbd

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

.....
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

.....
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=>'ORDL', -
  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
ME=>'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
ME=>'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
ME=>'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
ME=>'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;

.....
loadcust.sh
.....

rm -f loadcust*.log
cd $tpcc_bench
$tpcc_load -M 174720 -c -l1 -m 85 >>
loadcust0.log 2>&1 &
$tpcc_load -M 174720 -c -l86 -m 170 >>
loadcust1.log 2>&1 &
$tpcc_load -M 174720 -c -l171 -m 255 >>
loadcust2.log 2>&1 &
$tpcc_load -M 174720 -c -l256 -m 340 >>
loadcust3.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l341 -m 425 >>
loadcust4.log 2>&1 &

```



```
$tpcc_load -M 174720 -c -l 169303 -m 169388
>> loadcust1985.log 2>&1 &
$tpcc_load -M 174720 -c -l 169389 -m 169474
>> loadcust1986.log 2>&1 &
$tpcc_load -M 174720 -c -l 169475 -m 169560
>> loadcust1987.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 169561 -m 169646
>> loadcust1988.log 2>&1 &
$tpcc_load -M 174720 -c -l 169647 -m 169732
>> loadcust1989.log 2>&1 &
$tpcc_load -M 174720 -c -l 169733 -m 169818
>> loadcust1990.log 2>&1 &
$tpcc_load -M 174720 -c -l 169819 -m 169904
>> loadcust1991.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 169905 -m 169990
>> loadcust1992.log 2>&1 &
$tpcc_load -M 174720 -c -l 169991 -m 170076
>> loadcust1993.log 2>&1 &
$tpcc_load -M 174720 -c -l 170077 -m 170162
>> loadcust1994.log 2>&1 &
$tpcc_load -M 174720 -c -l 170163 -m 170248
>> loadcust1995.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 170249 -m 170334
>> loadcust1996.log 2>&1 &
$tpcc_load -M 174720 -c -l 170335 -m 170420
>> loadcust1997.log 2>&1 &
$tpcc_load -M 174720 -c -l 170421 -m 170506
>> loadcust1998.log 2>&1 &
$tpcc_load -M 174720 -c -l 170507 -m 170592
>> loadcust1999.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 170593 -m 170678
>> loadcust2000.log 2>&1 &
$tpcc_load -M 174720 -c -l 170679 -m 170764
>> loadcust2001.log 2>&1 &
$tpcc_load -M 174720 -c -l 170765 -m 170850
>> loadcust2002.log 2>&1 &
$tpcc_load -M 174720 -c -l 170851 -m 170936
>> loadcust2003.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 170937 -m 171022
>> loadcust2004.log 2>&1 &
$tpcc_load -M 174720 -c -l 171023 -m 171108
>> loadcust2005.log 2>&1 &
$tpcc_load -M 174720 -c -l 171109 -m 171194
>> loadcust2006.log 2>&1 &
$tpcc_load -M 174720 -c -l 171195 -m 171280
>> loadcust2007.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 171281 -m 171366
>> loadcust2008.log 2>&1 &
$tpcc_load -M 174720 -c -l 171367 -m 171452
>> loadcust2009.log 2>&1 &
$tpcc_load -M 174720 -c -l 171453 -m 171538
>> loadcust2010.log 2>&1 &
$tpcc_load -M 174720 -c -l 171539 -m 171624
>> loadcust2011.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 171625 -m 171710
>> loadcust2012.log 2>&1 &
$tpcc_load -M 174720 -c -l 171711 -m 171796
>> loadcust2013.log 2>&1 &
$tpcc_load -M 174720 -c -l 171797 -m 171882
>> loadcust2014.log 2>&1 &
$tpcc_load -M 174720 -c -l 171883 -m 171968
>> loadcust2015.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 171969 -m 172054
>> loadcust2016.log 2>&1 &
$tpcc_load -M 174720 -c -l 172055 -m 172140
>> loadcust2017.log 2>&1 &
```

```
$tpcc_load -M 174720 -c -l 172141 -m 172226
>> loadcust2018.log 2>&1 &
$tpcc_load -M 174720 -c -l 172227 -m 172312
>> loadcust2019.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 172313 -m 172398
>> loadcust2020.log 2>&1 &
$tpcc_load -M 174720 -c -l 172399 -m 172484
>> loadcust2021.log 2>&1 &
$tpcc_load -M 174720 -c -l 172485 -m 172570
>> loadcust2022.log 2>&1 &
$tpcc_load -M 174720 -c -l 172571 -m 172656
>> loadcust2023.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 172657 -m 172742
>> loadcust2024.log 2>&1 &
$tpcc_load -M 174720 -c -l 172743 -m 172828
>> loadcust2025.log 2>&1 &
$tpcc_load -M 174720 -c -l 172829 -m 172914
>> loadcust2026.log 2>&1 &
$tpcc_load -M 174720 -c -l 172915 -m 173000
>> loadcust2027.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 173001 -m 173086
>> loadcust2028.log 2>&1 &
$tpcc_load -M 174720 -c -l 173087 -m 173172
>> loadcust2029.log 2>&1 &
$tpcc_load -M 174720 -c -l 173173 -m 173258
>> loadcust2030.log 2>&1 &
$tpcc_load -M 174720 -c -l 173259 -m 173344
>> loadcust2031.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 173345 -m 173430
>> loadcust2032.log 2>&1 &
$tpcc_load -M 174720 -c -l 173431 -m 173516
>> loadcust2033.log 2>&1 &
$tpcc_load -M 174720 -c -l 173517 -m 173602
>> loadcust2034.log 2>&1 &
$tpcc_load -M 174720 -c -l 173603 -m 173688
>> loadcust2035.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 173689 -m 173774
>> loadcust2036.log 2>&1 &
$tpcc_load -M 174720 -c -l 173775 -m 173860
>> loadcust2037.log 2>&1 &
$tpcc_load -M 174720 -c -l 173861 -m 173946
>> loadcust2038.log 2>&1 &
$tpcc_load -M 174720 -c -l 173947 -m 174032
>> loadcust2039.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 174033 -m 174118
>> loadcust2040.log 2>&1 &
$tpcc_load -M 174720 -c -l 174119 -m 174204
>> loadcust2041.log 2>&1 &
$tpcc_load -M 174720 -c -l 174205 -m 174290
>> loadcust2042.log 2>&1 &
$tpcc_load -M 174720 -c -l 174291 -m 174376
>> loadcust2043.log 2>&1 &
wait
$tpcc_load -M 174720 -c -l 174377 -m 174462
>> loadcust2044.log 2>&1 &
$tpcc_load -M 174720 -c -l 174463 -m 174548
>> loadcust2045.log 2>&1 &
$tpcc_load -M 174720 -c -l 174549 -m 174634
>> loadcust2046.log 2>&1 &
$tpcc_load -M 174720 -c -l 174635 -m 174720
>> loadcust2047.log 2>&1 &
wait
.....
p_create.ora
.....
compatible = 10.1.0.0.0
```

```
db_name = tpcc
control_files = (/ora_dev/control_001,
/ora_dev/control_002)
db_block_size = 2048
db_cache_size = 85333M
db_8k_cache_size = 32000M
log_buffer = 1048576
db_16k_cache_size = 85333M
undo_management = manual
statistics_level = basic
shared_pool_size = 16000M
plsql_optimize_level=2
db_4k_cache_size = 20M

.....
p_build.ora
.....

compatible = 10.1.0.0.0
db_name = tpcc
control_files =
(/ora_dev/control_001,ora_dev/control_002)
parallel_max_servers = 100
recovery_parallelism = 40
db_files = 2621
db_cache_size = 50400M
db_8k_cache_size = 1890M
db_16k_cache_size = 5040M
db_recycle_cache_size = 7700M
dml_locks = 500
statistics_level = basic
log_buffer = 1048576
processes = 3200
sessions = 3200
transactions = 3200
shared_pool_size = 5000M
cursor_space_for_time = TRUE
db_block_size = 2048
undo_management = auto
undo_retention = 2
plsql_optimize_level=2

UNDO_TABLESPACE = undo_1
db_4k_cache_size = 2000M

.....
loaddist.sh
.....

cd $tpcc_bench
$tpcc_load -M $tpcc_scale -d > loaddist.log
2>&1

.....
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
| tpc.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 "tpccflags.h"

#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 TPCdumpord ();
extern void TPCdumpdel ();
extern void TPCdumpsto ();
extern void TPCdumpexit ();
extern void userlog(char* ftmp, ...);

/* 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 tkvcoint ();
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 multitrans;
extern int ord_init;

extern void errprt ();
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 execsstatus;
extern int errcode;

extern OCIEEnv *tpcenv;
extern OCIServer *tpcsrv;
extern OCIError *errhp;
extern OCISvcCtx *tpcsvc;
extern OCISession *tpcsur;
extern OCISlml *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;
#ifdef USE_IEEE_NUMBER
extern float threshold;
#else
extern int threshold;
#endif /* USE_IEEE_NUMBER */
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_las[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];
#ifdef USE_IEEE_NUMBER
extern float ol_quantity[15];
extern float ol_amount[15];
#else
extern int ol_quantity[15];
extern int ol_amount[15];
#endif /* USE_IEEE_NUMBER */
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;
#ifdef USE_IEEE_NUMBER
extern float h_amount;
#else
extern int h_amount;
#endif /* USE_IEEE_NUMBER */
extern char w_street_1[21];
extern char w_street_2[21];
extern char w_city[21];
extern char w_state[3];
extern char w_zip[10];
extern char d_street_1[21];
extern char d_street_2[21];
extern char d_city[21];
extern char d_state[3];
extern char d_zip[10];
extern char c_street_1[21];
extern char c_street_2[21];
extern char c_city[21];
extern char c_state[3];
extern char c_zip[10];
extern char c_phone[17];
extern text c_since_d[11];
extern char c_credit[3];
extern int c_credit_lim;
extern float c_discount;
extern char c_data[201];
extern text h_date[20];

/* for new order transaction */

extern int no_l_i_id[15];
extern int no_l_supply_w_id[15];
#ifdef USE_IEEE_NUMBER
extern float no_l_quantity[15];
extern float no_l_amount[15];
extern float s_quantity[15];
extern float l_price[15];
#else
extern int no_l_quantity[15];
extern int no_l_amount[15];
extern int s_quantity[15];
extern int l_price[15];
#endif /* USE_IEEE_NUMBER */
extern int no_l_quantity10[15];
extern int no_l_quantity91[15];
extern int no_l_ytdqty[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 char brand_gen[15];
extern ub2 brand_gen_len[15];
extern ub2 brand_gen_rcode[15];
extern ub4 brand_gen_csize;
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 o_d_base[15];

#ifdef DISCARD
# define DISCARD (void)
#endif

#ifdef 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 */

#ifdef 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(err,function)\
ocierror(__FILE__,__LINE__,(err),(function));

#define OCIBND(stmp, bndp, errp, sqlvar, 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)),\
(progvl), (progvl), (ftype), (indp), (alen), (rcode), 0, 0,\
(ftype), 0, 0, 0, 0, OCI_DEFAULT));

```

```

/* bind arrays for sql */
#define
OCIBNDRA(stmp,bndp,errp,sqlvar,progvl,progvl,ftype,indp,alen,rcode)\
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)),\
(progvl),(progvl),(ftype),(indp),(alen),(rcode),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,progvl,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*) (progvl), (progvl), (ftype),\
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,progvl,progvl,ftype,indp,alen,rcode)\
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)),\
(progvl),(progvl),(ftype),(indp),(alen),(rcode),0,0,\
OCI_DEFAULT));

/* bind in/out for plsql arrays without indicator and rcode */
#define
OCIBNDPLA(stmp,bndp,errp,sqlvar,progvl,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 *) (progvl),\
(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,progvl,progvl,ftype,indp,alen,rcode,\
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)),\
(progvl),(progvl),(ftype),(indp),(alen),(rcode),(ms),(cu),OCI_DEFAULT));

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

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

#define
OCIDEF(stmp,dfnp,errp,pos,progvl,progvl,ftype)\

OCIHandleAlloc((stmp),(dvoid*)&(dfnp),OCI_HTYPE_DEFINE,0,\
(dvoid**)0);\

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

```

```

#define
OCIDFNRA(stmp,dfnp,errp,pos,progv,proglv,ftype
e,indp,alen,arcode) \

OCIHandleAlloc((stmp),(dvoid*)&(dfnp),OCI_HT
YPE_DEFINE,0,\
(dvoid**)0);\

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(pro
gv),\
(proglv),(ftype),(indp),(alen),\
(arcode),OCI_DEFAULT);

#define
OCIDFNNDYN(stmp,dfnp,errp,pos,progv,proglv,ft
ype,indp,ctxp,cbf_data) \
ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid*)&(dfnp),OCI_HT
YPE_DEFINE,0,\
(dvoid**)0);\
ocierror(__FILE__,__LINE__,(errp), \

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(pro
gv), (proglv),(ftype),\
(indp),NULL,NULL,
OCI_DYNAMIC_FETCH));\
ocierror(__FILE__,__LINE__,(errp), \

OCIDefineDynamic((dfnp),(errp),(ctxp),(cbf_data
)));

/* New order */

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

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

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

/* Payment */

struct payinstruct {

```

```

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

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

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

/* Order status */

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

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

```

```

};

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

/* Delivery */

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

struct deloutstruct {
int terror;
int retry;
};

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

/* Stock level */

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

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

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

#endif

.....
loadhist.sh
.....

rm -f loadhist*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 174720 -h -b 1 -e 682 >>
loadhist0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 683 -e 1364 >>
loadhist1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 1365 -e 2046 >>
loadhist2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 2047 -e 2728 >>
loadhist3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 2729 -e 3410 >>
loadhist4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 3411 -e 4092 >>
loadhist5.log 2>&1 &

```



```

allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 156963 -e 157645
>> loadhist230.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 157646 -e 158328
>> loadhist231.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 158329 -e 159011
>> loadhist232.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 159012 -e 159694
>> loadhist233.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 159695 -e 160377
>> loadhist234.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 160378 -e 161060
>> loadhist235.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 161061 -e 161743
>> loadhist236.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 161744 -e 162426
>> loadhist237.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 162427 -e 163109
>> loadhist238.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 163110 -e 163792
>> loadhist239.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 163793 -e 164475
>> loadhist240.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 164476 -e 165158
>> loadhist241.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 165159 -e 165841
>> loadhist242.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 165842 -e 166524
>> loadhist243.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 166525 -e 167207
>> loadhist244.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 167208 -e 167890
>> loadhist245.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 167891 -e 168573
>> loadhist246.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 168574 -e 169256
>> loadhist247.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 169257 -e 169939
>> loadhist248.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 169940 -e 170622
>> loadhist249.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 170623 -e 171305
>> loadhist250.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 171306 -e 171988
>> loadhist251.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 171989 -e 172671
>> loadhist252.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 172672 -e 173354
>> loadhist253.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 173355 -e 174037
>> loadhist254.log 2>&1 &

```

```

allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -h -b 174038 -e 174720
>> loadhist255.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
    wait $curproc
    error=`expr $? + $error`
done
exit `expr $error != 0`

.....:
tpccflags.h
.....:

.....:
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 (cluster around
c_w_id)
|           -C load cust table (cluster
around c_id)
|           -i load item table
|           -n load new-order table
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)
|           -l <cid#> beginning cid number
(with -C)
|           -m <cid#> ending cid number
(with -C)
|           -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 "dpbccore.h"
# define gettime dpbtimef
# define getcpu dpbcpcu
# 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. */
#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 RECOVERERR -10
#define IRRRECERR -20

#define SQLXTW "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_06, \
:s_dist_07, :s_dist_08, :s_dist_09, :s_dist_10,
0, 0, 0, :s_data)"

#define SQLTXTI "INSERT INTO item
(I_ID, I_IM_ID, I_NAME, I_PRICE, I_DATA)
VALUES (:i_id, :i_im_id, :i_name, :i_price, \
:i_data)"

#define SQLTXTO1 "INSERT INTO odr (O_ID,
O_D_ID, O_W_ID, O_C_ID, O_ENTRY_D, O_CAR
RIER_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 odr (O_ID,
O_D_ID, O_W_ID, O_C_ID, O_ENTRY_D, O_CAR
RIER_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 SQLTXTO1 "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 SQLTXTO2 "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 SQLXTXNO "INSERT INTO nord
(no_o_id, no_d_id, no_w_id) VALUES
(:no_o_id, :no_d_id, :no_w_id)"

#define SQLXTXENHA "alter session set
'_enable_hash_overflow'=true"
#define SQLXTXDIHA "alter session set
'_enable_hash_overflow'=false"

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

char num9[10];
char num16[17];
char str2[3];
char str24[15][25];
int randperm3000[3000];

void initperm();
void randstr();
void randdatastr();
void randnum();
void randlastname (char*, int);
int NURand();
void sysdate();

OCIEnv *tpcenv;
OCIServer *tpcsrv;
OCIError *errhp;
OCISvcCtx *tpcsvc;
OCISession *tpcs;

OCISmt *curw;
OCISmt *curd;
OCISmt *curc;
OCISmt *curs;
OCISmt *curh;
OCISmt *curs;
OCISmt *curss;
OCISmt *curi;
OCISmt *curo1;
OCISmt *curo2;
OCISmt *curo1;
OCISmt *curo2;
OCISmt *curo;

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 *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 *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;
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:\tpccload -M
<multiplier> [options]\n");
    fprintf(stderr, "options:\n");
    fprintf(stderr, "\t-A :load all tables\n");
    fprintf(stderr, "\t-w :load ware table\n");
    fprintf(stderr, "\t-d :load dist table\n");
    fprintf(stderr, "\t-c :load cust table (cluster
around c_w_id\n");
    fprintf(stderr, "\t-C :load cust table (cluster
around c_id\n");
    fprintf(stderr, "\t-i :load item table\n");
    fprintf(stderr, "\t-s :load stok table (cluster
around s_w_id\n");
    fprintf(stderr, "\t-S :load stok table (cluster
around s_i_id\n");
    fprintf(stderr, "\t-h :load hist table\n");
    fprintf(stderr, "\t-n :load new-order table\n");
    fprintf(stderr, "\t-o <oline file> :load order and
order-line table\n");
    fprintf(stderr, "\t-b <ware#> :beginning ware
number\n");
    fprintf(stderr, "\t-e <ware#> :tending ware
number\n");
    fprintf(stderr, "\t-j <item#> :beginning item
number (with -S)\n");
    fprintf(stderr, "\t-k <item#> :tending item
number (with -S)\n");
    fprintf(stderr, "\t-l <cid#> :beginning cid
number (with -C)\n");
    fprintf(stderr, "\t-m <cid#> :tending cid
number (with -C)\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_SVCTX);
    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;
    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][25];
char s_dist_02[100][25];
char s_dist_03[100][25];
char s_dist_04[100][25];
char s_dist_05[100][25];
char s_dist_06[100][25];
char s_dist_07[100][25];
char s_dist_08[100][25];
char s_dist_09[100][25];
char s_dist_10[100][25];
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;
double begin_cpu, end_cpu;
double gettime(), getcpu();

extern int getopt();
extern char *optarg;
extern int optind, opterr;
int opt;
#endif

char *argstr="M:AwdcCisShno:b:e;j:k:l:m:g";
int do_A=0;
int do_w=0;
int do_d=0;
int do_i=0;
int do_c=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;
int bcid=1;
int ecid=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 '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 'l': bcid = atoi (*argv++);
            break;
        case 'm': ecid = 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;
    case 'w': do_w = 1;

```

```

        break;
    case 'd': do_d = 1;
        break;
    case 'c': do_c = 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 'l': bcid = atoi (optarg);
        break;
    case 'm': ecid = 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_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_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 (do_c && (do_a || do_c)) {
    fprintf(stderr, "Cluster cust table around
c_w_id or c_id?!\n");
    myusage ();
}

if (eware <= 0)
    ewart = scale;
if (ecid <= 0)
    ecid = CUSTFAC;
if (eitem <= 0)
    eitem = STOCFAC;

if (do_c) {
    if ((bcid < 1) || (bcid > CUSTFAC)) {
        fprintf(stderr, "Invalid beginning cid
number: %d!\n", bcid);
        myusage ();
    }

    if ((ecid < bcid) || (ecid > CUSTFAC)) {
        fprintf(stderr, "Invalid ending cid number:
%d!\n", ecid);
        myusage ();
    }
}

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 ((ewart < bware) || (ewart > scale)) {
    fprintf(stderr, "Invalid ending warehouse
number: %d!\n", ewart);
    myusage ();
}

if (gen && do_o) {
    if ((olfp = fopen (olfname, "w")) == NULL) {
        fprintf(stderr, "Can't open '%s' for writing
order lines!\n", olfname);
        myusage ();
    }
}

}

/-----+
| Prepare to insert into database.        |
+-----+
*/
sysdate (sdate);
if (!gen) {

    /* log on to Oracle */

    OCIInitialize(OCI_DEFAULT|OCI_OBJECT,(dvoi
d *)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,(dvoi
d **) &curw, OCI_HTYPE_STMT, 0, (dvoid**)0));
        OCIERROR(errhp,OCISmtPrepare(curw,
errhp, (text *)SQLXTW,
strlen((char *)SQLXTW), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_a || do_d) {

        OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoi
d **) &curd, OCI_HTYPE_STMT, 0, (dvoid**)0));
        OCIERROR(errhp,OCISmtPrepare(curd,
errhp, (text *)SQLXTD,
strlen((char *)SQLXTD), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_a || do_c || do_C) {

        OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoi
d **) &curc, OCI_HTYPE_STMT, 0, (dvoid**)0));
        OCIERROR(errhp,OCISmtPrepare(curc,
errhp, (text *)SQLXTC,
strlen((char *)SQLXTC), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));

        OCIERROR(errhp,OCISmtPrepare(curc,
errhp, (text *)SQLXTXTCQUERY,
strlen((char *)SQLXTXTCQUERY),
(ub4) OCI_NTV_SYNTAX, (ub4)
OCI_DEFAULT));
    }

    if (do_a || do_h) {

        OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoi
d **) &curh, OCI_HTYPE_STMT, 0, (dvoid**)0));
        OCIERROR(errhp,OCISmtPrepare(curh,
errhp, (text *)SQLXTH,
strlen((char *)SQLXTH), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_a || do_s || do_S) {

        OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoi
d **) &curS, OCI_HTYPE_STMT, 0, (dvoid**)0));
        OCIERROR(errhp,OCISmtPrepare(curS,
errhp, (text *)SQLXTS,
strlen((char *)SQLXTS), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));

        OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoi
d **) &curSS, OCI_HTYPE_STMT, 0, (dvoid**)0));
        OCIERROR(errhp,OCISmtPrepare(curSS,
errhp, (text *)SQLXTSSQUERY,
strlen((char *)SQLXTSSQUERY),
(ub4) OCI_NTV_SYNTAX, (ub4)
OCI_DEFAULT));
    }

    if (do_a || do_i) {

        OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoi
d **) &curi, OCI_HTYPE_STMT, 0, (dvoid**)0));
        OCIERROR(errhp,OCISmtPrepare(curI,
errhp, (text *)SQLXTI,
strlen((char *)SQLXTI), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_a || do_o) {
        int stat;
        char fname[160];

        OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoi
d **) &curO1, OCI_HTYPE_STMT, 0, (dvoid**)0));
        DISCARD strcpy(fname,basename);
        DISCARD strcat(fname, "/");
        DISCARD strcat(fname,
"benchrun/blocks/load_ordordl.sql");
        stat = sqfile(fname, stmbuf);
        if (!stat)
        {
            fprintf(stderr, "unable to open %s
\n",fname);
            quit();
            exit(1);
        }
        OCIERROR(errhp,OCISmtPrepare(curO1,
errhp, stmbuf,

```

```

        strlen((char *)stmbuf), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_A || do_n) {

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoi
d **>(&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,
        (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 || 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));

        OCIERROR(errhp, OCIBindByName(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",
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",
strlen("s_dist_01"), (ub1
*)s_dist_01, 25, SQLT_STR,
(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, 25, SQLT_STR,
(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, 25, SQLT_STR,
(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, 25, SQLT_STR,
(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, 25, SQLT_STR,
(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, 25, SQLT_STR,
(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, 25, SQLT_STR,
        (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, 25, SQLT_STR,
        (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, 25, SQLT_STR,
        (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, 25, SQLT_STR,
        (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_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(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));

        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) 15*ORDEARR, (ub4 *)
&o_carrier_id_clen, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curo1,
&o_l_id_bp, errhp, (text *)":o_l_id",
        strlen(":o_l_id"), (ub1 *)o_l_id,
sizeof(int), SQLT_INT,
        (dvoid *) 0, (ub2 *)o_l_id_len, (ub2
*)0,
        (ub4) 15*ORDEARR, (ub4 *)
&o_l_id_clen, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curo1,
&o_supply_w_id_bp, errhp, (text
*)":o_supply_w_id",
        strlen(":o_supply_w_id"), (ub1
*)o_supply_w_id, sizeof(int), SQLT_INT,
        (dvoid *) 0, (ub2
*)o_supply_w_id_len, (ub2 *)0,
        (ub4) 15*ORDEARR, (ub4 *)
&o_supply_w_id_clen, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curo1,
&o_dist_info_bp, errhp, (text *)":o_dist_info",
        strlen(":o_dist_info"), (ub1
*)o_dist_info, 24, SQLT_CHR,
        (dvoid *) 0, (ub2 *)o_dist_info_len,
(ub2 *)0,
        (ub4) 15*ORDEARR, (ub4 *)
&o_dist_info_clen, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curo1,
&o_amount_bp, errhp, (text *)":o_amount",
        strlen(":o_amount"), (ub1
*)o_amount, sizeof(int), SQLT_INT,
        (dvoid *) 0, (ub2 *)o_amount_len,
(ub2 *)0,
        (ub4) 15*ORDEARR, (ub4 *)
&o_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));

        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) 15*ORDEARR, (ub4 *)
&o_carrier_id_clen, (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_o_cnt_len[i] = sizeof(int);
        }

        OCIERROR(errhp, OCIBindByName(curo1,
&o_o_id_bp, errhp, (text *)":o_o_id",
        strlen(":o_o_id"), (ub1 *)o_o_id,
sizeof(int), SQLT_INT,
        (dvoid *) 0, (ub2 *)o_o_id_len,
(ub2 *)0,
        (ub4) 15*ORDEARR, (ub4
*)&o_o_id_clen, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curo1,
&o_l_d_id_bp, errhp, (text *)":o_l_d_id",
        strlen(":o_l_d_id"), (ub1 *)o_l_d_id,
sizeof(int), SQLT_INT,
        (dvoid *) 0, (ub2 *)o_l_d_id_len,
(ub2 *)0,
        (ub4) 15*ORDEARR, (ub4 *)
&o_l_d_id_clen, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curo1,
&o_l_w_id_bp, errhp, (text *)":o_l_w_id",
        strlen(":o_l_w_id"), (ub1 *)o_l_w_id,
sizeof(int), SQLT_INT,
        (dvoid *) 0, (ub2 *)o_l_w_id_len,
(ub2 *)0,
        (ub4) 15*ORDEARR, (ub4 *)
&o_l_w_id_clen, (ub4) OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curo1,
&o_l_number_bp, errhp, (text *)":o_l_number",
        strlen(":o_l_number"), (ub1
*)o_l_number, sizeof(int), SQLT_INT,
        (dvoid *) 0, (ub2 *)o_l_number_len,
(ub2 *)0,
        (ub4) 15*ORDEARR, (ub4 *)
&o_l_number_clen, (ub4) OCI_DEFAULT));
    }

```

```

        strlen(":o_carrier_id"), (ub1
*)o_carrier_id, sizeof(int), SOLT_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), SOLT_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), SOLT_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), SOLT_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), SOLT_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), SOLT_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), SOLT_INT,
(dvoid *) 0, (ub2 *)0, (ub2 *)0,
(ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
    }
}

/*-----+
| Initialize random number generator |
+-----+
*/

    srand (SEED);
#ifdef 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) ((Irand48 () % 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);
                fflush (stdout);
            }
            else {
                w_id = loop;
                strncpy (w_state, str2, 2);
                strncpy (w_zip, num9, 9);

                status = OCISmtExecute(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) ((Irand48 () % 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 30000000 %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 = OCISmtExecute(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, aware, 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);
    OCIERROR(errhp,OCIStmtExecute(tpcsvc,
curi, errhp,1,0,0,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 = gettimeofday();
begin_cpu = getcpu();

s_c_id = 1;
s_c_d_id = 1;
s_c_w_id = bware;

while (s_c_w_id <= aware) {
    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 > aware) s_c_w_id = aware;
    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) {
    if (s_c_d_id == DISTFAC) {
        s_c_d_id = 1;
        s_c_w_id++;
    } else {
        s_c_d_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 = (aware - 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_lastname, cid - 1);
            else
                randlastname(c_lastname, NURand(255,
0, 999, CNUM1));
            c_credit[i][1] = 'C';
            if (lrand48() % 10)
                c_credit[i][0] = 'G';
            else
                c_credit[i][0] = 'B';
            c_discount[i] = (float)((lrand48() % 5001)
* 0.0001);
            randstr(c_first[i], 8, 16);
            randstr(c_street_1[i], 10, 20);
            randstr(c_street_2[i], 10, 20);
            randstr(c_city[i], 10, 20);
            randstr(str2, 2, 2);
            randnum(num9, 9);
            num9[4] = num9[5] = num9[6] = num9[7]
= num9[8] = '1';
}
}
}

```

```

randnum(num16, 16);
randstr(c_data[i], 300, 500);

if (gen) {
    printf("%d %d %d %s
OE %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s
5000000 %6.4f -1000 1000 1 0 %s\n",
        cid, cdid, cwid, c_first[i], c_lastname[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 = gettimeofday();
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,OCI_DEFAULT));
    OCIHandleFree(curi, OCI_HTYPE_STMT);
}

/*-----
| Load the CUSTOMER table (cluster around
c_id |
*/

```

```

+-----
*/
if (do_C) {
    srand (bcid);
#ifdef ORA_NT
    srand48 (bcid);
#endif

    nrows = (ecid - bcid + 1) * (eware - bware
+1) * DISTFAC;

    fprintf (stderr, "Loading/generating customer:
c%d - c%d, w%d - w%d (%d rows)\n ",
bcid, ecid, 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, SQLTXTHA);
        OCIStmtPrepare(curi, errhp, stmbuf,
strlen(char *)stmbuf,
OCI_NTV_SYNTAX,
OCI_DEFAULT);
        OCIERROR(errhp, OCIStmtExecute(tpcsvc,
curi, errhp, 1, 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 = bcid;
    s_c_d_id = 1;
    s_c_w_id = bware;

    while (s_c_id <= ecid) {
        status = OCIStmtExecute(tpcsvc, curcs,
errhp, (ub4) 1, (ub4) 0,
(CONST OCI_Snapshot*) 0,
(OCI_Snapshot*) 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_id--;
            break;
        }
        else s_c_id++;
    }

    if (s_c_id < bcid) s_c_id = bcid;
    else {
        if (s_c_id > ecid) s_c_id = ecid;
        while (s_c_w_id <= eware) {
            status = OCIStmtExecute(tpcsvc, curcs,
errhp, (ub4) 1, (ub4) 0,
(CONST OCI_Snapshot*) 0,
(OCI_Snapshot*) 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_w_id--;
        break;
    }
    else s_c_w_id++;
}
if (s_c_w_id > eware) s_c_w_id = eware;
else if (s_c_w_id < bware) s_c_w_id =
bware;

while (s_c_d_id <= DISTFAC) {
    status = OCIStmtExecute(tpcsvc, curcs,
errhp, (ub4) 1, (ub4) 0,
(CONST OCI_Snapshot*) 0,
(OCI_Snapshot*) 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_d_id++;
}

if (s_c_d_id > DISTFAC) {
    s_c_d_id = 1;
    if (s_c_w_id == eware) {
        s_c_w_id = bware;
        s_c_id++;
    }
    else s_c_w_id++;
}

fprintf (stderr, "start at cid: %d, wid: %d,
did: %d\n ", s_c_id, s_c_w_id, s_c_d_id);
cid = s_c_id;
cdid = s_c_d_id - 1;
cwid = s_c_w_id;
nrows = (ecid - s_c_id + 1) * (eware - bware
+ 1) * DISTFAC - (s_c_w_id - 1) * DISTFAC -
s_c_d_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++) {
        cdid++;
        if (cdid > DISTFAC) { /* cycle dist id
*/
            cdid = 1; /* cheap mod */
            cwid++; /* shift dist cycle */
            if (cwid > eware) {
                cwid = bware; /* shift ware
cycle */
                cid++;
            }
        }
        c_id[i] = cid;
        c_d_id[i] = cdid;
        c_w_id[i] = cwid;
        if (cid <= 1000)
            randlastname (c_last[i], cid - 1);
        else
            randlastname (c_last[i], NURand (255,
0, 999, CNUM1));

```

```

        c_credit[i][1] = 'C';
        if (lrand48 () % 10)
            c_credit[i][0] = 'G';
        else
            c_credit[i][0] = 'B';
        c_discount[i] = (float)((lrand48 () % 5001)
* 0.0001);
        randstr (c_first[i], 8, 16);
        randstr (c_street_1[i], 10, 20);
        randstr (c_street_2[i], 10, 20);
        randstr (c_city[i], 10, 20);
        randstr (str2, 2, 2);
        randnum (num9, 9);
        num9[4] = num9[5] = num9[6] = num9[7]
= num9[8] = '1';
        randnum (num16, 16);
        randstr (c_data[i], 300, 500);

        if (gen) {
            printf ("%d %d %d %s
OE %s %s %s %s %s %s %s %s %cC
5000000 %6.4f -1000 1000 1 0 %s\n",
cid, cdid, cwid, c_first[i], c_last[i],
c_street_1[i], c_street_2[i], c_city[i],
str2, num9,
num16, sdate, c_credit[i][0],
c_discount[i], c_data[i]);
        }
        else {
            strncpy (c_state[i], str2, 2);
            strncpy (c_zip[i], num9, 9);
            strncpy (c_phone[i], num16, 16);
        }

        if (gen) {
            fflush (stdout);
        }
        else {
            status = OCIStmtExecute(tpcsvc, curc,
errhp, (ub4) i, (ub4) 0,
(CONST OCI_Snapshot*) 0,
(OCI_Snapshot*) 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,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]);
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, aware, nrows);

begin_time = gettime ();
begin_cpu = getcpu ();

s_s_i_id = 1;
s_s_w_id = bware;

while (s_s_w_id <= aware) {
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 > aware) s_s_w_id = aware;
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 = (aware - 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
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 {
/* Changed to STOCKARR to i - alex.ni */
status = OCIStmtExecute(tpcsvc, curss,
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 (0);
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 = OCISmtExecute(tpcsvc, cursr,
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 = OCISmtExecute(tpcsvc, cursr,
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;

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 %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 = OCISmtExecute(tpcsvc, cursr,
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 (0);
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, cid,
cwid, sdate, h_data[i]);
}
}

if (gen) {
fflush (stdout);
}
else {
status = OCISmtExecute(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 ord)\n
",
        bware, aware, 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[j] = cid;
            o_d_id[j] = cdid;
            o_w_id[j] = cwid;
            o_c_id[j] = randperm3000[cid - 1];
            if (cid >= 2101) {
                o_carrier_id[j] = 11;
            }
        }

        for (j = 0; j < o_ol_cnt[j]; 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]);
                    }
                }
            }
            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,
OCISntExecute(tpcscv, cur01, 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, aware, nrows);

    begin_time = gettime ();
    begin_cpu = getcpu ();

    cid = 0;
    cdid = 1;
    cwid = bware;
    loopcount = 0;

    for (row = 0; row < nrows; ) {
        for (i = 0; i < NEWOARR; i++, row++) {
            cid++;

```

```

if (cid > NEWOFAC) {
    cid = 1;
    cdid++;
    if (cdid > DISTFAC) {
        cdid = 1;
        cwid++;
    }
}

if (gen) {
    printf ("%d %d %d\n", cid + 2100, cdid,
cwid);
}
else {
    no_o_id[i] = cid + 2100;
    no_d_id[i] = cdid;
    no_w_id[i] = cwid;
}
}

if (gen) {
    fflush (stdout);
}
else {
    status = OCISmtExecute(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 (0);
        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 (0);
exit (0);
}

void initperm ()
{
    int i;
    int pos;
    int temp;

    /* init randperm3000 */

```

```

for (i = 0; i < 3000; i++)
    randperm3000[i] = i + 1;
for (i = 3000; i > 0; i--) {
    pos = lrand48 () % i;
    temp = randperm3000[i - 1];
    randperm3000[i - 1] = randperm3000[pos];
    randperm3000[pos] = temp;
}
}

void randstr (str, x, y)
char *str;
int x;
int y;
{
    int i, j;
    int len;

    len = (lrand48 () % (y - x + 1)) + x;
    for (i = 0; i < len; i++) {
        j = lrand48 () % 62;
        if (j < 26)
            str[i] = (char) (j + 'a');
        else if (j < 52)
            str[i] = (char) (j - 26 + 'A');
        else
            str[i] = (char) (j - 52 + '0');
    }
    str[len] = '\0';
}

void randdatastr (str, x, y)
char *str;
int x;
int y;
{
    int i, j;
    int len;
    int pos;

    len = (lrand48 () % (y - x + 1)) + x;
    for (i = 0; i < len; i++) {
        j = lrand48 () % 62;
        if (j < 26)
            str[i] = (char) (j + 'a');
        else if (j < 52)
            str[i] = (char) (j - 26 + 'A');
        else
            str[i] = (char) (j - 52 + '0');
    }
    str[len] = '\0';
    if ((lrand48 () % 10) == 0) {
        pos = (lrand48 () % (len - 8));
        str[pos] = 'O';
        str[pos + 1] = 'R';
        str[pos + 2] = 'I';
        str[pos + 3] = 'G';
        str[pos + 4] = 'I';
        str[pos + 5] = 'N';
        str[pos + 6] = 'A';
        str[pos + 7] = 'L';
    }
}

void randnum (str, len)
char *str;
int len;
{
    int i;

    for (i = 0; i < len; i++)
        str[i] = (char) (lrand48 () % 10 + '0');
    str[len] = '\0';
}

```

```

}

void randlastname (str, id)
char *str;
int id;
{
    id = id % 1000;
    strcpy (str, lastname[id / 100]);
    strcat (str, lastname[(id / 10) % 10]);
    strcat (str, lastname[id % 10]);
}

int NURand (A, x, y, cnum)
int A, x, y, cnum;
{
    int a, b;

    a = lrand48 () % (A + 1);
    b = (lrand48 () % (y - x + 1)) + x;
    return (((a | b) + cnum) % (y - x + 1)) + x;
}

void sysdate (sdate)
char *sdate;
{
    time_t tp;
    struct tm *tmptr;

    time (&tp);
    tmptr = localtime (&tp);
    strftime (sdate, 29, "%d-%b-%Y", tmptr);
}

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);
    fprintf(stderr,"Status - %s\n", status);
    return (IRRECERR);
}
return (RECOVERERR);
}

```

```

.....
loaditem.sh
.....

```

```

cd $tpcc_bench
$tpcc_load -M $tpcc_scale -i > loaditem.log
2>&1

```

```

.....
loadnord.sh
.....

```

```

rm -f loadnord*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 174720 -n -b 1 -e 682 >>
loadnord0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 683 -e 1364 >>
loadnord1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 1365 -e 2046 >>
loadnord2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 2047 -e 2728 >>
loadnord3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 2729 -e 3410 >>
loadnord4.log 2>&1 &
allprocs="$allprocs ${!}"

```

```

$tpcc_load -M 174720 -n -b 3411 -e 4092 >>
loadnord5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 4093 -e 4774 >>
loadnord6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 4775 -e 5456 >>
loadnord7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 5457 -e 6138 >>
loadnord8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 6139 -e 6820 >>
loadnord9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 6821 -e 7502 >>
loadnord10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 7503 -e 8184 >>
loadnord11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 8185 -e 8866 >>
loadnord12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 8867 -e 9548 >>
loadnord13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 9549 -e 10230 >>
loadnord14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 10231 -e 10912 >>
loadnord15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 10913 -e 11594 >>
loadnord16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 11595 -e 12276 >>
loadnord17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 12277 -e 12958 >>
loadnord18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 12959 -e 13640 >>
loadnord19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 13641 -e 14322 >>
loadnord20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 14323 -e 15004 >>
loadnord21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 15005 -e 15686 >>
loadnord22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 15687 -e 16368 >>
loadnord23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 16369 -e 17050 >>
loadnord24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 17051 -e 17732 >>
loadnord25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 17733 -e 18414 >>
loadnord26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 18415 -e 19096 >>
loadnord27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 19097 -e 19778 >>
loadnord28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 19779 -e 20460 >>
loadnord29.log 2>&1 &
allprocs="$allprocs ${!}"

```

```

$tpcc_load -M 174720 -n -b 20461 -e 21142 >>
loadnord30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 21143 -e 21824 >>
loadnord31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 21825 -e 22506 >>
loadnord32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 22507 -e 23188 >>
loadnord33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 23189 -e 23870 >>
loadnord34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 23871 -e 24552 >>
loadnord35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 24553 -e 25234 >>
loadnord36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 25235 -e 25916 >>
loadnord37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 25917 -e 26598 >>
loadnord38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 26599 -e 27280 >>
loadnord39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 27281 -e 27962 >>
loadnord40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 27963 -e 28644 >>
loadnord41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 28645 -e 29326 >>
loadnord42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 29327 -e 30008 >>
loadnord43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 30009 -e 30690 >>
loadnord44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 30691 -e 31372 >>
loadnord45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 31373 -e 32054 >>
loadnord46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 32055 -e 32736 >>
loadnord47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 32737 -e 33418 >>
loadnord48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 33419 -e 34100 >>
loadnord49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 34101 -e 34782 >>
loadnord50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 34783 -e 35464 >>
loadnord51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 35465 -e 36146 >>
loadnord52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 36147 -e 36828 >>
loadnord53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 36829 -e 37510 >>
loadnord54.log 2>&1 &
allprocs="$allprocs ${!}"

```



```

$tpcc_load -M 174720 -n -b 139888 -e 140570
>> loadnord205.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 140571 -e 141253
>> loadnord206.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 141254 -e 141936
>> loadnord207.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 141937 -e 142619
>> loadnord208.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 142620 -e 143302
>> loadnord209.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 143303 -e 143985
>> loadnord210.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 143986 -e 144668
>> loadnord211.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 144669 -e 145351
>> loadnord212.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 145352 -e 146034
>> loadnord213.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 146035 -e 146717
>> loadnord214.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 146718 -e 147400
>> loadnord215.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 147401 -e 148083
>> loadnord216.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 148084 -e 148766
>> loadnord217.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 148767 -e 149449
>> loadnord218.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 149450 -e 150132
>> loadnord219.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 150133 -e 150815
>> loadnord220.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 150816 -e 151498
>> loadnord221.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 151499 -e 152181
>> loadnord222.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 152182 -e 152864
>> loadnord223.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 152865 -e 153547
>> loadnord224.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 153548 -e 154230
>> loadnord225.log 2>&1 &
wait
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 154231 -e 154913
>> loadnord226.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 154914 -e 155596
>> loadnord227.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 155597 -e 156279
>> loadnord228.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 156280 -e 156962
>> loadnord229.log 2>&1 &

```

```

allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 156963 -e 157645
>> loadnord230.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 157646 -e 158328
>> loadnord231.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 158329 -e 159011
>> loadnord232.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 159012 -e 159694
>> loadnord233.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 159695 -e 160377
>> loadnord234.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 160378 -e 161060
>> loadnord235.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 161061 -e 161743
>> loadnord236.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 161744 -e 162426
>> loadnord237.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 162427 -e 163109
>> loadnord238.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 163110 -e 163792
>> loadnord239.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 163793 -e 164475
>> loadnord240.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 164476 -e 165158
>> loadnord241.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 165159 -e 165841
>> loadnord242.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 165842 -e 166524
>> loadnord243.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 166525 -e 167207
>> loadnord244.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 167208 -e 167890
>> loadnord245.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 167891 -e 168573
>> loadnord246.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 168574 -e 169256
>> loadnord247.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 169257 -e 169939
>> loadnord248.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 169940 -e 170622
>> loadnord249.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 170623 -e 171305
>> loadnord250.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 171306 -e 171988
>> loadnord251.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 171989 -e 172671
>> loadnord252.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 172672 -e 173354
>> loadnord253.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 173355 -e 174037
>> loadnord254.log 2>&1 &

```

```

allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -n -b 174038 -e 174720
>> loadnord255.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
    wait $curproc
    error=`expr $? + $error`
done
exit `expr $error != 0`

.....
loadordrordl.sh
.....

rm -f loadordrordl*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys0.dat -b 1 -e 682
>> loadordrordl0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys1.dat -b 683 -e
1364 >> loadordrordl1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys2.dat -b 1365 -e
2046 >> loadordrordl2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys3.dat -b 2047 -e
2728 >> loadordrordl3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys4.dat -b 2729 -e
3410 >> loadordrordl4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys5.dat -b 3411 -e
4092 >> loadordrordl5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys6.dat -b 4093 -e
4774 >> loadordrordl6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys7.dat -b 4775 -e
5456 >> loadordrordl7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys8.dat -b 5457 -e
6138 >> loadordrordl8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys9.dat -b 6139 -e
6820 >> loadordrordl9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys10.dat -b 6821 -e
7502 >> loadordrordl10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys11.dat -b 7503 -e
8184 >> loadordrordl11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys12.dat -b 8185 -e
8866 >> loadordrordl12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
$(tpcc_disks_location)dummys13.dat -b 8867 -e
9548 >> loadordrordl13.log 2>&1 &
allprocs="$allprocs ${!}"

```



```

$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy236.dat -b 161061
-e 161743 >> loadordrordl236.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy237.dat -b 161744
-e 162426 >> loadordrordl237.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy238.dat -b 162427
-e 163109 >> loadordrordl238.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy239.dat -b 163110
-e 163792 >> loadordrordl239.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy240.dat -b 163793
-e 164475 >> loadordrordl240.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy241.dat -b 164476
-e 165158 >> loadordrordl241.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy242.dat -b 165159
-e 165841 >> loadordrordl242.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy243.dat -b 165842
-e 166524 >> loadordrordl243.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy244.dat -b 166525
-e 167207 >> loadordrordl244.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy245.dat -b 167208
-e 167890 >> loadordrordl245.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy246.dat -b 167891
-e 168573 >> loadordrordl246.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy247.dat -b 168574
-e 169256 >> loadordrordl247.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy248.dat -b 169257
-e 169939 >> loadordrordl248.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy249.dat -b 169940
-e 170622 >> loadordrordl249.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy250.dat -b 170623
-e 171305 >> loadordrordl250.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy251.dat -b 171306
-e 171988 >> loadordrordl251.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy252.dat -b 171989
-e 172671 >> loadordrordl252.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy253.dat -b 172672
-e 173354 >> loadordrordl253.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy254.dat -b 173355
-e 174037 >> loadordrordl254.log 2>&1 &

```

```

allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -o
${tpcc_disks_location}dummy255.dat -b 174038
-e 174720 >> loadordrordl255.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
  wait $curproc
  error=`expr $? + $error`
done
exit `expr $error != 0`

.....
loadstok.sh
.....

rm -f loadstok*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 174720 -S -j 1 -k 48 >>
loadstok0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 49 -k 96 >>
loadstok1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97 -k 144 >>
loadstok2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 145 -k 192 >>
loadstok3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 193 -k 240 >>
loadstok4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 241 -k 288 >>
loadstok5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 289 -k 336 >>
loadstok6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 337 -k 384 >>
loadstok7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 385 -k 432 >>
loadstok8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 433 -k 480 >>
loadstok9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 481 -k 528 >>
loadstok10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 529 -k 576 >>
loadstok11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 577 -k 624 >>
loadstok12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 625 -k 672 >>
loadstok13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 673 -k 720 >>
loadstok14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 721 -k 768 >>
loadstok15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 769 -k 816 >>
loadstok16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 817 -k 864 >>
loadstok17.log 2>&1 &
allprocs="$allprocs ${!}"

```

```

$tpcc_load -M 174720 -S -j 865 -k 912 >>
loadstok18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 913 -k 960 >>
loadstok19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 961 -k 1008 >>
loadstok20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1009 -k 1056 >>
loadstok21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1057 -k 1104 >>
loadstok22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1105 -k 1152 >>
loadstok23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1153 -k 1200 >>
loadstok24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1201 -k 1248 >>
loadstok25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1249 -k 1296 >>
loadstok26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1297 -k 1344 >>
loadstok27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1345 -k 1392 >>
loadstok28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1393 -k 1440 >>
loadstok29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1441 -k 1488 >>
loadstok30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1489 -k 1536 >>
loadstok31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1537 -k 1584 >>
loadstok32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1585 -k 1632 >>
loadstok33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1633 -k 1680 >>
loadstok34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1681 -k 1728 >>
loadstok35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1729 -k 1776 >>
loadstok36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1777 -k 1824 >>
loadstok37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1825 -k 1872 >>
loadstok38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1873 -k 1920 >>
loadstok39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1921 -k 1968 >>
loadstok40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 1969 -k 2016 >>
loadstok41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 2017 -k 2064 >>
loadstok42.log 2>&1 &
allprocs="$allprocs ${!}"

```



```
$tpcc_load -M 174720 -S -j 97306 -k 97354 >>
loadstok1993.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97355 -k 97403 >>
loadstok1994.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97404 -k 97452 >>
loadstok1995.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97453 -k 97501 >>
loadstok1996.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97502 -k 97550 >>
loadstok1997.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97551 -k 97599 >>
loadstok1998.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97600 -k 97648 >>
loadstok1999.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97649 -k 97697 >>
loadstok2000.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97698 -k 97746 >>
loadstok2001.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97747 -k 97795 >>
loadstok2002.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97796 -k 97844 >>
loadstok2003.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97845 -k 97893 >>
loadstok2004.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97894 -k 97942 >>
loadstok2005.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97943 -k 97991 >>
loadstok2006.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 97992 -k 98040 >>
loadstok2007.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98041 -k 98089 >>
loadstok2008.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98090 -k 98138 >>
loadstok2009.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98139 -k 98187 >>
loadstok2010.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98188 -k 98236 >>
loadstok2011.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98237 -k 98285 >>
loadstok2012.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98286 -k 98334 >>
loadstok2013.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98335 -k 98383 >>
loadstok2014.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98384 -k 98432 >>
loadstok2015.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98433 -k 98481 >>
loadstok2016.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98482 -k 98530 >>
loadstok2017.log 2>&1 &
allprocs="$allprocs ${!}"
```

```
$tpcc_load -M 174720 -S -j 98531 -k 98579 >>
loadstok2018.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98580 -k 98628 >>
loadstok2019.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98629 -k 98677 >>
loadstok2020.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98678 -k 98726 >>
loadstok2021.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98727 -k 98775 >>
loadstok2022.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98776 -k 98824 >>
loadstok2023.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98825 -k 98873 >>
loadstok2024.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98874 -k 98922 >>
loadstok2025.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98923 -k 98971 >>
loadstok2026.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 98972 -k 99020 >>
loadstok2027.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99021 -k 99069 >>
loadstok2028.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99070 -k 99118 >>
loadstok2029.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99119 -k 99167 >>
loadstok2030.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99168 -k 99216 >>
loadstok2031.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99217 -k 99265 >>
loadstok2032.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99266 -k 99314 >>
loadstok2033.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99315 -k 99363 >>
loadstok2034.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99364 -k 99412 >>
loadstok2035.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99413 -k 99461 >>
loadstok2036.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99462 -k 99510 >>
loadstok2037.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99511 -k 99559 >>
loadstok2038.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99560 -k 99608 >>
loadstok2039.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99609 -k 99657 >>
loadstok2040.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99658 -k 99706 >>
loadstok2041.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99707 -k 99755 >>
loadstok2042.log 2>&1 &
allprocs="$allprocs ${!}"
```

```
$tpcc_load -M 174720 -S -j 99756 -k 99804 >>
loadstok2043.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99805 -k 99853 >>
loadstok2044.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99854 -k 99902 >>
loadstok2045.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99903 -k 99951 >>
loadstok2046.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 174720 -S -j 99952 -k 100000 >>
loadstok2047.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
    wait $curproc
    error= expr $? + $error
done
exit `expr $error / 0`

.....
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
!

.....
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

.....
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;
END;
/

```

```

show errors;

set echo off;

REM
REM end plsql_mon.sql
REM

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

@$tpcc_sql_dir/extent

@$tpcc_sql_dir/freeext

exit sql.sqlcode;
!

.....
createts.sh

```

```

.....

# Tablespace ware, ts size 400M (409600K)
# each file 400M (409600K)
# extents 399064K (399064K)
# 1 files

$tpcc_createts ware 1 1 400M 399064K unix
0 0 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for ware failed.
Exiting.
exit 0
fi

# Tablespace dist, ts size 3900M (3993600K)
# each file 3900M (3993600K)
# extents 1991224K (1991224K)
# 1 files

$tpcc_createts dist 1 1 3900M 1991224K
unix 0 769 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for dist failed.
Exiting.
exit 0
fi

# Tablespace hist, ts size 522240M
(534773760K)
# each file 2720M (2785280K)
# extents 102758K (102758K)
# 192 files

$tpcc_createts hist 192 1 2720M 102758K
unix 0 770 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for hist failed.
Exiting.
exit 0
fi

# Tablespace item, ts size 20M (20480K)
# each file 20M (20480K)
# extents 16892K (16892K)
# 1 files

$tpcc_createts item 1 1 20M 16892K unix 0
1730 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for item failed.
Exiting.
exit 0
fi

# Tablespace ordr, ts size 7278720M
(7453409280K)
# each file 37910M (38819840K)
# extents 103232K (103232K)
# 192 files

$tpcc_createts ordr 192 1 37910M 103232K
unix 0 1731 128 16K t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for ordr failed.
Exiting.
exit 0
fi

# Tablespace nord, ts size 64980M (66539520K)
# each file 7220M (7393280K)
# extents 738136K (738136K)
# 9 files

```

```

$tpcc_createts nord 9 1 7220M 738136K
unix 0 1923 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for nord failed.
Exiting.
exit 0
fi

# Tablespace iware, ts size 240M (245760K)
# each file 240M (245760K)
# extents 233524K (233524K)
# 1 files

$tpcc_createts iware 1 1 240M 233524K
unix 0 1932 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for iware failed.
Exiting.
exit 0
fi

# Tablespace icust1, ts size 133890M
(137103360K)
# each file 44630M (45701120K)
# extents 44624K (44624K)
# 3 files

$tpcc_createts icust1 3 1 44630M 44624K
unix 0 1933 128 16K t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for icust1 failed.
Exiting.
exit 0
fi

# Tablespace icust2, ts size 483840M
(495452160K)
# each file 2520M (2580480K)
# extents 2512K (2512K)
# 192 files

$tpcc_createts icust2 192 1 2520M 2512K
unix 0 1936 128 16K t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for icust2 failed.
Exiting.
exit 0
fi

# Tablespace idist, ts size 920M (942080K)
# each file 920M (942080K)
# extents 931024K (931024K)
# 1 files

$tpcc_createts idist 1 1 920M 931024K unix
0 2128 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for idist failed.
Exiting.
exit 0
fi

# Tablespace istok, ts size 394730M
(404203520K)
# each file 56390M (57743360K)
# extents 56384K (56384K)
# 7 files

$tpcc_createts istok 7 1 56390M 56384K
unix 0 2129 128 16K t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for istok failed.
Exiting.
exit 0
fi

```

```

# Tablespace iitem, ts size 20M (20480K)
# each file 20M (20480K)
# extents 11264K (11264K)
# 1 files

$tpcc_createts iitem 1 1 20M 11264K unix 0
2136 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for iitem failed.
Exiting.
exit 0
fi

# Tablespace iordr2, ts size 499200M
(511180800K)
# each file 2600M (2662400K)
# extents 2592K (2592K)
# 192 files

$tpcc_createts iordr2 192 1 2600M 2592K
unix 0 2137 128 16K t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for iordr2 failed.
Exiting.
exit 0
fi

# Tablespace temp, ts size 975360M
(998768640K)
# each file 5080M (5201920K)
# extents 199742K (199742K)
# 192 files

$tpcc_createts temp 192 1 5080M 199742K
unix 1 2329 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for temp failed.
Exiting.
exit 0
fi

.....
loadware.sh
.....

cd $tpcc_bench
$tpcc_load -M $tpcc_scale -w > loadware.log
2>&1

.....
tkvcin.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 initpcc;
/
show errors;

CREATE OR REPLACE PACKAGE BODY
initpcc 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 initpcc;
/
show errors
exit

.....
createuser.sql
.....

spool createusertpcc.log;

set echo on;

create user tpcc identified by tpcc;

grant dba to tpcc;

set echo off;
spool off;

exit ;

.....
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 /*+ leading(s) use_nl(i) */
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;

.....
bcexpr.sh
.....

#!/bin/sh
# send command line to bc
echo "$*" | bc

.....
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=$7
if test -z "$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 / $tpcc_np`
  procpemode=`$tpcc_bcexpr $tpcc_cpu \* 2`
  curstuff=`$tpcc_bcexpr $warepernode \*
\($tpcc_rac_node - 1\) + 1`
  stuffinc=`expr $warepernode / $procpemode`
  stuffextra=`expr $warepernode \%
$procpemode`
  curloader=`$tpcc_bcexpr $procpemode \*
\($tpcc_rac_node - 1\) + 1`
  endloader=`$tpcc_bcexpr $procpemode \*
$tpcc_rac_node + 1`

  while expr $curloader \< $endloader >
/dev/null ; do

    newstuff=`expr $curstuff + $stuffinc +
\($stuffextra / $procpemode \)`
    if expr x$6 = xt > /dev/null; then
      if expr $tpcc_os = unix > /dev/null; then

adddummy=\${tpcc_disks_location}dummy${curl
oader}.dat
      else
        # is this what we actually want to do?
check nt stuff
        adddummy=\\\\\\\\\\\\\\\\dummy${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=${tpcc_disks_location}dummy${curlo
ader}.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 ${1}" >> $loadout
curstuff=$newstuff

stuffextra=`expr $stuffextra + 1`
curloader=`expr 1 + $curloader`
done
fi

cat >> $loadout <<!
error=0
for curproc in `allprocs`; do
wait `lcurproc`
error=`expr $? + $error`
done
exit `expr $error != 0`
!

exit 0

.....
extractcols.sh
.....

#!/bin/sh

extractcols(){
table=$1
tablecols=`tp $table cols`

lines=`echo "$tablecols" | sed -e's/^ */' | cut -d'
-f1 | \
sed -e's/^(.*)/1"/, /' | tr -s '\n' ' '
echo "tablecols[${table}] = [${lines}]" | sed -e's/
$/./'
}

defaultcols(){
table=$1
tableinds=`tp $table indices`

indarr=`echo "$tableinds" | sed -e's/^([0-9][0-
9]*)/1,./g`
echo "tableinds[${table}] = [${indarr};" | sed -
e's/-//g' | sed -e's/,/ /' | sed -e's/\[no\]/ /g'
}

../stepenv.sh
.$tpcc_scripts/taledata.sh
for table in $tpcc_table_list; do
extractcols $table
done

for table in $tpcc_table_list $tpcc_index_list; do
defaultcols $table

.....
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
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

.....
isneg.sh
.....

#!/bin/sh
# exit true if negative, else false

if test ` $tpcc_bcexpr "$*" | cut -b1` = -; then
exit 0
else
exit 1
fi

.....
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

.....
lib/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:
#
# dpbibuniox.o
#

all: compile dpbibuniox.o

#include
$(ORACLE_HOME)/bench/buildtools/prefix.mk
L_SYM=-I
#include
$(ORACLE_HOME)/rdbms/lib/env_rdbms.mk
REMOVE=rm
#CC=/opt/SunProd/SUNWsp6.1/bin/./WS6U1/
bin/cc
CC=/usr/bin/gcc

TARGS=compile cleanup

TPCBIN=.
INCLUDE=$(L_SYM).
$(L_SYM)$(ORACLE_HOME)/rdbms/demo \
$(L_SYM)$(ORACLE_HOME)/rdbms/public \
$(L_SYM)$(ORACLE_HOME)/rdbms/include \
$(L_SYM)$(ORACLE_HOME)/plsqli/public \
$(L_SYM)$(ORACLE_HOME)/network/public
ITUX=$(L_SYM)$(ROOTDIR)/include

MEMBS=
OBSJS=gettime.o dpbproc.o dpbwait.o dpbpchk.o
dpbtimef.o

CFLAGS=

files:

compile: $(OBSJS)
@-$(DOTARGS)

cleanup:
$(REMOVE) $(OBSJS) dpbibuniox.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

dpbibuniox.o: $(OBSJS)
$(LD) -r -o $@ $(OBSJS)

c_trans_tux.o: $(CTRANTUX_OBSJS)
$(LD) -r -o $@ $(CTRANTUX_OBSJS)

```

```

.....
lib/dpbccore.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 __dpbccore__
#define __dpbccore__

#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__ */

#ifndef __dpbccore__
.....
lib/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 */

.....
lib/dpbetime.c
.....

/* Copyright (c) Oracle Corporation 1995. All
Rights Reserved. */

/*
NAME DPBETIME.C

DESCRIPTION

```

```

Get elapsed time in 10ths of milliseconds 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
milliseconds */

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 */
}

.....
lib/dpbfsync.c
.....

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

/*
NAME DPBFSYNC.C

DESCRIPTION
Flush o/s buffers to disk for a file.

TPC Benchmark C Full Disclosure

```

Calling `fclose()` or `fflush()` is not enough. These calls will only flush the buffer in the FILE structure 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)
{
    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 */
```

```
.....
lib/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 samtgjid;
```

```
#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 */
    samtgjid = GetThreadGroupID(); /* get this
program's thread group id */
```

```
# else /* ORA_NW */
```

```
    return; /* do nothing for everything else
*/
```

```
# endif /* ORA_NW */
}
```

```
.....
lib/dpboradt.c
.....
```

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

```
/*
NAME DPBORADT.C
```

DESCRIPTION
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;
        *oradt++ = 1;
    }
```



```

}
return;      /* do nothing for everything
else */
}

.....
lib/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 */

#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 */

.....
lib/dpbpcntl.h
.....

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

/*
NAME   DPBPCTL.H

DESCRIPTION
    OSD structures for process control

NOTES
    Desktop Performance Group

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

*/

#ifdef __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__ */

.....
lib/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");

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);
}
else
{
free(path);
return 0;
};
}
#endif /* ORA_AUX */

.....
lib/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 - Initial 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;
};

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;
};

this_thread=GetCurrentThread();
tpri=GetThreadPriority(this_thread);
switch (tpri)
{
case THREAD_PRIORITY_IDLE:
strcpy(os_pri,"1");
break;

case THREAD_PRIORITY_LOWEST:
strcpy(os_pri,"2");
break;

case THREAD_PRIORITY_BELOW_NORMAL:
strcpy(os_pri,"3");
break;

case THREAD_PRIORITY_NORMAL:
strcpy(os_pri,"4");
break;

case THREAD_PRIORITY_ABOVE_NORMAL:
strcpy(os_pri,"5");
break;

case THREAD_PRIORITY_HIGHEST:
strcpy(os_pri,"6");
break;

case THREAD_PRIORITY_TIME_CRITICAL:
strcpy(os_pri,"7");
break;

default:
strcpy(os_pri,"?");
break;
};

return 0;
#endif /* ORA_NT */
}

.....
lib/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

#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 */

.....
lib/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);
    };
}
#endif /* ORA_DOS */

.....
lib/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 samtgid;
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 oldtid;

oldtid = SetThreadGroupID(samtid); /*
switch to this NLM's thread group */
OraClientExit(samtid); /* cleanup
Oracle connection */
SetThreadGroupID(oldtid); /* reset the
thread group */

#else /* ORA_NW */

return; /* do nothing for everything else
*/

#endif /* ORA_NW */
}

.....:
lib/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_P SX)

# define GET_P_STATS
#endif /* sequent */

#if defined(aix) || defined(AIXRIOS)
# define TIME_W_GETTIME
# define CPU_W_TIMES
# define GETRU_STATS
#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 */
# if !defined(_SC_CLK_TCK)
# include <unistd.h>
# endif
#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 */

}

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
        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_invocsw);
    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);
}

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 */
```

```

.....
lib/tstetime.c
.....

```

```

#include <windows.h>
#include <systypes.h>
#include <time.h>

```

```
clock_t dpbetime();
```

```
main()
{
```

```

    clock_t begin, middle, end;

    begin = dpbetime();
    Sleep(2000);
    middle = dpbetime();
    Sleep(2000);
    end = dpbetime();
    printf(" begin = %lu\n middle = %lu\n end
= %lu\n",begin,middle,end);
}

```

Appendix F: 60 Day Space Calculation

TPM	2,196,268.05							
Warehouses	174,720							
SEGMENT	TYPE	TSPACE	BLOCKS	BLOCK_SIZE	KB	FIVE_PCT (KB)	DAILY_GROW(KB)	TOTAL(KB)
CUSTCLUSTER	CLUSTER	CUST	2,796,202,505	2,048	5,592,405,010	279,620,250	0	5,872,025,260
DB_STAT	SYS	SYSTEM	1,048,576	2,048	2,097,152	0		2,097,152
DCLUSTER	CLUSTER	DIST	1,991,224	2,048	3,982,448	199,122	0	4,181,570
HIST	TABLE	HIST	77,737,940	4,096	310,951,760	0	62,279,828	373,231,588
ICUST1	INDEX	ICUST1	7,245,822	16,384	115,933,152	5,796,656	0	121,729,808
ICUST2	INDEX	ICUST2	15,764,213	16,384	252,227,408	12,611,376	0	264,838,784
IDIST	INDEX	IDIST	465,512	2,048	931,024	46,552	0	977,576
IITEM	INDEX	ITEMS	5,632	2,048	11,264	564	0	11,828
IORDR2	INDEX	IORDR2	11,282,166	16,384	180,514,656	9,025,728	0	189,540,384
ISTOK	INDEX	ISTK	21,820,608	16,384	349,129,728	17,456,480	0	366,586,208
ITEMCLUSTER	CLUSTER	ITEMS	8,446	2,048	16,892	844	0	17,736
IWARE	INDEX	WARE	116,762	2,048	233,524	11,676	0	245,200
NORDCLUSTER_Q UEUE	CLUSTER	NORD	19,560,604	2,048	39,121,208	1,956,060	0	41,077,268
ORDRCLUSTER_Q UEUE	CLUSTER	ORDR	269,738,764	16,384	4,315,820,224	0	864,405,904	5,180,226,128
STOCKCLUSTER	CLUSTER	STOK	2,496,812,505	2,048	4,993,625,010	249,681,250	0	5,243,306,260
SYS_IQ0000020211 \$\$	INDEX	SYSTEM	738,136	2,048	1,476,272	73,814	0	1,550,086
SYS_IQ0000020214 \$\$	INDEX	SYSTEM	1,193,620	16,384	19,097,920	954,896	0	20,052,816
SYSAUX	SYS	SYSTEM	61,440	2,048	122,880	0	0	122,880
SYSTEM	SYS	SYSTEM	204,800	2,048	409,600	0	0	409,600
WCLUSTER	CLUSTER	WARE	199,532	2,048	399,064	19,954	0	419,018
Total			5,722,198,807		16,178,506,196	577,455,222	926,685,732	17,682,647,150
Dynamic space(KB)		4,626,771,984						
Static space(KB)		12,129,189,434						
Free space(KB)		926,685,732						
Daily growth(KB)		926,685,732						
Daily spread		0	Oracle may be configured such that daily spread is 0					
60-day (GB)		64,592.68						
Log KB/tpmC		5.10	KB of log used per New-Order					
8-hour log (GB)		5,127.40						
		Capacity	NumDisk				Total GB	
Database Disks	ETERNUS8000	67.99	3,072				208,865.28	
							208,865.28	
		Capacity	NumDisk(RAID0+1)					
8-Hr Log Disks	ETERNUS8000	67.99	384				26,108.16	
							26,108.16	

Client	c1123	c1124	c1125	c1126	c1127	c1128	c1129	c1130	c1131	c1132	c1133	c1134	c1135
tpmC	29134.77	29113.86	29128.12	29086.02	29118.00	29138.93	29186.80	29077.76	29154.17	29054.90	29067.78	29067.64	29096.56
user	23000	23000	23000	23000	23000	23000	23000	23000	23000	23000	23000	23000	23000
menu-ave	0.104	0.104	0.104	0.104	0.104	0.104	0.103	0.103	0.103	0.103	0.103	0.103	0.103
menu-ninety	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104
NewOrder													
new-ave	0.177	0.189	0.194	0.213	0.195	0.172	0.191	0.221	0.172	0.203	0.214	0.211	0.199
new-ninety	0.149	0.152	0.190	0.220	0.183	0.163	0.173	0.202	0.152	0.153	0.202	0.219	0.194
new-think	12.015	12.017	12.007	12.021	12.003	12.020	11.995	12.021	12.026	12.022	12.019	12.009	12.013
new-keying	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012
new-mix	44.920	44.910	44.950	44.920	44.930	44.930	44.970	44.930	44.970	44.920	44.910	44.920	44.910
Payment													
pay-ave	0.163	0.175	0.181	0.197	0.181	0.161	0.177	0.204	0.160	0.186	0.199	0.196	0.185
pay-ninety	0.140	0.143	0.181	0.210	0.174	0.154	0.164	0.192	0.144	0.145	0.193	0.209	0.184
pay-think	12.016	12.011	12.019	12.013	12.026	12.017	12.013	12.014	12.014	12.023	12.030	12.018	12.027
pay-keying	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012
pay-mix	43.030	43.040	43.010	43.030	43.010	43.030	43.000	43.010	42.990	43.020	43.050	43.040	43.030
OrderStatus													
ord-ave	0.169	0.181	0.188	0.202	0.186	0.168	0.185	0.215	0.164	0.190	0.207	0.203	0.192
ord-ninety	0.149	0.152	0.189	0.217	0.182	0.164	0.175	0.203	0.153	0.154	0.201	0.217	0.193
ord-think	10.038	10.058	9.990	10.008	10.017	10.006	9.987	10.019	10.038	10.052	10.022	10.028	9.974
ord-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
ord-mix	4.020	4.020	4.010	4.020	4.020	4.020	4.010	4.020	4.010	4.030	4.010	4.010	4.010
Delivery													
del-ave	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
del-ninety	0.104	0.104	0.104	0.104	0.104	0.104	0.103	0.103	0.103	0.103	0.103	0.103	0.103
del-think	5.004	5.034	5.025	5.023	5.027	5.027	5.016	5.020	5.005	5.020	5.016	5.020	5.027
del-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
del-mix	4.010	4.010	4.020	4.010	4.020	4.010	4.010	4.030	4.010	4.020	4.010	4.010	4.020
StockLevel													
stk-ave	0.154	0.168	0.173	0.191	0.175	0.154	0.170	0.197	0.153	0.181	0.193	0.187	0.178
stk-ninety	0.132	0.135	0.173	0.203	0.165	0.146	0.156	0.185	0.136	0.137	0.186	0.202	0.176
stk-think	5.032	5.030	5.025	5.023	5.027	5.029	5.022	5.005	5.011	5.029	4.997	5.019	5.019
stk-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
stk-mix	4.020	4.020	4.010	4.020	4.020	4.010	4.010	4.010	4.020	4.010	4.020	4.020	4.030
NewOrder-txn	3496173	3493664	3495375	3490323	3494160	3496432	3497856	3489332	3498501	3491389	3488132	3490517	3491636

Client	c1136	c1137	c1138	c1139	c1140
tpmC	29112.04	29133.49	29008.48	29151.21	29106.03
user	23000	23000	23000	23000	23000
menu-ave	0.103	0.103	0.103	0.103	0.103
menu-ninety	0.104	0.104	0.104	0.104	0.104
NewOrder					
new-ave	0.213	0.194	0.273	0.178	0.198
new-ninety	0.216	0.174	0.237	0.150	0.156
new-think	12.010	12.013	12.027	12.008	12.024
new-keying	18.012	18.012	18.012	18.012	18.012
new-mix	44.970	44.950	44.940	44.930	44.930
Payment					
pay-ave	0.198	0.180	0.245	0.165	0.182
pay-ninety	0.207	0.165	0.227	0.141	0.148
pay-think	12.009	12.007	12.026	12.007	12.011
pay-keying	3.012	3.012	3.012	3.012	3.012
pay-mix	43.000	43.000	43.010	43.010	43.020
OrderStatus					
ord-ave	0.205	0.186	0.254	0.169	0.189
ord-ninety	0.217	0.175	0.237	0.150	0.157
ord-think	10.033	9.991	10.032	10.025	10.021
ord-keying	2.012	2.012	2.012	2.012	2.012
ord-mix	4.010	4.030	4.030	4.020	4.020
Delivery					
del-ave	0.103	0.103	0.103	0.103	0.103
del-ninety	0.103	0.103	0.103	0.103	0.103
del-think	5.014	5.024	5.020	5.021	5.017
del-keying	2.012	2.012	2.012	2.012	2.012
del-mix	4.010	4.010	4.010	4.020	4.010
StockLevel					
stk-ave	0.190	0.171	0.236	0.157	0.176
stk-ninety	0.199	0.157	0.219	0.133	0.140
stk-think	5.020	5.002	5.014	5.005	5.017
stk-keying	2.012	2.012	2.012	2.012	2.012
stk-mix	4.010	4.010	4.010	4.020	4.020
NewOrder-txn	3493445	3496019	3481018	3498146	3492724

Appendix H: Price Quotes

From: MaryBeth Pierantoni [mailto:mary.beth.pierantoni@oracle.com]
 Sent: Tuesday, October 30, 2007 7:16 AM
 To: sky@jp.fujitsu.com
 Subject: Oracle Price

Product	Price	Quantity	Extended Price
Oracle Database 10g Enterprise Edition, Per Processor, Unlimited Users,3 years	20,000	32	640,000
Oracle Database Server Support Package for 3 years	2,000	3	6,000
Partitioning, Per Processor, Unlimited Users for 3 years	5,000	32	160,000
Oracle Mandatory E-Business Discount			<161,200>
TOTAL			644,800

*32 = 0.50 * 64. Explanation: For the purpose of counting the number of processors which require licensing, an Intel multicore chip with “n” cores shall be determined by multiplying “n” cores by a factor of .50.



October 3, 2007

Shin'ichi Kurogi, Manager
TRIOLE Technology Development Div., Software Unit
Fujitsu Ltd.
NOF Shin-Yokohama Bldg, 2-15-16
Shin-Yokohama, Kohoku-ku, Yokohama,
Kanagawa Pref, Japan

Per your request I am enclosing the pricing information regarding TUXEDO 8.1 that you requested. Core functionality services (CFS)-R pricing is appropriate for your activities. Server 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 PRIMERGY RX200 S3 dual core based servers are Tier 1 machines – price is \$1,200 per server (License), eligible for a 10% discount = \$1,080 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. 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,

A handwritten signature in black ink that reads "Robert J. Gieringer". The signature is written in a cursive style with a large, prominent "R" and "G".

Rob Gieringer,
Worldwide Pricing Director

10/03/2007

BEA SYSTEMS, INC.

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

Cisco Catalyst 2950T [WS-C2950T-24]
Fast Ethernet Desktop Switch
24 10/100 Autosensing P...

http://www.computeronline.com/cisco2950t24.html

COL - Computer Online

CALL TOLL FREE: 877.377.2250
Mon-Fri 8:00AM - 7:00PM PST

Search [input type="text"] GO

Chat Online Live

COL HOME
UP THIS SECTION

- Networking Hardware
- Projectors
- Computer Hardware
- Computer Software
- Notebooks
- Computer Systems
- Plasma Displays
- Monitors/Displays
- Storage
- Printers/Scanners
- Consumer Electronics

ABOUT US
CONTACT US

Call us to get a 2% cash discount off our low prices.

SECURE Ordering Fraud Protection Guaranteed.

FREE UPS Ground Shipping Offer!



Cisco Catalyst 2950T [WS-C2950T-24] Fast Ethernet Desktop Switch 24 10/100 Autosensing Ports 2 Fixed Gigabit Ethernet Ports

The **Cisco Catalyst 2950T-24** is a fixed configuration, wire-speed Fast Ethernet desktop switch which delivers premium performance and functionality for local-area networks (LANs). The Cisco Catalyst 2950T-24 is a standalone, 10/100 autosensing switch that provides enhanced quality of service (QoS) and multicast management features—managed with the easy-to-use, Web-based Cisco Cluster Management Suite (CMS) and integrated Cisco IOS Software. The Cisco Catalyst 2950T-24 offers medium-sized businesses and enterprise branch offices with an ideal solution to migrate from Fast Ethernet to a higher-performance Gigabit Ethernet backbone using existing Category 5 copper cabling.

The Cisco Catalyst 2950T-24 has 24 10/100 ports with 2 fixed 10/100/1000Base T uplink ports. It has a one rack-unit (RU) form factor, making them very flexible to deploy, either on a desktop or mounted in a wiring closet.

Features:

- Wire-speed, nonblocking performance on all ports, including Gigabit ports
- 8.8-Gbps switching fabric and 6.6 million packets-per-second maximum forwarding rate ensures maximum throughput—even for the most performance-sensitive applications
- 24- 10Base T/100Base TX autosensing ports, each delivering up to 200 Mbps of bandwidth to individual users, servers or workgroups to support bandwidth-intensive applications
- *Catalyst 2950T-24* has two built-in, Gigabit Ethernet (1000Base T) ports that deliver up to 4 Gbps aggregated bandwidth to the Gigabit Ethernet backbone, Gigabit Ethernet servers or between switches—leveraging existing Category 5 cabling infrastructure—up to a distance of 100 meters
- 8 MB shared memory architecture ensures the highest possible throughput with a design that eliminates head-of-line blocking, minimizes packet loss, and delivers better overall performance in environments with extensive multicast and broadcast traffic
- 16 MB of DRAM and 8 MB of Flash on-board enable the addition of future feature upgrades, maximizing customer investments

Cisco Catalyst 2950T [WS-C2950T-24]
Fast Ethernet Desktop Switch
24 10/100 Autosensing P...

ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)

http://www.computeronline.com/cisco2950t24.html

RJ-45-to-DB25 female DTE adapter (can be ordered separately from Cisco. Part Number: ACS-DSBUSYN=)

Indicators

- ◆ Per-port status LEDs: link integrity, disabled, activity, speed, and full-duplex indications
- ◆ System status LEDs: system, RPS, and bandwidth utilization indications

Physical Dimensions

- ◆ Dimensions (H x W x D): 1.72 x 17.5 x 9.52 in. (4.36 x 44.45 x 24.18 cm)
- ◆ One rack-unit (RU) high (1.72 in./4.36 cm)
- ◆ Weight: 6.5 lbs (3.0 kg)

Environmental Ranges

- ◆ Operating temperature: 23 to 113° F (-5° C to 45° C)
- ◆ Storage temperature: -13 to 158° F (-25 to 70° C)
- ◆ Operating relative humidity: 10 to 95% (non-condensing)
- ◆ Operating altitude: Up to 10,000 ft (3,000 m)
- ◆ Storage Altitude: Up to 15,000 ft (4,500 m)

Power Requirements

- ◆ Power consumption: 30W (maximum), 102 BTUs per hour
- ◆ AC input voltage/frequency: 100 to 127 or 200 to 240 VAC (auto-ranging), 50 to 60 Hz
- ◆ DC Input Voltages: +12V @ 4.5A

Availability: Usually ships the next business day.

Cisco Catalyst 2950T [WS-C2950T-24]
Fast Ethernet Desktop Switch
24 10/100 Autosensing Ports
2 Fixed Gigabit Ethernet Ports
 WS-C2950T-24 Regular price: \$2,395.00 **Sale price: \$799.00**
Expedite: Same Day Order Processing and Shipping (+10)

Home | Shopping | Shipping | Policies | Forms | Apply for a Job | Feedback **Products Search**

Established 1985 San Jose, California
[Contact Us](#) | [Add to Favorites](#) | [Site Disclaimer](#)

All Products Listed on This Website Are Brand New

Copyright © 1997-2007, [Computer Online](#)
 All rights reserved. All trademarks and logos are properties of their respective legal owners.

COL - Computer Online CALL TOLL FREE: 877.377.2250
Mon-Fri 8:00AM - 7:00PM PST

Search

Chat Online Live

COL HOME
UP THIS SECTION

- Networking Hardware
- Projectors
- Computer Hardware
- Computer Software
- Notebooks
- Computer Systems
- Plasma Displays
- Monitors/Displays
- Storage
- Printers/Scanners
- Consumer Electronics

ABOUT US
CONTACT US

Cisco SMARTnet Support Service
CON-SNTP-PKG1 to CON-SNTP-PKG18
1 Year 4 Hour 24 x 7 Package
Category 1 - Category 18

Cisco® SMARTnet support services help protect your network investment by enabling you to extend and enhance the operational lifetime of your Cisco networking devices and Cisco IOS® Software. Cisco SMARTnet support services help enable improved productivity and can increase operational efficiency by complementing your in-house resources with world-class networking expertise. Cisco® SMARTnet support services can maximize availability and minimize risks for systems running mission-critical applications by delivering:

- Ongoing Cisco IOS Software updates, allowing you to efficiently evolve your network infrastructure to address the needs of an ever-changing business environment
- Rapid hardware and Cisco IOS Software technical problem resolution with 24-hour, global access to an extensive team of expert technical engineers to help resolve your network problems—online or on the telephone
- Knowledge transfer of Cisco expertise, to enhance in-house technical skill levels
- Advance hardware replacement, to help reduce the risk of network downtime
- Registered access to an array of powerful online tools, allowing you to more quickly address common network problems
- Around-the-clock access to comprehensive technical information and a collection of configuration, installation, troubleshooting, and case management tools
- A broad base of expertise in networking technology, including voice, video, and data communications

Cisco® SMARTnet support services accelerate your success by improving productivity, increasing operational efficiency, and extending the life of your network assets.

Cisco 1YR 4HR 24x7 SMARTnet support services deliver advance replacement parts within 4 hours of determining that part replacement is required (24 hours a day, 7 days a week).

Cisco SMARTnet Support Service
CON-SNTP-PKG1 to CON-SNTP-PKG18
1 Year 4 Hour 24 x 7...

ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)

http://www.computeronline.com/ciscosnet24x7.html

Cisco 1YR 4HR 24x7 SMARTnet support services deliver advance replacement parts within 4 hours of determining that part replacement is required (24 hours a day, 7 days a week).

Specifications:

Model No.	Brief Description	Price (\$)
CON-SNTP-PKG1	1YR 4HR 24x7 SMARTnet Cat 1	109
CON-SNTP-PKG2	1YR 4HR 24x7 SMARTnet Cat 2	189
CON-SNTP-PKG3	1YR 4HR 24x7 SMARTnet Cat 3	269
CON-SNTP-PKG4	1YR 4HR 24x7 SMARTnet Cat 4	399
CON-SNTP-PKG5	1YR 4HR 24x7 SMARTnet Cat 5	539
CON-SNTP-PKG6	1YR 4HR 24x7 SMARTnet Cat 6	719
CON-SNTP-PKG7	1YR 4HR 24x7 SMARTnet Cat 7	859
CON-SNTP-PKG8	1YR 4HR 24x7 SMARTnet Cat 8	1079
CON-SNTP-PKG9	1YR 4HR 24x7 SMARTnet Cat 9	1309
CON-SNTP-PKG10	1YR 4HR 24x7 SMARTnet Cat 10	1519
CON-SNTP-PKG11	1YR 4HR 24x7 SMARTnet Cat 11	2079
CON-SNTP-PKG12	1YR 4HR 24x7 SMARTnet Cat 12	2459
CON-SNTP-PKG13	1YR 4HR 24x7 SMARTnet Cat 13	3099
CON-SNTP-PKG14	1YR 4HR 24x7 SMARTnet Cat 14	4019
CON-SNTP-PKG15	1YR 4HR 24x7 SMARTnet Cat 15	4679
CON-SNTP-PKG16	1YR 4HR 24x7 SMARTnet Cat 16	7099
CON-SNTP-PKG17	1YR 4HR 24x7 SMARTnet Cat 17	9319
CON-SNTP-PKG18	1YR 4HR 24x7 SMARTnet Cat 18	13619

Availability: Usually ships the next business day.

**Cisco SMARTnet Support Service
CON-SNTP-PKG1 to CON-SNTP-PKG18
1 Year 4 Hour 24 x 7 Package
Category 1 - Category 18**

CON-SNTP-PKG1 Regular price: \$179.00 **Starting from: \$109.00**

Options:

Home | Shopping | Shipping | Policies | Forms | Apply for a Job | Feedback |

Established 1985 San Jose, California
[Contact Us](#) | [Add to Favorites](#) | [Site Disclaimer](#)

All Products Listed on This Website Are Brand New

Copyright © 1997-2007, [Computer Online](#)
All rights reserved. All trademarks and logos are properties of their respective legal owners.

Appendix I: Auditor's attestation letter



Benchmark Sponsor: Shin'ichi Kurogi
 Manager, TRIOLE Technology Development Division
 Software Unit
 Fujitsu Limited
 Shin-Yokohama Nikko Bldg.
 2-15-16 Shin-Yokohama, Kohoku-ku, Yokohama
 Kanagawa Pref. 222-0033, Japan

October 30, 2007

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

Platform: Fujitsu PRIMEQUEST 580 c/s
 Operating system: Red Hat Enterprise Linux 4 AS
 Database Manager: Oracle Database 10g R2 Enterprise Edition with Partitioning
 Transaction Manager: BEA Tuxedo 8.1

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: Fujitsu PRIMEQUEST 580				
32 x Itanium2 Dual-Core (1.6GHz)	2048 GB (24 MB L3)	3456 x 73 GB 15Krpm 1 x 73 GB 10Krpm int.	0.344 Seconds	2,196,268.05
Ninety-six (96) Client: Fujitsu PRIMERGY RX200 S3 (each with)				
2 x Xeon Dual-Core (3.0 GHz)	3.0 GB (4 MB L2)	1 x 73 GB SAS	n/a	n/a

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

The following verification items were given special attention:

- The transactions were correctly implemented
- The database records were the proper size

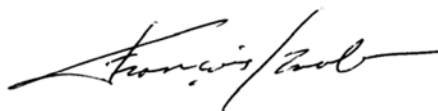
- The database was properly scaled and populated
- The ACID properties were met
- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured.
- At least 90% of all delivery 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:

The tested configuration included (1) priced ETERNUS8000 storage subsystem and (8) non-priced ETERNUS6000 storage subsystems. The priced configuration includes (9) ETERNUS8000 storage subsystems. These two types of storage subsystems were configured with the same number of controllers and disk drives. Based on the analysis of the performance data collected for each type of storage subsystems during the measurements, it is my opinion that this substitution has no significant effect on performance.

The tested configuration included a mix of PRIMERGY client systems, including one RX200 S3, forty-three RX200 S2, twelve F250 and forty C200. Based on the specifications of these client systems, they were substituted one-for-one by the more recent PRIMERGY RX200 R3 in the priced configuration.

Respectfully Yours,



François Raab, President