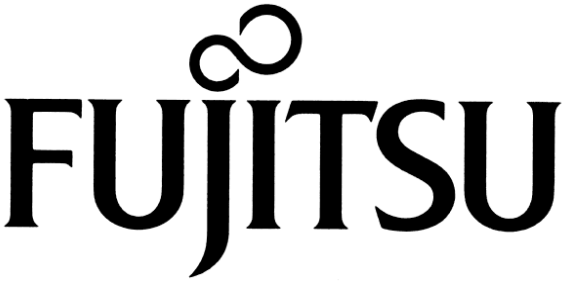


and

TPC Benchmark™C

Full Disclosure Report

Fujitsu
PRIMEQUEST 540A c/s
W/ 48 Front-Ends

running

Oracle Database 10g Release 2
Enterprise Edition

December 12, 2008

First Edition – November 22, 2008

Second Edition –December 12, 2008

The benchmark results contained in this document were submitted for compliance with version 5.10 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 December 2008.

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.

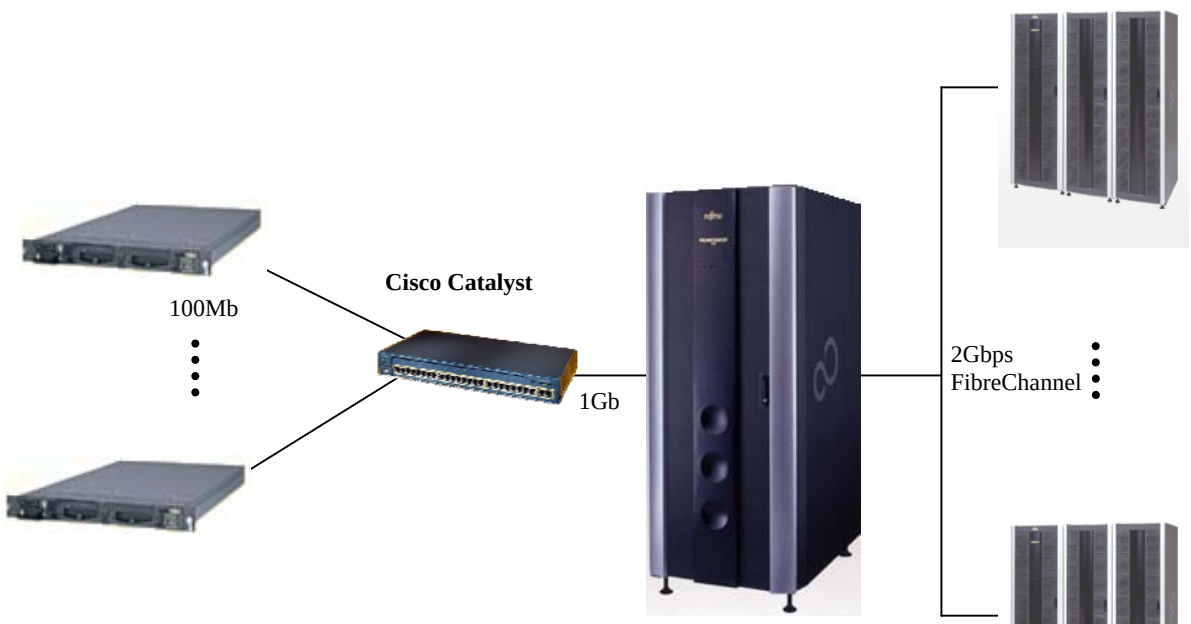
Benchmark results are highly dependent upon workload, specific application requirements, and system design and implementation. Relative system performance will vary as a result of these and other factors. Therefore TPC Benchmark C should not be used as a substitute for a specific customer application 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.

Printed in USA, December 2008.

FUJITSU ORACLE	**PRIMEQUEST 540A c/s w/48 Front-Ends**		**TPC-C Rev 5.10** **TPC Pricing 1.3.0** **Report Date: November 22,2008** **Revised: December 12, 2008**
Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$ 4,397,010 USD	**1,354,086 tpmC**	**$ 3.25 USD/tpmC**	**November 22, 2008**

Database Server Processors/Cores/Threads	Database Manager	Operating system	Other Software	Number of users
16/32/64 **Intel Itanium 1.66GHz**	**Oracle Database 10g Release 2 Enterprise Edition**	**Red Hat Enterprise Linux 4 AS**	**Tuxedo 8.1**	**1,084,800**



48 PRIMERGY RX200 S4
1 x 3.33GHz Dual-core Xeon
6MB L2 Cache
3GB Memory
73GB SAS Disk

PRIMEQUEST 540A
16 x 1.66GHz Dual-Core Intel Itanium
with 24MB L3 Cache
1TB Memory
1 x 73GB 10K RPM SCSI Drive

6 ETERNUS8000 Model900
2112 73GB 15K RPM Drives

System Component	Qty	Server:	Qty	Each of 48 Clients:
Processors/Cores/ Threads	16/32/64	1.66GHz Dual-Core Intel Itanium 9150M	1/2/2	3.33GHz Intel Dual-core Xeon X5260
Cache Memory		24MB L3 Cache		6MB L2 Cache
Memory	32	32GB (4 x 8GB DDR2-400)	3	1GB(2 x 512MB PC2-5300F) 2GB(2 x 1024MB PC2-5300F)
Disk Controllers	24	4G bps FibreChannel (used at 2Gbps)	1	Serial Attached SCSI (SAS)
Disk Drives	1 2112	73GB 10K rpm 73GB 15K rpm	1	73 GB 15K rpm
Total Storage		154,249 GB		3,504 GB

FUJITSU ORACLE

PRIMEQUEST 540A c/s w/48 Front-Ends

TPC-C Revision 5.10
TPC Pricing 1.3.0

Report Date: November 22, 2008
Revised Date : December 12, 2008

Description	Part Number	Third Party Brand Pricing	Unit Price	Qty	Extended Price	3Year Maint. Price
Server Hardware						
PRIMEQUEST 540A Base Unit	MC4C0P311U	1	14,000.00	1	14,000.00	136,392.00
System Board	MC-97SB11	1	18,000.00	4	72,000.00	
CPU Module(Dual core Itanium 9150M/1.66GHz/24MB	MC-01HA114	1	20,900.00	16	334,400.00	
32GB Memory Module (4x8GB DDR2-400)	MC-02A6114	1	32,880.00	32	1,052,160.00	
I/O Unit	MC-97UX11	1	15,000.00	4	60,000.00	
BMC Module	MC-97BM11	1	1,720.00	1	1,720.00	
Disk Drive Unit (3.5inch, 73GB, 10,000rpm, Ultra320)	MC-03D321	1	680.00	1	680.00	
Gigabit Switch Board (w/ 8 external 1000Base-T ports)	MC-87GE11	1	11,850.00	1	11,850.00	
External I/O Cabinet	MC-87RK11U	1	20,850.00	1	20,850.00	
PCI-Box	MC-07PB21U	1	20,700.00	1	20,700.00	
PCI Unit	MC-07PU31	1	4,250.00	2	8,500.00	
PCI Cassette	MC-07PC11	1	370.00	6	2,220.00	
PCI Unit Cable (5m)	MC-07CA11	1	890.00	2	1,780.00	
FibreChannel Card (4Gbps, PCI-X, dual port)	MC-08FC41	1	4,270.00	22	93,940.00	
FibreChannel Cable (15m, LC-LC)	CBL-MLLB15	1	330.00	44	14,520.00	
17" Flat Panel Display	96-343680-000	1	520.00	1	520.00	
USB Keyboard W/Trackball	MC-07KB51	1	600.00	1	600.00	
		Server Hardware Subtotals			1,710,440.00	136,392.00
Storage						
ETERNUS8000 Model900 Base Unit w/ 2 Controllers, 4 Drive Enclosures	E890S20AU	1	180,000.00	6	1,080,000.00	485,580.00
Additional Expansion Rack	E800CR1U	1	12,000.00	11	132,000.00	
Additional Controllers (2 sets)	E800CJ01U	1	40,000.00	6	240,000.00	
Cache Memory (2GBx4)	E800CM41U	1	14,000.00	6	84,000.00	
FibreChannel Host Interfaces (4Gbps, dual port, 2	E800CH14U	1	12,800.00	22	281,600.00	
Drive Enclosures for Base Unit (4 sets)	E800CE11U	1	39,400.00	6	236,400.00	
Drive Enclosures for Expansion Rack (4 sets)	E800CE21U	1	31,000.00	33	1,023,000.00	
Disk Drive Unit (73GB, 15,000rpm)	E800CA3U	1	650.00	2112	1,372,800.00	
		Storage Subtotals			4,449,800.00	485,580.00
Server Software						
Red Hat Enterprise Linux 4 AS (for Intel Itanium)	PMK0348US	1	7,497.00	1	7,497.00	7,497.00
Oracle Database 10g Enterprise Edition, Per Processor, Unlimited Users for 3 years		2	23,750.00	16*	380,000.00	
Oracle Database Server Support for 3 years		2	2,300.00	3		6,900.00
		Server Software Subtotals			387,497.00	14,397.00
Client Hardware						
PRIMERGY RX200 S4	S26361-K1167-V101	1	1,174.84	48	56,392.32	45,360.00
Xeon DP X5260 3.33 GHz 6MB 1333MHz	S26361-F3881-E333	1	1,422.00	48	68,256.00	
1GB Base Memory (2 x 512MB PC2-5300F)	S26361-F3263-E521	1	109.36	48	5,249.28	
2GB Memory Module (2 x 1024MB PC-5300F)	S26361-F3263-E522	1	149.42	48	7,172.16	
Drive Bay (2 x 3.5" HDD slots)	S26361-F3317-E400	1	45.00	48	2,160.00	
Hard Disk Drive (3.5inch, 73GB, 15,000rpm Hot Plug, S	S26361-F3204-E573	1	227.00	48	10,896.00	
RAID 0/1 SAS based on LSI MegaRAID 4Port	S26361-F3257-E4	1	127.00	48	6,096.00	
Internal CD-RW / DVD Unit	S26361-F3268-E1	1	68.40	48	3,283.20	
USB FDD	S26361-F2382-L20	1	42.34	1	42.34	
PRIMECENTER Rack 19inch Rack (24U)	S26361-K826-V102	1	2,011.01	3	6,033.03	
Dummy panel, plastics, 2U + assembly	S26361-F2735-E131	1	16.00	3	48.00	
RackMountKit 1-2U servers	S26361-F2735-E110	1	67.00	48	3,216.00	
Cable Powercord (USA) 1.8m, grey	T26139-Y1742-E10	1	11.00	48	528.00	
LCDRC23 17" TFT	S26361-K1023-V200	1	1,716.00	3	5,148.00	
KVM Switch (8ports, 1U)	S26361-F2293-E801	1	708.00	9	6,372.00	
Console Switch adapter KVM S2 PS/2-VGA	S26361-F2293-L201	1	110.00	48	5,280.00	
KVM Cable (1.8m)	S26361-F2293-L20	1	10.80	54	583.20	
		Client Hardware Subtotals			186,755.53	45,360.00

FUJITSU ORACLE

PRIMEQUEST 540A c/s w/48 Front-Ends

TPC-C Revision 5.10
TPC Pricing 1.3.0

Report Date: November 22, 2008
Revised Date : December 12, 2008

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)	PMK0343US	1	672.00	48	32,256.00	
Tuxedo CFS-R Tier 1		2	1,800.00	48	86,400.00	
Oracle Premium Support for 3 years		2	19,008.00	3		57,024.00
		Client Software Subtotals			118,656.00	57,024.00
User Connectivity						
Cisco Catalyst 2950T-24 Switch		3	839.00	4	3,356.00	
Cisco SMARTnet 24x7x4 Maintenance		4		4		1,439.88
		User Connectivity Subtotals			3,356.00	1,439.88
Oracle Mandatory E-Business Discount		2			-106,065.00	
Large Configuration Discount** and Support Prepayment***		1			-2,794,058.66	-299,564.70
				Total	3,956,381.00	440,629.00
				Three-Year Cost of Ownership		**$4,397,010**

Pricing Sources: 1 = Fujitsu , 2 = Oracle , 3 = getITnew.com , 4 = TechOnWeb.com
Audited by: Francois Raab, InfoSizing, Inc. (www.sizing.com)
Oracle pricing contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081
* 16=0.50 x 32. 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. ** applied to the Fujitsu hardware and storage price subtotals.
*** applied to the Fujitsu hardware, storage and server software maintenance.

Three-Year Cost of Ownership USD	**$4,397,010**
tpmC	**1,354,086**
$ USD / tpmC	**$3.25**

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 540A c/s w/ 48 Front-Ends

Oracle Database 10g Release 2 Enterprise Edition

MQTH, Computed Maximum Qualified Throughput **1,354,086 tpmC**

Response Times (in seconds)	**Average**	**90th %**	**Maximum**
New-Order	0.508	0.992	6.389
Payment	0.499	0.983	6.401
Order-Status	0.507	0.992	6.132
Delivery (interactive portion)	0.103	0.104	0.324
Delivery (deferred portion)	0.404	0.887	4.988
Stock-Level	0.491	0.976	6.137
Menu	0.103	0.104	0.358

Transaction Mix, in percent of total transaction

New-Order	44.96%
Payment	43.01%
Order-Status	4.01%
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	18.034	0.000	12.016	120.198
Payment	3.002	3.012	3.036	0.000	12.015	120.196
Order-Status	2.003	2.012	2.034	0.000	10.016	100.187
Delivery (interactive)	2.003	2.012	2.033	0.000	5.017	50.184
Stock-Level	2.003	2.012	2.030	0.000	5.018	50.193

Test Duration

Ramp-up time	179 Min. 0 Sec.
Measurement interval	120 minutes
Transactions during measurement interval(all types)	361,713,360

Checkpointing

Number of checkpoints	4
Checkpoint interval (seconds)	1,717

Table Of Contents

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C test conducted by Fujitsu Ltd. on the Fujitsu PRIMEQUEST 540A w/48 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.
The operating system on the clients was Red Hat Enterprise Linux 4 ES for x86.
Those clients ran Apache HTTP Server and 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:

1,354,086 tpmC
$3.25 USD/tpmC
November 22, 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 5.10.

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.).*

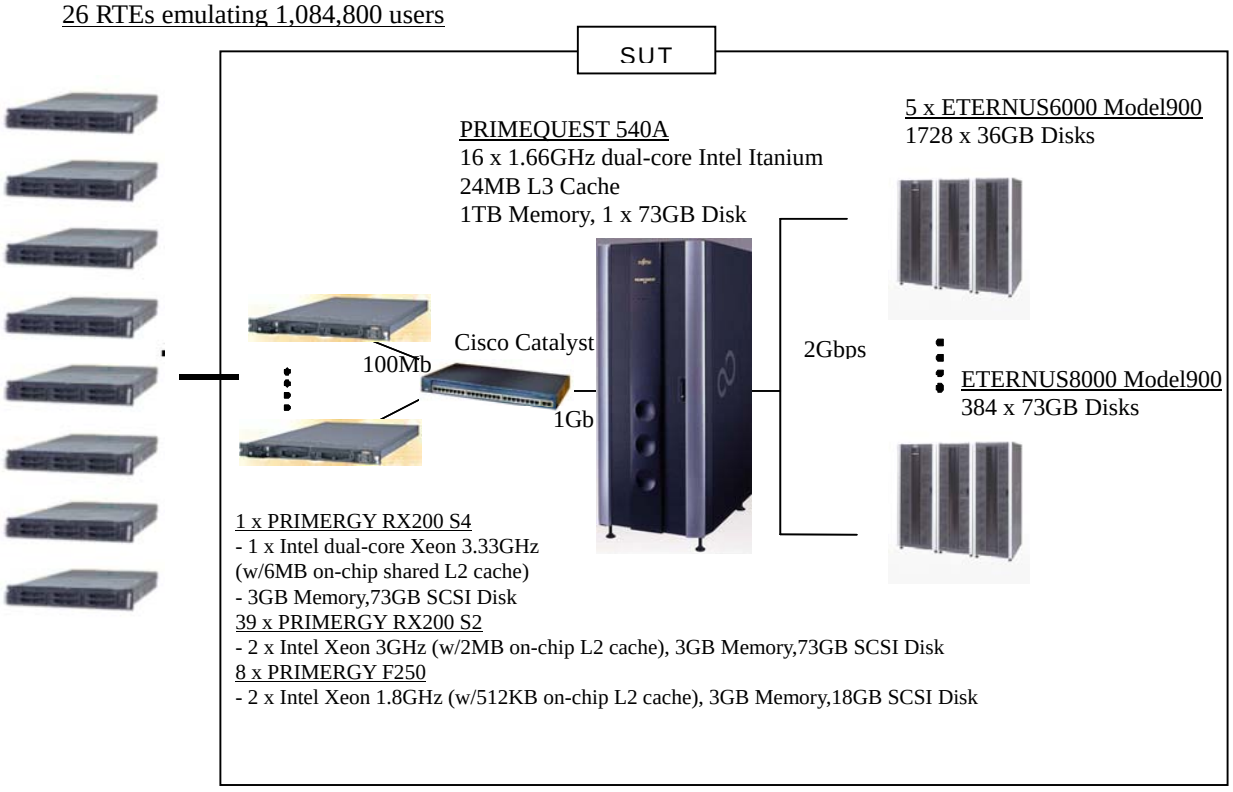
The System Under Test (SUT), a PRIMEQUEST 540A c/s w/48 Front-Ends, is depicted in the following diagrams.

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

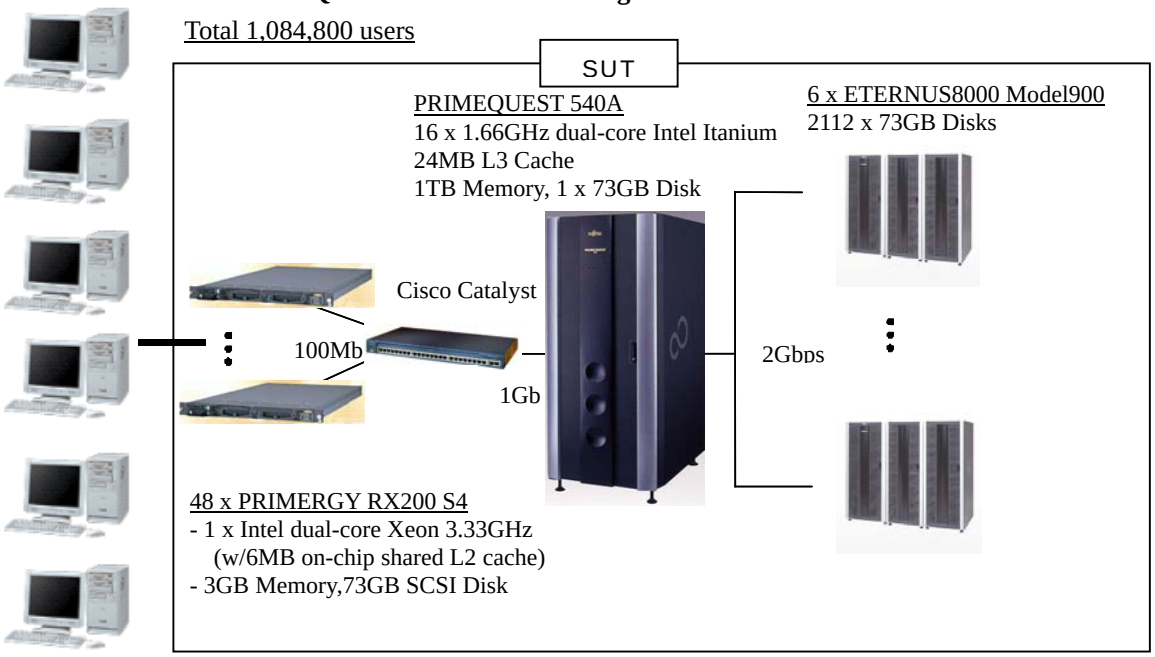
There were differences between the priced and measured configurations. The differences are:

- A RTE was used in the tested configuration.
- The clients other than the PRIMERGY RX200 S4 were replaced by the PRIMERGY RX200 S4 in the priced configuration.
- The ETERNUS6000 Storage subsystems were replaced by the ETERNUS8000 Storage subsystems in the priced configuration.

PRIMEQUEST 540A Tested Configuration



PRIMEQUEST 540A Priced Configuration



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 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.

Partitioning was not used for any of the measurement reported in this full disclosure.

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	59.99%
Order Status	Accessed by last name	60.00%
Delivery	Skipped transactions	None
Transaction Mix	New Order	44.96%
	Payment	43.01%
	Order status	4.01%
	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 108,480 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 1,084,800 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 5 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 108,480 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 1,084,800 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 4 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 108,480 warehouses under a full load of 1,084,800 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 1,084,800 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 ETERNUS8000 storage subsystem and with the Oracle TPC-C tables on 5 ETERNUS6000 storage subsystems.

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 Clasuse 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 112,000 warehouses.

Table 4.1 Number of Rows for Server

Table	Occurrences
Warehouse	112,000
District	1,120,000
Customer	3,360,000,000
History	3,360,000,000
Order	3,360,000,000
New Order	1,008,000,000
Order Line	33,602,101,184
Stock	11,200,000,000
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 5 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 10 partitions (Type A) and the other was configured with 7 partitions (Type B) to lay the TPC-C data files.

Type A:
- 4 partitions for stok
- 5 partitions for cust
- 1 partition for ordr

Type B:
- 1 partition for icust2
- 1 partition for iordr2
- 1 partition for hist
- 1 partition for nord
- 1 partition for temp
- 1 partition for icust1 or istok1
- 1 partition for nord or other tables and indexes

2) Log files

The database logs were configured with 8 RAID0+1 volumes. Each RAID0+1 volume consisted of 24 disks (12 disks + 12 disks mirrored). A log file was configured with 8 LUs (1 LU from each RAID0+1 volume) using Linux mdadm software RAID utility to spread accesses across all 8 volumes.

3) Partition type

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

[Type A : the size of each device is 200GB]
sdb sdd sdf sdh sdj sdl sdn sdp
sdr sdt sdv sdx sdbv sdbx sdbz sdcb
sdcd sdcf sdch sdcj sdcl sdcn sdcp sdcr
sdz sdab sdad sdaf sdah sdaj sdal sdan
sdap sdar sdat sdav sdax sdaz sdbb sdbd
sdbf sdbh sdbj sdbl sdbn sdbp sdbr sdbt
sdhg sdhi sdhk sdhm sdho sdhq sdhs sdhu
sdhw sdhy sdia sdic sdie sdig sdii sdik
sdim sdio sdiq sdis sdiu sdiw sdiy sdja
sdjc sdje sdjg sdji sdjk sdjm sdjo sdjq
sdjs sdju sdjw sdjy sdka sdkc sdke sdkg
sdki sdkk sdkm sdko sdkq sdks sdku sdkw
sdky sdla sdlc sdle sdlg sdli sdlk sdlm
sdlo sdlq sdls sdlu sdlw sdly sdma sdmc

sdme sdmg sdmi sdmk sdmm sdmo sdmq sdms
[Type B : the size of each device is 150GB]
sds sdu sdw sdy sdbw sdby sdca sdcc
sdce sdcg sdci sdck sdcm sdco sdcq sdcs
sdaa sdac sdae sdag sdai sdak sdam sdao
sdaq sdas sdau sdaw sday sdba sdbc sdbe
sdbg sdbi sdbk sdbm sdbo sdbq sdbs sdbu
sdhh sdhj sdhl sdhn sdhp sdhr sdht sdhv
sdhx sdhz sdib sdid sdif sdih sdij sdil
sdin sdip sdir sdit sdiv sdix sdiz sdjb
sdjd sdjf sdjh sdjj sdjl sdjn sdjp sdjr
sdjt sdjv sdjx sdjz sdkb sdkd sdkf sdkh
sdkj sdkl sdkn sdkp sdkr sdkt sdkv sdkx
sdkz sdlb sdld sdlf sdlh sdlj sdll sdln
sdlp sdlr sdlt sdlv sdlx sdlz sdmb sdmd
sdmf sdmh sdmj sdml sdmn sdmp sdmr sdmt

4) Relation between data files and devices

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

datafile name	raw number	dev name	datafile name	raw number	dev name
cust_0_0	raw481	sdb7	iordr2_0_52	raw1325	sdip2
cust_0_1	raw482	sdd7	iordr2_0_53	raw1326	sdir2
cust_0_10	raw491	sdv7	iordr2_0_54	raw1327	sdit2
cust_0_100	raw581	sdlg7	iordr2_0_55	raw1328	sdiv2
cust_0_101	raw582	sdli7	iordr2_0_56	raw1329	sdiz2
cust_0_102	raw583	sdlk7	iordr2_0_57	raw1330	sdjb2
cust_0_103	raw584	sdlm7	iordr2_0_58	raw1331	sdjd2
cust_0_104	raw585	sdlo7	iordr2_0_59	raw1332	sdjf2
cust_0_105	raw586	sdlq7	iordr2_0_6	raw1279	sdq2
cust_0_106	raw587	sdls7	iordr2_0_60	raw1333	sdjj2
cust_0_107	raw588	sdlu7	iordr2_0_61	raw1334	sdjl2
cust_0_108	raw589	sdlw7	iordr2_0_62	raw1335	sdjn2
cust_0_109	raw590	sdly7	iordr2_0_63	raw1336	sdjp2
cust_0_11	raw492	sdx7	iordr2_0_64	raw1337	sdjt2
cust_0_110	raw591	sdma7	iordr2_0_65	raw1338	sdjv2
cust_0_111	raw592	sdmc7	iordr2_0_66	raw1339	sdjx2
cust_0_112	raw593	sdme7	iordr2_0_67	raw1340	sdjz2
cust_0_113	raw594	sdmg7	iordr2_0_68	raw1341	sdkd2
cust_0_114	raw595	sdmi7	iordr2_0_69	raw1342	sdkf2
cust_0_115	raw596	sdmk7	iordr2_0_7	raw1280	sds2
cust_0_116	raw597	sdmm7	iordr2_0_70	raw1343	sdkh2
cust_0_117	raw598	sdmo7	iordr2_0_71	raw1344	sdkj2
cust_0_118	raw599	sdmq7	iordr2_0_72	raw1345	sdkn2
cust_0_119	raw600	sdms7	iordr2_0_73	raw1346	sdkp2
cust_0_12	raw493	sdbv7	iordr2_0_74	raw1347	sdkr2
cust_0_120	raw601	sdb8	iordr2_0_75	raw1348	sdkt2
cust_0_121	raw602	sdd8	iordr2_0_76	raw1349	sdkx2
cust_0_122	raw603	sdf8	iordr2_0_77	raw1350	sdkz2
cust_0_123	raw604	sdh8	iordr2_0_78	raw1351	sdlb2
cust_0_124	raw605	sdj8	iordr2_0_79	raw1352	sdld2
cust_0_125	raw606	sdl8	iordr2_0_8	raw1281	sdw2
cust_0_126	raw607	sdn8	iordr2_0_80	raw1353	sdlh2
cust_0_127	raw608	sdp8	iordr2_0_81	raw1354	sdlj2
cust_0_128	raw609	sdr8	iordr2_0_82	raw1355	sdll2
cust_0_129	raw610	sdt8	iordr2_0_83	raw1356	sdln2
cust_0_13	raw494	sdbx7	iordr2_0_84	raw1357	sdlr2
cust_0_130	raw611	sdv8	iordr2_0_85	raw1358	sdlt2
cust_0_131	raw612	sdx8	iordr2_0_86	raw1359	sdlv2
cust_0_132	raw613	sdbv8	iordr2_0_87	raw1360	sdlx2
cust_0_133	raw614	sdbx8	iordr2_0_88	raw1361	sdmb2
cust_0_134	raw615	sdbz8	iordr2_0_89	raw1362	sdmd2
cust_0_135	raw616	sdcb8	iordr2_0_9	raw1282	sdy2
cust_0_136	raw617	sdcd8	iordr2_0_90	raw1363	sdmf2
cust_0_137	raw618	sdcf8	iordr2_0_91	raw1364	sdmh2
cust_0_138	raw619	sdch8	iordr2_0_92	raw1365	sdml2
cust_0_139	raw620	sdcj8	iordr2_0_93	raw1366	sdmn2
cust_0_14	raw495	sdbz7	iordr2_0_94	raw1367	sdmp2
cust_0_140	raw621	sdcl8	iordr2_0_95	raw1368	sdmr2
cust_0_141	raw622	sdcn8	istok_0_0	raw1561	sdc6
cust_0_142	raw623	sdcp8	istok_0_1	raw1562	sde6
cust_0_143	raw624	sdcr8	istok_0_10	raw1571	sdbw6
cust_0_144	raw625	sdz8	istok_0_11	raw1572	sdby6
cust_0_145	raw626	sdab8	istok_0_12	raw1573	sdcc6
cust_0_146	raw627	sdad8	istok_0_13	raw1574	sdce6

datafile name	raw number	dev name	datafile name	raw number	dev name
cust_0_147	raw628	sdaf8	istok_0_14	raw1575	sdcg6
cust_0_148	raw629	sdah8	istok_0_15	raw1576	sdci6
cust_0_149	raw630	sdaj8	istok_0_16	raw1577	sdcm6
cust_0_15	raw496	sdcb7	istok_0_17	raw1578	sdco6
cust_0_150	raw631	sdal8	istok_0_18	raw1579	sdcq6
cust_0_151	raw632	sdan8	istok_0_19	raw1580	sdcs6
cust_0_152	raw633	sdap8	istok_0_2	raw1563	sdg6
cust_0_153	raw634	sdar8	istok_0_20	raw1581	sdac6
cust_0_154	raw635	sdat8	istok_0_21	raw1582	sdae6
cust_0_155	raw636	sdav8	istok_0_22	raw1583	sdag6
cust_0_156	raw637	sdax8	istok_0_23	raw1584	sdai6
cust_0_157	raw638	sdaz8	istok_0_24	raw1585	sdam6
cust_0_158	raw639	sdbb8	istok_0_25	raw1586	sdao6
cust_0_159	raw640	sdbd8	istok_0_26	raw1587	sdaq6
cust_0_16	raw497	sdcd7	istok_0_27	raw1588	sdas6
cust_0_160	raw641	sdbf8	istok_0_28	raw1589	sdaw6
cust_0_161	raw642	sdbh8	istok_0_29	raw1590	sday6
cust_0_162	raw643	sdbj8	istok_0_3	raw1564	sdi6
cust_0_163	raw644	sdbl8	istok_0_30	raw1591	sdba6
cust_0_164	raw645	sdbn8	istok_0_31	raw1592	sdbc6
cust_0_165	raw646	sdbp8	istok_0_32	raw1593	sdbg6
cust_0_166	raw647	sdbr8	istok_0_33	raw1594	sdbi6
cust_0_167	raw648	sdbt8	istok_0_34	raw1595	sdbk6
cust_0_168	raw649	sdhg8	istok_0_35	raw1596	sdbm6
cust_0_169	raw650	sdhi8	istok_0_36	raw1597	sdbq6
cust_0_17	raw498	sdcf7	istok_0_37	raw1598	sdbs6
cust_0_170	raw651	sdhk8	istok_0_38	raw1599	sdbu6
cust_0_171	raw652	sdhm8	istok_0_39	raw1600	sdhh6
cust_0_172	raw653	sdho8	istok_0_4	raw1565	sdm6
cust_0_173	raw654	sdhq8	istok_0_5	raw1566	sdo6
cust_0_174	raw655	sdhs8	istok_0_6	raw1567	sdq6
cust_0_175	raw656	sdhu8	istok_0_7	raw1568	sds6
cust_0_176	raw657	sdhw8	istok_0_8	raw1569	sdw6
cust_0_177	raw658	sdhy8	istok_0_9	raw1570	sdy6
cust_0_178	raw659	sdia8	item_0_0	raw1755	sdg8
cust_0_179	raw660	sdic8	iware_0_0	raw1756	sdi8
cust_0_18	raw499	sdch7	log_1_1	raw1761	sdw8
cust_0_180	raw661	sdie8	log_1_2	raw1762	sdy8
cust_0_181	raw662	sdig8	nord_0_0	raw1465	sdc5
cust_0_182	raw663	sdii8	nord_0_1	raw1466	sde5
cust_0_183	raw664	sdik8	nord_0_10	raw1475	sdbw5
cust_0_184	raw665	sdim8	nord_0_11	raw1476	sdby5
cust_0_185	raw666	sdio8	nord_0_12	raw1477	sdcc5
cust_0_186	raw667	sdiq8	nord_0_13	raw1478	sdce5
cust_0_187	raw668	sdis8	nord_0_14	raw1479	sdcg5
cust_0_188	raw669	sdiu8	nord_0_15	raw1480	sdci5
cust_0_189	raw670	sdiw8	nord_0_16	raw1481	sdcm5
cust_0_19	raw500	sdcj7	nord_0_17	raw1482	sdco5
cust_0_190	raw671	sdiy8	nord_0_18	raw1483	sdcq5
cust_0_191	raw672	sdja8	nord_0_19	raw1484	sdcs5
cust_0_192	raw673	sdjc8	nord_0_2	raw1467	sdg5
cust_0_193	raw674	sdje8	nord_0_20	raw1485	sdac5
cust_0_194	raw675	sdjg8	nord_0_21	raw1486	sdae5
cust_0_195	raw676	sdji8	nord_0_22	raw1487	sdag5

datafile name	raw number	dev name	datafile name	raw number	dev name
cust_0_196	raw677	sdjk8	nord_0_23	raw1488	sdai5
cust_0_197	raw678	sdjm8	nord_0_24	raw1489	sdam5
cust_0_198	raw679	sdjo8	nord_0_25	raw1490	sdao5
cust_0_199	raw680	sdjq8	nord_0_26	raw1491	sdaq5
cust_0_2	raw483	sdf7	nord_0_27	raw1492	sdas5
cust_0_20	raw501	sdcl7	nord_0_28	raw1493	sdaw5
cust_0_200	raw681	sdjs8	nord_0_29	raw1494	sday5
cust_0_201	raw682	sdju8	nord_0_3	raw1468	sdi5
cust_0_202	raw683	sdjw8	nord_0_30	raw1495	sdba5
cust_0_203	raw684	sdjy8	nord_0_31	raw1496	sdbc5
cust_0_204	raw685	sdka8	nord_0_32	raw1497	sdbg5
cust_0_205	raw686	sdkc8	nord_0_33	raw1498	sdbi5
cust_0_206	raw687	sdke8	nord_0_34	raw1499	sdbk5
cust_0_207	raw688	sdkg8	nord_0_35	raw1500	sdbm5
cust_0_208	raw689	sdki8	nord_0_36	raw1501	sdbq5
cust_0_209	raw690	sdkk8	nord_0_37	raw1502	sdbs5
cust_0_21	raw502	sdcn7	nord_0_38	raw1503	sdbu5
cust_0_210	raw691	sdkm8	nord_0_39	raw1504	sdhh5
cust_0_211	raw692	sdko8	nord_0_4	raw1469	sdm5
cust_0_212	raw693	sdkq8	nord_0_40	raw1505	sdhl5
cust_0_213	raw694	sdks8	nord_0_41	raw1506	sdhn5
cust_0_214	raw695	sdku8	nord_0_42	raw1507	sdhp5
cust_0_215	raw696	sdkw8	nord_0_43	raw1508	sdhr5
cust_0_216	raw697	sdky8	nord_0_44	raw1509	sdhv5
cust_0_217	raw698	sdla8	nord_0_45	raw1510	sdhx5
cust_0_218	raw699	sdlc8	nord_0_46	raw1511	sdhz5
cust_0_219	raw700	sdle8	nord_0_47	raw1512	sdib5
cust_0_22	raw503	sdcp7	nord_0_48	raw1513	sdif5
cust_0_220	raw701	sdlg8	nord_0_49	raw1514	sdih5
cust_0_221	raw702	sdli8	nord_0_5	raw1470	sdo5
cust_0_222	raw703	sdlk8	nord_0_50	raw1515	sdij5
cust_0_223	raw704	sdlm8	nord_0_51	raw1516	sdil5
cust_0_224	raw705	sdlo8	nord_0_52	raw1517	sdip5
cust_0_225	raw706	sdlq8	nord_0_53	raw1518	sdir5
cust_0_226	raw707	sdls8	nord_0_54	raw1519	sdit5
cust_0_227	raw708	sdlu8	nord_0_55	raw1520	sdiv5
cust_0_228	raw709	sdlw8	nord_0_56	raw1521	sdiz5
cust_0_229	raw710	sdly8	nord_0_57	raw1522	sdjb5
cust_0_23	raw504	sdcr7	nord_0_58	raw1523	sdjd5
cust_0_230	raw711	sdma8	nord_0_59	raw1524	sdjf5
cust_0_231	raw712	sdmc8	nord_0_6	raw1471	sdq5
cust_0_232	raw713	sdme8	nord_0_60	raw1525	sdjj5
cust_0_233	raw714	sdmg8	nord_0_61	raw1526	sdjl5
cust_0_234	raw715	sdmi8	nord_0_62	raw1527	sdjn5
cust_0_235	raw716	sdmk8	nord_0_63	raw1528	sdjp5
cust_0_236	raw717	sdmm8	nord_0_64	raw1529	sdjt5
cust_0_237	raw718	sdmo8	nord_0_65	raw1530	sdjv5
cust_0_238	raw719	sdmq8	nord_0_66	raw1531	sdjx5
cust_0_239	raw720	sdms8	nord_0_67	raw1532	sdjz5
cust_0_24	raw505	sdz7	nord_0_68	raw1533	sdkd5
cust_0_240	raw721	sdb9	nord_0_69	raw1534	sdkf5
cust_0_241	raw722	sdd9	nord_0_7	raw1472	sds5
cust_0_242	raw723	sdf9	nord_0_70	raw1535	sdkh5
cust_0_243	raw724	sdh9	nord_0_71	raw1536	sdkj5

datafile name	raw number	dev name	datafile name	raw number	dev name
cust_0_244	raw725	sdj9	nord_0_72	raw1537	sdkn5
cust_0_245	raw726	sdl9	nord_0_73	raw1538	sdkp5
cust_0_246	raw727	sdn9	nord_0_74	raw1539	sdkr5
cust_0_247	raw728	sdp9	nord_0_75	raw1540	sdkt5
cust_0_248	raw729	sdr9	nord_0_76	raw1541	sdkx5
cust_0_249	raw730	sdt9	nord_0_77	raw1542	sdkz5
cust_0_25	raw506	sdab7	nord_0_78	raw1543	sdlb5
cust_0_250	raw731	sdv9	nord_0_79	raw1544	sdld5
cust_0_251	raw732	sdx9	nord_0_8	raw1473	sdw5
cust_0_252	raw733	sdbv9	nord_0_80	raw1545	sdlh5
cust_0_253	raw734	sdbx9	nord_0_81	raw1546	sdlj5
cust_0_254	raw735	sdbz9	nord_0_82	raw1547	sdll5
cust_0_255	raw736	sdcb9	nord_0_83	raw1548	sdln5
cust_0_256	raw737	sdcd9	nord_0_84	raw1549	sdlr5
cust_0_257	raw738	sdcf9	nord_0_85	raw1550	sdlt5
cust_0_258	raw739	sdch9	nord_0_86	raw1551	sdlv5
cust_0_259	raw740	sdcj9	nord_0_87	raw1552	sdlx5
cust_0_26	raw507	sdad7	nord_0_88	raw1553	sdmb5
cust_0_260	raw741	sdcl9	nord_0_89	raw1554	sdmd5
cust_0_261	raw742	sdcn9	nord_0_9	raw1474	sdy5
cust_0_262	raw743	sdcp9	nord_0_90	raw1555	sdmf5
cust_0_263	raw744	sdcr9	nord_0_91	raw1556	sdmh5
cust_0_264	raw745	sdz9	nord_0_92	raw1557	sdml5
cust_0_265	raw746	sdab9	nord_0_93	raw1558	sdmn5
cust_0_266	raw747	sdad9	nord_0_94	raw1559	sdmp5
cust_0_267	raw748	sdaf9	nord_0_95	raw1560	sdmr5
cust_0_268	raw749	sdah9	ordr_0_0	raw1081	sdb13
cust_0_269	raw750	sdaj9	ordr_0_1	raw1082	sdd13
cust_0_27	raw508	sdaf7	ordr_0_10	raw1091	sdbv13
cust_0_270	raw751	sdal9	ordr_0_11	raw1092	sdbx13
cust_0_271	raw752	sdan9	ordr_0_12	raw1093	sdcb13
cust_0_272	raw753	sdap9	ordr_0_13	raw1094	sdcd13
cust_0_273	raw754	sdar9	ordr_0_14	raw1095	sdcf13
cust_0_274	raw755	sdat9	ordr_0_15	raw1096	sdch13
cust_0_275	raw756	sdav9	ordr_0_16	raw1097	sdcl13
cust_0_276	raw757	sdax9	ordr_0_17	raw1098	sdcn13
cust_0_277	raw758	sdaz9	ordr_0_18	raw1099	sdcp13
cust_0_278	raw759	sdbb9	ordr_0_19	raw1100	sdcr13
cust_0_279	raw760	sdbd9	ordr_0_2	raw1083	sdf13
cust_0_28	raw509	sdah7	ordr_0_20	raw1101	sdab13
cust_0_280	raw761	sdbf9	ordr_0_21	raw1102	sdad13
cust_0_281	raw762	sdbh9	ordr_0_22	raw1103	sdaf13
cust_0_282	raw763	sdbj9	ordr_0_23	raw1104	sdah13
cust_0_283	raw764	sdbl9	ordr_0_24	raw1105	sdal13
cust_0_284	raw765	sdbn9	ordr_0_25	raw1106	sdan13
cust_0_285	raw766	sdbp9	ordr_0_26	raw1107	sdap13
cust_0_286	raw767	sdbr9	ordr_0_27	raw1108	sdar13
cust_0_287	raw768	sdbt9	ordr_0_28	raw1109	sdav13
cust_0_288	raw769	sdhg9	ordr_0_29	raw1110	sdax13
cust_0_289	raw770	sdhi9	ordr_0_3	raw1084	sdh13
cust_0_29	raw510	sdaj7	ordr_0_30	raw1111	sdaz13
cust_0_290	raw771	sdhk9	ordr_0_31	raw1112	sdbb13
cust_0_291	raw772	sdhm9	ordr_0_32	raw1113	sdbf13
cust_0_292	raw773	sdho9	ordr_0_33	raw1114	sdbh13

datafile name	raw number	dev name	datafile name	raw number	dev name
cust_0_293	raw774	sdhq9	ordr_0_34	raw1115	sdbj13
cust_0_294	raw775	sdhs9	ordr_0_35	raw1116	sdbl13
cust_0_295	raw776	sdhu9	ordr_0_36	raw1117	sdbp13
cust_0_296	raw777	sdhw9	ordr_0_37	raw1118	sdbr13
cust_0_297	raw778	sdhy9	ordr_0_38	raw1119	sdbt13
cust_0_298	raw779	sdia9	ordr_0_39	raw1120	sdhg13
cust_0_299	raw780	sdic9	ordr_0_4	raw1085	sdl13
cust_0_3	raw484	sdh7	ordr_0_40	raw1121	sdhk13
cust_0_30	raw511	sdal7	ordr_0_41	raw1122	sdhm13
cust_0_300	raw781	sdie9	ordr_0_42	raw1123	sdho13
cust_0_301	raw782	sdig9	ordr_0_43	raw1124	sdhq13
cust_0_302	raw783	sdii9	ordr_0_44	raw1125	sdhu13
cust_0_303	raw784	sdik9	ordr_0_45	raw1126	sdhw13
cust_0_304	raw785	sdim9	ordr_0_46	raw1127	sdhy13
cust_0_305	raw786	sdio9	ordr_0_47	raw1128	sdia13
cust_0_306	raw787	sdiq9	ordr_0_48	raw1129	sdie13
cust_0_307	raw788	sdis9	ordr_0_49	raw1130	sdig13
cust_0_308	raw789	sdiu9	ordr_0_5	raw1086	sdn13
cust_0_309	raw790	sdiw9	ordr_0_50	raw1131	sdii13
cust_0_31	raw512	sdan7	ordr_0_51	raw1132	sdik13
cust_0_310	raw791	sdiy9	ordr_0_52	raw1133	sdio13
cust_0_311	raw792	sdja9	ordr_0_53	raw1134	sdiq13
cust_0_312	raw793	sdjc9	ordr_0_54	raw1135	sdis13
cust_0_313	raw794	sdje9	ordr_0_55	raw1136	sdiu13
cust_0_314	raw795	sdjg9	ordr_0_56	raw1137	sdiy13
cust_0_315	raw796	sdji9	ordr_0_57	raw1138	sdja13
cust_0_316	raw797	sdjk9	ordr_0_58	raw1139	sdjc13
cust_0_317	raw798	sdjm9	ordr_0_59	raw1140	sdje13
cust_0_318	raw799	sdjo9	ordr_0_6	raw1087	sdp13
cust_0_319	raw800	sdjq9	ordr_0_60	raw1141	sdji13
cust_0_32	raw513	sdap7	ordr_0_61	raw1142	sdjk13
cust_0_320	raw801	sdjs9	ordr_0_62	raw1143	sdjm13
cust_0_321	raw802	sdju9	ordr_0_63	raw1144	sdjo13
cust_0_322	raw803	sdjw9	ordr_0_64	raw1145	sdjs13
cust_0_323	raw804	sdjy9	ordr_0_65	raw1146	sdju13
cust_0_324	raw805	sdka9	ordr_0_66	raw1147	sdjw13
cust_0_325	raw806	sdkc9	ordr_0_67	raw1148	sdjy13
cust_0_326	raw807	sdke9	ordr_0_68	raw1149	sdkc13
cust_0_327	raw808	sdkg9	ordr_0_69	raw1150	sdke13
cust_0_328	raw809	sdki9	ordr_0_7	raw1088	sdr13
cust_0_329	raw810	sdkk9	ordr_0_70	raw1151	sdkg13
cust_0_33	raw514	sdar7	ordr_0_71	raw1152	sdki13
cust_0_330	raw811	sdkm9	ordr_0_72	raw1153	sdkm13
cust_0_331	raw812	sdko9	ordr_0_73	raw1154	sdko13
cust_0_332	raw813	sdkq9	ordr_0_74	raw1155	sdkq13
cust_0_333	raw814	sdks9	ordr_0_75	raw1156	sdks13
cust_0_334	raw815	sdku9	ordr_0_76	raw1157	sdkw13
cust_0_335	raw816	sdkw9	ordr_0_77	raw1158	sdky13
cust_0_336	raw817	sdky9	ordr_0_78	raw1159	sdla13
cust_0_337	raw818	sdla9	ordr_0_79	raw1160	sdlc13
cust_0_338	raw819	sdlc9	ordr_0_8	raw1089	sdv13
cust_0_339	raw820	sdle9	ordr_0_80	raw1161	sdlg13
cust_0_34	raw515	sdat7	ordr_0_81	raw1162	sdli13
cust_0_340	raw821	sdlg9	ordr_0_82	raw1163	sdlk13

datafile name	raw number	dev name	datafile name	raw number	dev name
cust_0_341	raw822	sdli9	ordr_0_83	raw1164	sdlm13
cust_0_342	raw823	sdlk9	ordr_0_84	raw1165	sdlq13
cust_0_343	raw824	sdlm9	ordr_0_85	raw1166	sdls13
cust_0_344	raw825	sdlo9	ordr_0_86	raw1167	sdlu13
cust_0_345	raw826	sdlq9	ordr_0_87	raw1168	sdlw13
cust_0_346	raw827	sdls9	ordr_0_88	raw1169	sdma13
cust_0_347	raw828	sdlu9	ordr_0_89	raw1170	sdmc13
cust_0_348	raw829	sdlw9	ordr_0_9	raw1090	sdx13
cust_0_349	raw830	sdly9	ordr_0_90	raw1171	sdme13
cust_0_35	raw516	sdav7	ordr_0_91	raw1172	sdmg13
cust_0_350	raw831	sdma9	ordr_0_92	raw1173	sdmk13
cust_0_351	raw832	sdmc9	ordr_0_93	raw1174	sdmm13
cust_0_352	raw833	sdme9	ordr_0_94	raw1175	sdmo13
cust_0_353	raw834	sdmg9	ordr_0_95	raw1176	sdmq13
cust_0_354	raw835	sdmi9	roll1	raw1753	sdc8
cust_0_355	raw836	sdmk9	sp_0	raw1764	sdby8
cust_0_356	raw837	sdmm9	stok_0_0	raw1	sdb1
cust_0_357	raw838	sdmo9	stok_0_1	raw2	sdd1
cust_0_358	raw839	sdmq9	stok_0_10	raw11	sdv1
cust_0_359	raw840	sdms9	stok_0_100	raw101	sdlg1
cust_0_36	raw517	sdax7	stok_0_101	raw102	sdli1
cust_0_360	raw841	sdb10	stok_0_102	raw103	sdlk1
cust_0_361	raw842	sdd10	stok_0_103	raw104	sdlm1
cust_0_362	raw843	sdf10	stok_0_104	raw105	sdlo1
cust_0_363	raw844	sdh10	stok_0_105	raw106	sdlq1
cust_0_364	raw845	sdj10	stok_0_106	raw107	sdls1
cust_0_365	raw846	sdl10	stok_0_107	raw108	sdlu1
cust_0_366	raw847	sdn10	stok_0_108	raw109	sdlw1
cust_0_367	raw848	sdp10	stok_0_109	raw110	sdly1
cust_0_368	raw849	sdr10	stok_0_11	raw12	sdx1
cust_0_369	raw850	sdt10	stok_0_110	raw111	sdma1
cust_0_37	raw518	sdaz7	stok_0_111	raw112	sdmc1
cust_0_370	raw851	sdv10	stok_0_112	raw113	sdme1
cust_0_371	raw852	sdx10	stok_0_113	raw114	sdmg1
cust_0_372	raw853	sdbv10	stok_0_114	raw115	sdmi1
cust_0_373	raw854	sdbx10	stok_0_115	raw116	sdmk1
cust_0_374	raw855	sdbz10	stok_0_116	raw117	sdmm1
cust_0_375	raw856	sdcb10	stok_0_117	raw118	sdmo1
cust_0_376	raw857	sdcd10	stok_0_118	raw119	sdmq1
cust_0_377	raw858	sdcf10	stok_0_119	raw120	sdms1
cust_0_378	raw859	sdch10	stok_0_12	raw13	sdbv1
cust_0_379	raw860	sdcj10	stok_0_120	raw121	sdb2
cust_0_38	raw519	sdbb7	stok_0_121	raw122	sdd2
cust_0_380	raw861	sdcl10	stok_0_122	raw123	sdf2
cust_0_381	raw862	sdcn10	stok_0_123	raw124	sdh2
cust_0_382	raw863	sdcp10	stok_0_124	raw125	sdj2
cust_0_383	raw864	sdcr10	stok_0_125	raw126	sdl2
cust_0_384	raw865	sdz10	stok_0_126	raw127	sdn2
cust_0_385	raw866	sdab10	stok_0_127	raw128	sdp2
cust_0_386	raw867	sdad10	stok_0_128	raw129	sdr2
cust_0_387	raw868	sdaf10	stok_0_129	raw130	sdt2
cust_0_388	raw869	sdah10	stok_0_13	raw14	sdbx1
cust_0_389	raw870	sdaj10	stok_0_130	raw131	sdv2
cust_0_39	raw520	sdbd7	stok_0_131	raw132	sdx2

datafile name	raw number	dev name	datafile name	raw number	dev name
cust_0_390	raw871	sdal10	stok_0_132	raw133	sdbv2
cust_0_391	raw872	sdan10	stok_0_133	raw134	sdbx2
cust_0_392	raw873	sdap10	stok_0_134	raw135	sdbz2
cust_0_393	raw874	sdar10	stok_0_135	raw136	sdcb2
cust_0_394	raw875	sdat10	stok_0_136	raw137	sdcd2
cust_0_395	raw876	sdav10	stok_0_137	raw138	sdcf2
cust_0_396	raw877	sdax10	stok_0_138	raw139	sdch2
cust_0_397	raw878	sdaz10	stok_0_139	raw140	sdcj2
cust_0_398	raw879	sdbb10	stok_0_14	raw15	sdbz1
cust_0_399	raw880	sdbd10	stok_0_140	raw141	sdcl2
cust_0_4	raw485	sdj7	stok_0_141	raw142	sdcn2
cust_0_40	raw521	sdbf7	stok_0_142	raw143	sdcp2
cust_0_400	raw881	sdbf10	stok_0_143	raw144	sdcr2
cust_0_401	raw882	sdbh10	stok_0_144	raw145	sdz2
cust_0_402	raw883	sdbj10	stok_0_145	raw146	sdab2
cust_0_403	raw884	sdbl10	stok_0_146	raw147	sdad2
cust_0_404	raw885	sdbn10	stok_0_147	raw148	sdaf2
cust_0_405	raw886	sdbp10	stok_0_148	raw149	sdah2
cust_0_406	raw887	sdbr10	stok_0_149	raw150	sdaj2
cust_0_407	raw888	sdbt10	stok_0_15	raw16	sdcb1
cust_0_408	raw889	sdhg10	stok_0_150	raw151	sdal2
cust_0_409	raw890	sdhi10	stok_0_151	raw152	sdan2
cust_0_41	raw522	sdbh7	stok_0_152	raw153	sdap2
cust_0_410	raw891	sdhk10	stok_0_153	raw154	sdar2
cust_0_411	raw892	sdhm10	stok_0_154	raw155	sdat2
cust_0_412	raw893	sdho10	stok_0_155	raw156	sdav2
cust_0_413	raw894	sdhq10	stok_0_156	raw157	sdax2
cust_0_414	raw895	sdhs10	stok_0_157	raw158	sdaz2
cust_0_415	raw896	sdhu10	stok_0_158	raw159	sdbb2
cust_0_416	raw897	sdhw10	stok_0_159	raw160	sdbd2
cust_0_417	raw898	sdhy10	stok_0_16	raw17	sdcd1
cust_0_418	raw899	sdia10	stok_0_160	raw161	sdbf2
cust_0_419	raw900	sdic10	stok_0_161	raw162	sdbh2
cust_0_42	raw523	sdbj7	stok_0_162	raw163	sdbj2
cust_0_420	raw901	sdie10	stok_0_163	raw164	sdbl2
cust_0_421	raw902	sdig10	stok_0_164	raw165	sdbn2
cust_0_422	raw903	sdii10	stok_0_165	raw166	sdbp2
cust_0_423	raw904	sdik10	stok_0_166	raw167	sdbr2
cust_0_424	raw905	sdim10	stok_0_167	raw168	sdbt2
cust_0_425	raw906	sdio10	stok_0_168	raw169	sdhg2
cust_0_426	raw907	sdiq10	stok_0_169	raw170	sdhi2
cust_0_427	raw908	sdis10	stok_0_17	raw18	sdcf1
cust_0_428	raw909	sdiu10	stok_0_170	raw171	sdhk2
cust_0_429	raw910	sdiw10	stok_0_171	raw172	sdhm2
cust_0_43	raw524	sdbl7	stok_0_172	raw173	sdho2
cust_0_430	raw911	sdiy10	stok_0_173	raw174	sdhq2
cust_0_431	raw912	sdja10	stok_0_174	raw175	sdhs2
cust_0_432	raw913	sdjc10	stok_0_175	raw176	sdhu2
cust_0_433	raw914	sdje10	stok_0_176	raw177	sdhw2
cust_0_434	raw915	sdjg10	stok_0_177	raw178	sdhy2
cust_0_435	raw916	sdji10	stok_0_178	raw179	sdia2
cust_0_436	raw917	sdjk10	stok_0_179	raw180	sdic2
cust_0_437	raw918	sdjm10	stok_0_18	raw19	sdch1
cust_0_438	raw919	sdjo10	stok_0_180	raw181	sdie2

datafile name	raw number	dev name	datafile name	raw number	dev name
cust_0_439	raw920	sdjq10	stok_0_181	raw182	sdig2
cust_0_44	raw525	sdbn7	stok_0_182	raw183	sdii2
cust_0_440	raw921	sdjs10	stok_0_183	raw184	sdik2
cust_0_441	raw922	sdju10	stok_0_184	raw185	sdim2
cust_0_442	raw923	sdjw10	stok_0_185	raw186	sdio2
cust_0_443	raw924	sdjy10	stok_0_186	raw187	sdiq2
cust_0_444	raw925	sdka10	stok_0_187	raw188	sdis2
cust_0_445	raw926	sdkc10	stok_0_188	raw189	sdiu2
cust_0_446	raw927	sdke10	stok_0_189	raw190	sdiw2
cust_0_447	raw928	sdkg10	stok_0_19	raw20	sdcj1
cust_0_448	raw929	sdki10	stok_0_190	raw191	sdiy2
cust_0_449	raw930	sdkk10	stok_0_191	raw192	sdja2
cust_0_45	raw526	sdbp7	stok_0_192	raw193	sdjc2
cust_0_450	raw931	sdkm10	stok_0_193	raw194	sdje2
cust_0_451	raw932	sdko10	stok_0_194	raw195	sdjg2
cust_0_452	raw933	sdkq10	stok_0_195	raw196	sdji2
cust_0_453	raw934	sdks10	stok_0_196	raw197	sdjk2
cust_0_454	raw935	sdku10	stok_0_197	raw198	sdjm2
cust_0_455	raw936	sdkw10	stok_0_198	raw199	sdjo2
cust_0_456	raw937	sdky10	stok_0_199	raw200	sdjq2
cust_0_457	raw938	sdla10	stok_0_2	raw3	sdf1
cust_0_458	raw939	sdlc10	stok_0_20	raw21	sdcl1
cust_0_459	raw940	sdle10	stok_0_200	raw201	sdjs2
cust_0_46	raw527	sdbr7	stok_0_201	raw202	sdju2
cust_0_460	raw941	sdlg10	stok_0_202	raw203	sdjw2
cust_0_461	raw942	sdli10	stok_0_203	raw204	sdjy2
cust_0_462	raw943	sdlk10	stok_0_204	raw205	sdka2
cust_0_463	raw944	sdlm10	stok_0_205	raw206	sdkc2
cust_0_464	raw945	sdlo10	stok_0_206	raw207	sdke2
cust_0_465	raw946	sdlq10	stok_0_207	raw208	sdkg2
cust_0_466	raw947	sdls10	stok_0_208	raw209	sdki2
cust_0_467	raw948	sdlu10	stok_0_209	raw210	sdkk2
cust_0_468	raw949	sdlw10	stok_0_21	raw22	sdcn1
cust_0_469	raw950	sdly10	stok_0_210	raw211	sdkm2
cust_0_47	raw528	sdbt7	stok_0_211	raw212	sdko2
cust_0_470	raw951	sdma10	stok_0_212	raw213	sdkq2
cust_0_471	raw952	sdmc10	stok_0_213	raw214	sdks2
cust_0_472	raw953	sdme10	stok_0_214	raw215	sdku2
cust_0_473	raw954	sdmg10	stok_0_215	raw216	sdkw2
cust_0_474	raw955	sdmi10	stok_0_216	raw217	sdky2
cust_0_475	raw956	sdmk10	stok_0_217	raw218	sdla2
cust_0_476	raw957	sdmm10	stok_0_218	raw219	sdlc2
cust_0_477	raw958	sdmo10	stok_0_219	raw220	sdle2
cust_0_478	raw959	sdmq10	stok_0_22	raw23	sdcp1
cust_0_479	raw960	sdms10	stok_0_220	raw221	sdlg2
cust_0_48	raw529	sdhg7	stok_0_221	raw222	sdli2
cust_0_480	raw961	sdb11	stok_0_222	raw223	sdlk2
cust_0_481	raw962	sdd11	stok_0_223	raw224	sdlm2
cust_0_482	raw963	sdf11	stok_0_224	raw225	sdlo2
cust_0_483	raw964	sdh11	stok_0_225	raw226	sdlq2
cust_0_484	raw965	sdj11	stok_0_226	raw227	sdls2
cust_0_485	raw966	sdl11	stok_0_227	raw228	sdlu2
cust_0_486	raw967	sdn11	stok_0_228	raw229	sdlw2
cust_0_487	raw968	sdp11	stok_0_229	raw230	sdly2

datafile name	raw number	dev name	datafile name	raw number	dev name
cust_0_488	raw969	sdr11	stok_0_23	raw24	sdcr1
cust_0_489	raw970	sdt11	stok_0_230	raw231	sdma2
cust_0_49	raw530	sdhi7	stok_0_231	raw232	sdmc2
cust_0_490	raw971	sdv11	stok_0_232	raw233	sdme2
cust_0_491	raw972	sdx11	stok_0_233	raw234	sdmg2
cust_0_492	raw973	sdbv11	stok_0_234	raw235	sdmi2
cust_0_493	raw974	sdbx11	stok_0_235	raw236	sdmk2
cust_0_494	raw975	sdbz11	stok_0_236	raw237	sdmm2
cust_0_495	raw976	sdcb11	stok_0_237	raw238	sdmo2
cust_0_496	raw977	sdcd11	stok_0_238	raw239	sdmq2
cust_0_497	raw978	sdcf11	stok_0_239	raw240	sdms2
cust_0_498	raw979	sdch11	stok_0_24	raw25	sdz1
cust_0_499	raw980	sdcj11	stok_0_240	raw241	sdb3
cust_0_5	raw486	sdl7	stok_0_241	raw242	sdd3
cust_0_50	raw531	sdhk7	stok_0_242	raw243	sdf3
cust_0_500	raw981	sdcl11	stok_0_243	raw244	sdh3
cust_0_501	raw982	sdcn11	stok_0_244	raw245	sdj3
cust_0_502	raw983	sdcp11	stok_0_245	raw246	sdl3
cust_0_503	raw984	sdcr11	stok_0_246	raw247	sdn3
cust_0_504	raw985	sdz11	stok_0_247	raw248	sdp3
cust_0_505	raw986	sdab11	stok_0_248	raw249	sdr3
cust_0_506	raw987	sdad11	stok_0_249	raw250	sdt3
cust_0_507	raw988	sdaf11	stok_0_25	raw26	sdab1
cust_0_508	raw989	sdah11	stok_0_250	raw251	sdv3
cust_0_509	raw990	sdaj11	stok_0_251	raw252	sdx3
cust_0_51	raw532	sdhm7	stok_0_252	raw253	sdbv3
cust_0_510	raw991	sdal11	stok_0_253	raw254	sdbx3
cust_0_511	raw992	sdan11	stok_0_254	raw255	sdbz3
cust_0_512	raw993	sdap11	stok_0_255	raw256	sdcb3
cust_0_513	raw994	sdar11	stok_0_256	raw257	sdcd3
cust_0_514	raw995	sdat11	stok_0_257	raw258	sdcf3
cust_0_515	raw996	sdav11	stok_0_258	raw259	sdch3
cust_0_516	raw997	sdax11	stok_0_259	raw260	sdcj3
cust_0_517	raw998	sdaz11	stok_0_26	raw27	sdad1
cust_0_518	raw999	sdbb11	stok_0_260	raw261	sdcl3
cust_0_519	raw1000	sdbd11	stok_0_261	raw262	sdcn3
cust_0_52	raw533	sdho7	stok_0_262	raw263	sdcp3
cust_0_520	raw1001	sdbf11	stok_0_263	raw264	sdcr3
cust_0_521	raw1002	sdbh11	stok_0_264	raw265	sdz3
cust_0_522	raw1003	sdbj11	stok_0_265	raw266	sdab3
cust_0_523	raw1004	sdbl11	stok_0_266	raw267	sdad3
cust_0_524	raw1005	sdbn11	stok_0_267	raw268	sdaf3
cust_0_525	raw1006	sdbp11	stok_0_268	raw269	sdah3
cust_0_526	raw1007	sdbr11	stok_0_269	raw270	sdaj3
cust_0_527	raw1008	sdbt11	stok_0_27	raw28	sdaf1
cust_0_528	raw1009	sdhg11	stok_0_270	raw271	sdal3
cust_0_529	raw1010	sdhi11	stok_0_271	raw272	sdan3
cust_0_53	raw534	sdhq7	stok_0_272	raw273	sdap3
cust_0_530	raw1011	sdhk11	stok_0_273	raw274	sdar3
cust_0_531	raw1012	sdhm11	stok_0_274	raw275	sdat3
cust_0_532	raw1013	sdho11	stok_0_275	raw276	sdav3
cust_0_533	raw1014	sdhq11	stok_0_276	raw277	sdax3
cust_0_534	raw1015	sdhs11	stok_0_277	raw278	sdaz3
cust_0_535	raw1016	sdhu11	stok_0_278	raw279	sdbb3

datafile name	raw number	dev name	datafile name	raw number	dev name
cust_0_536	raw1017	sdhw11	stok_0_279	raw280	sdbd3
cust_0_537	raw1018	sdhy11	stok_0_28	raw29	sdah1
cust_0_538	raw1019	sdia11	stok_0_280	raw281	sdbf3
cust_0_539	raw1020	sdic11	stok_0_281	raw282	sdbh3
cust_0_54	raw535	sdhs7	stok_0_282	raw283	sdbj3
cust_0_540	raw1021	sdie11	stok_0_283	raw284	sdbl3
cust_0_541	raw1022	sdig11	stok_0_284	raw285	sdbn3
cust_0_542	raw1023	sdii11	stok_0_285	raw286	sdbp3
cust_0_543	raw1024	sdik11	stok_0_286	raw287	sdbr3
cust_0_544	raw1025	sdim11	stok_0_287	raw288	sdbt3
cust_0_545	raw1026	sdio11	stok_0_288	raw289	sdhg3
cust_0_546	raw1027	sdiq11	stok_0_289	raw290	sdhi3
cust_0_547	raw1028	sdis11	stok_0_29	raw30	sdaj1
cust_0_548	raw1029	sdiu11	stok_0_290	raw291	sdhk3
cust_0_549	raw1030	sdiw11	stok_0_291	raw292	sdhm3
cust_0_55	raw536	sdhu7	stok_0_292	raw293	sdho3
cust_0_550	raw1031	sdiy11	stok_0_293	raw294	sdhq3
cust_0_551	raw1032	sdja11	stok_0_294	raw295	sdhs3
cust_0_552	raw1033	sdjc11	stok_0_295	raw296	sdhu3
cust_0_553	raw1034	sdje11	stok_0_296	raw297	sdhw3
cust_0_554	raw1035	sdjg11	stok_0_297	raw298	sdhy3
cust_0_555	raw1036	sdji11	stok_0_298	raw299	sdia3
cust_0_556	raw1037	sdjk11	stok_0_299	raw300	sdic3
cust_0_557	raw1038	sdjm11	stok_0_3	raw4	sdh1
cust_0_558	raw1039	sdjo11	stok_0_30	raw31	sdal1
cust_0_559	raw1040	sdjq11	stok_0_300	raw301	sdie3
cust_0_56	raw537	sdhw7	stok_0_301	raw302	sdig3
cust_0_560	raw1041	sdjs11	stok_0_302	raw303	sdii3
cust_0_561	raw1042	sdju11	stok_0_303	raw304	sdik3
cust_0_562	raw1043	sdjw11	stok_0_304	raw305	sdim3
cust_0_563	raw1044	sdjy11	stok_0_305	raw306	sdio3
cust_0_564	raw1045	sdka11	stok_0_306	raw307	sdiq3
cust_0_565	raw1046	sdkc11	stok_0_307	raw308	sdis3
cust_0_566	raw1047	sdke11	stok_0_308	raw309	sdiu3
cust_0_567	raw1048	sdkg11	stok_0_309	raw310	sdiw3
cust_0_568	raw1049	sdki11	stok_0_31	raw32	sdan1
cust_0_569	raw1050	sdkk11	stok_0_310	raw311	sdiy3
cust_0_57	raw538	sdhy7	stok_0_311	raw312	sdja3
cust_0_570	raw1051	sdkm11	stok_0_312	raw313	sdjc3
cust_0_571	raw1052	sdko11	stok_0_313	raw314	sdje3
cust_0_572	raw1053	sdkq11	stok_0_314	raw315	sdjg3
cust_0_573	raw1054	sdks11	stok_0_315	raw316	sdji3
cust_0_574	raw1055	sdku11	stok_0_316	raw317	sdjk3
cust_0_575	raw1056	sdkw11	stok_0_317	raw318	sdjm3
cust_0_58	raw539	sdia7	stok_0_318	raw319	sdjo3
cust_0_59	raw540	sdic7	stok_0_319	raw320	sdjq3
cust_0_6	raw487	sdn7	stok_0_32	raw33	sdap1
cust_0_60	raw541	sdie7	stok_0_320	raw321	sdjs3
cust_0_61	raw542	sdig7	stok_0_321	raw322	sdju3
cust_0_62	raw543	sdii7	stok_0_322	raw323	sdjw3
cust_0_63	raw544	sdik7	stok_0_323	raw324	sdjy3
cust_0_64	raw545	sdim7	stok_0_324	raw325	sdka3
cust_0_65	raw546	sdio7	stok_0_325	raw326	sdkc3
cust_0_66	raw547	sdiq7	stok_0_326	raw327	sdke3

datafile name	raw number	dev name	datafile name	raw number	dev name
cust_0_67	raw548	sdis7	stok_0_327	raw328	sdkg3
cust_0_68	raw549	sdiu7	stok_0_328	raw329	sdki3
cust_0_69	raw550	sdiw7	stok_0_329	raw330	sdkk3
cust_0_7	raw488	sdp7	stok_0_33	raw34	sdar1
cust_0_70	raw551	sdiy7	stok_0_330	raw331	sdkm3
cust_0_71	raw552	sdja7	stok_0_331	raw332	sdko3
cust_0_72	raw553	sdjc7	stok_0_332	raw333	sdkq3
cust_0_73	raw554	sdje7	stok_0_333	raw334	sdks3
cust_0_74	raw555	sdjg7	stok_0_334	raw335	sdku3
cust_0_75	raw556	sdji7	stok_0_335	raw336	sdkw3
cust_0_76	raw557	sdjk7	stok_0_336	raw337	sdky3
cust_0_77	raw558	sdjm7	stok_0_337	raw338	sdla3
cust_0_78	raw559	sdjo7	stok_0_338	raw339	sdlc3
cust_0_79	raw560	sdjq7	stok_0_339	raw340	sdle3
cust_0_8	raw489	sdr7	stok_0_34	raw35	sdat1
cust_0_80	raw561	sdjs7	stok_0_340	raw341	sdlg3
cust_0_81	raw562	sdju7	stok_0_341	raw342	sdli3
cust_0_82	raw563	sdjw7	stok_0_342	raw343	sdlk3
cust_0_83	raw564	sdjy7	stok_0_343	raw344	sdlm3
cust_0_84	raw565	sdka7	stok_0_344	raw345	sdlo3
cust_0_85	raw566	sdkc7	stok_0_345	raw346	sdlq3
cust_0_86	raw567	sdke7	stok_0_346	raw347	sdls3
cust_0_87	raw568	sdkg7	stok_0_347	raw348	sdlu3
cust_0_88	raw569	sdki7	stok_0_348	raw349	sdlw3
cust_0_89	raw570	sdkk7	stok_0_349	raw350	sdly3
cust_0_9	raw490	sdt7	stok_0_35	raw36	sdav1
cust_0_90	raw571	sdkm7	stok_0_350	raw351	sdma3
cust_0_91	raw572	sdko7	stok_0_351	raw352	sdmc3
cust_0_92	raw573	sdkq7	stok_0_352	raw353	sdme3
cust_0_93	raw574	sdks7	stok_0_353	raw354	sdmg3
cust_0_94	raw575	sdku7	stok_0_354	raw355	sdmi3
cust_0_95	raw576	sdkw7	stok_0_355	raw356	sdmk3
cust_0_96	raw577	sdky7	stok_0_356	raw357	sdmm3
cust_0_97	raw578	sdla7	stok_0_357	raw358	sdmo3
cust_0_98	raw579	sdlc7	stok_0_358	raw359	sdmq3
cust_0_99	raw580	sdle7	stok_0_359	raw360	sdms3
dist_0_0	raw1763	sdbw8	stok_0_36	raw37	sdax1
hist_0_0	raw1369	sdc3	stok_0_360	raw361	sdb5
hist_0_1	raw1370	sde3	stok_0_361	raw362	sdd5
hist_0_10	raw1379	sdbw3	stok_0_362	raw363	sdf5
hist_0_11	raw1380	sdby3	stok_0_363	raw364	sdh5
hist_0_12	raw1381	sdcc3	stok_0_364	raw365	sdj5
hist_0_13	raw1382	sdce3	stok_0_365	raw366	sdl5
hist_0_14	raw1383	sdcg3	stok_0_366	raw367	sdn5
hist_0_15	raw1384	sdci3	stok_0_367	raw368	sdp5
hist_0_16	raw1385	sdcm3	stok_0_368	raw369	sdr5
hist_0_17	raw1386	sdco3	stok_0_369	raw370	sdt5
hist_0_18	raw1387	sdcq3	stok_0_37	raw38	sdaz1
hist_0_19	raw1388	sdcs3	stok_0_370	raw371	sdv5
hist_0_2	raw1371	sdg3	stok_0_371	raw372	sdx5
hist_0_20	raw1389	sdac3	stok_0_372	raw373	sdbv5
hist_0_21	raw1390	sdae3	stok_0_373	raw374	sdbx5
hist_0_22	raw1391	sdag3	stok_0_374	raw375	sdbz5
hist_0_23	raw1392	sdai3	stok_0_375	raw376	sdcb5

datafile name	raw number	dev name	datafile name	raw number	dev name
hist_0_24	raw1393	sdam3	stok_0_376	raw377	sdcd5
hist_0_25	raw1394	sdao3	stok_0_377	raw378	sdcf5
hist_0_26	raw1395	sdaq3	stok_0_378	raw379	sdch5
hist_0_27	raw1396	sdas3	stok_0_379	raw380	sdcj5
hist_0_28	raw1397	sdaw3	stok_0_38	raw39	sdbb1
hist_0_29	raw1398	sday3	stok_0_380	raw381	sdcl5
hist_0_3	raw1372	sdi3	stok_0_381	raw382	sdcn5
hist_0_30	raw1399	sdba3	stok_0_382	raw383	sdcp5
hist_0_31	raw1400	sdbc3	stok_0_383	raw384	sdcr5
hist_0_32	raw1401	sdbg3	stok_0_384	raw385	sdz5
hist_0_33	raw1402	sdbi3	stok_0_385	raw386	sdab5
hist_0_34	raw1403	sdbk3	stok_0_386	raw387	sdad5
hist_0_35	raw1404	sdbm3	stok_0_387	raw388	sdaf5
hist_0_36	raw1405	sdbq3	stok_0_388	raw389	sdah5
hist_0_37	raw1406	sdbs3	stok_0_389	raw390	sdaj5
hist_0_38	raw1407	sdbu3	stok_0_39	raw40	sdbd1
hist_0_39	raw1408	sdhh3	stok_0_390	raw391	sdal5
hist_0_4	raw1373	sdm3	stok_0_391	raw392	sdan5
hist_0_40	raw1409	sdhl3	stok_0_392	raw393	sdap5
hist_0_41	raw1410	sdhn3	stok_0_393	raw394	sdar5
hist_0_42	raw1411	sdhp3	stok_0_394	raw395	sdat5
hist_0_43	raw1412	sdhr3	stok_0_395	raw396	sdav5
hist_0_44	raw1413	sdhv3	stok_0_396	raw397	sdax5
hist_0_45	raw1414	sdhx3	stok_0_397	raw398	sdaz5
hist_0_46	raw1415	sdhz3	stok_0_398	raw399	sdbb5
hist_0_47	raw1416	sdib3	stok_0_399	raw400	sdbd5
hist_0_48	raw1417	sdif3	stok_0_4	raw5	sdj1
hist_0_49	raw1418	sdih3	stok_0_40	raw41	sdbf1
hist_0_5	raw1374	sdo3	stok_0_400	raw401	sdbf5
hist_0_50	raw1419	sdij3	stok_0_401	raw402	sdbh5
hist_0_51	raw1420	sdil3	stok_0_402	raw403	sdbj5
hist_0_52	raw1421	sdip3	stok_0_403	raw404	sdbl5
hist_0_53	raw1422	sdir3	stok_0_404	raw405	sdbn5
hist_0_54	raw1423	sdit3	stok_0_405	raw406	sdbp5
hist_0_55	raw1424	sdiv3	stok_0_406	raw407	sdbr5
hist_0_56	raw1425	sdiz3	stok_0_407	raw408	sdbt5
hist_0_57	raw1426	sdjb3	stok_0_408	raw409	sdhg5
hist_0_58	raw1427	sdjd3	stok_0_409	raw410	sdhi5
hist_0_59	raw1428	sdjf3	stok_0_41	raw42	sdbh1
hist_0_6	raw1375	sdq3	stok_0_410	raw411	sdhk5
hist_0_60	raw1429	sdjj3	stok_0_411	raw412	sdhm5
hist_0_61	raw1430	sdjl3	stok_0_412	raw413	sdho5
hist_0_62	raw1431	sdjn3	stok_0_413	raw414	sdhq5
hist_0_63	raw1432	sdjp3	stok_0_414	raw415	sdhs5
hist_0_64	raw1433	sdjt3	stok_0_415	raw416	sdhu5
hist_0_65	raw1434	sdjv3	stok_0_416	raw417	sdhw5
hist_0_66	raw1435	sdjx3	stok_0_417	raw418	sdhy5
hist_0_67	raw1436	sdjz3	stok_0_418	raw419	sdia5
hist_0_68	raw1437	sdkd3	stok_0_419	raw420	sdic5
hist_0_69	raw1438	sdkf3	stok_0_42	raw43	sdbj1
hist_0_7	raw1376	sds3	stok_0_420	raw421	sdie5
hist_0_70	raw1439	sdkh3	stok_0_421	raw422	sdig5
hist_0_71	raw1440	sdkj3	stok_0_422	raw423	sdii5
hist_0_72	raw1441	sdkn3	stok_0_423	raw424	sdik5
hist_0_73	raw1442	sdkp3	stok_0_424	raw425	sdim5

datafile name	raw number	dev name	datafile name	raw number	dev name
hist_0_74	raw1443	sdkr3	stok_0_425	raw426	sdio5
hist_0_75	raw1444	sdkt3	stok_0_426	raw427	sdiq5
hist_0_76	raw1445	sdkx3	stok_0_427	raw428	sdis5
hist_0_77	raw1446	sdkz3	stok_0_428	raw429	sdiu5
hist_0_78	raw1447	sdlb3	stok_0_429	raw430	sdiw5
hist_0_79	raw1448	sdld3	stok_0_43	raw44	sdbl1
hist_0_8	raw1377	sdw3	stok_0_430	raw431	sdiy5
hist_0_80	raw1449	sdlh3	stok_0_431	raw432	sdja5
hist_0_81	raw1450	sdlj3	stok_0_432	raw433	sdjc5
hist_0_82	raw1451	sdll3	stok_0_433	raw434	sdje5
hist_0_83	raw1452	sdln3	stok_0_434	raw435	sdjg5
hist_0_84	raw1453	sdlr3	stok_0_435	raw436	sdji5
hist_0_85	raw1454	sdlt3	stok_0_436	raw437	sdjk5
hist_0_86	raw1455	sdlv3	stok_0_437	raw438	sdjm5
hist_0_87	raw1456	sdlx3	stok_0_438	raw439	sdjo5
hist_0_88	raw1457	sdmb3	stok_0_439	raw440	sdjq5
hist_0_89	raw1458	sdmd3	stok_0_44	raw45	sdbn1
hist_0_9	raw1378	sdy3	stok_0_440	raw441	sdjs5
hist_0_90	raw1459	sdmf3	stok_0_441	raw442	sdju5
hist_0_91	raw1460	sdmh3	stok_0_442	raw443	sdjw5
hist_0_92	raw1461	sdml3	stok_0_443	raw444	sdjy5
hist_0_93	raw1462	sdmn3	stok_0_444	raw445	sdka5
hist_0_94	raw1463	sdmp3	stok_0_445	raw446	sdkc5
hist_0_95	raw1464	sdmr3	stok_0_446	raw447	sdke5
icust1_0_0	raw1601	sdhl6	stok_0_447	raw448	sdkg5
icust1_0_1	raw1602	sdhn6	stok_0_448	raw449	sdki5
icust1_0_10	raw1611	sdij6	stok_0_449	raw450	sdkk5
icust1_0_11	raw1612	sdil6	stok_0_45	raw46	sdbp1
icust1_0_12	raw1613	sdip6	stok_0_450	raw451	sdkm5
icust1_0_13	raw1614	sdir6	stok_0_451	raw452	sdko5
icust1_0_14	raw1615	sdit6	stok_0_452	raw453	sdkq5
icust1_0_15	raw1616	sdiv6	stok_0_453	raw454	sdks5
icust1_0_16	raw1617	sdiz6	stok_0_454	raw455	sdku5
icust1_0_17	raw1618	sdjb6	stok_0_455	raw456	sdkw5
icust1_0_18	raw1619	sdjd6	stok_0_456	raw457	sdky5
icust1_0_19	raw1620	sdjf6	stok_0_457	raw458	sdla5
icust1_0_2	raw1603	sdhp6	stok_0_458	raw459	sdlc5
icust1_0_20	raw1621	sdjj6	stok_0_459	raw460	sdle5
icust1_0_21	raw1622	sdjl6	stok_0_46	raw47	sdbr1
icust1_0_22	raw1623	sdjn6	stok_0_460	raw461	sdlg5
icust1_0_23	raw1624	sdjp6	stok_0_461	raw462	sdli5
icust1_0_24	raw1625	sdjt6	stok_0_462	raw463	sdlk5
icust1_0_25	raw1626	sdjv6	stok_0_463	raw464	sdlm5
icust1_0_26	raw1627	sdjx6	stok_0_464	raw465	sdlo5
icust1_0_27	raw1628	sdjz6	stok_0_465	raw466	sdlq5
icust1_0_28	raw1629	sdkd6	stok_0_466	raw467	sdls5
icust1_0_29	raw1630	sdkf6	stok_0_467	raw468	sdlu5
icust1_0_3	raw1604	sdhr6	stok_0_468	raw469	sdlw5
icust1_0_30	raw1631	sdkh6	stok_0_469	raw470	sdly5
icust1_0_31	raw1632	sdkj6	stok_0_47	raw48	sdbt1
icust1_0_32	raw1633	sdkn6	stok_0_470	raw471	sdma5
icust1_0_33	raw1634	sdkp6	stok_0_471	raw472	sdmc5
icust1_0_34	raw1635	sdkr6	stok_0_472	raw473	sdme5
icust1_0_35	raw1636	sdkt6	stok_0_473	raw474	sdmg5
icust1_0_36	raw1637	sdkx6	stok_0_474	raw475	sdmi5

datafile name	raw number	dev name	datafile name	raw number	dev name
icust1_0_37	raw1638	sdkz6	stok_0_475	raw476	sdmk5
icust1_0_38	raw1639	sdlb6	stok_0_476	raw477	sdmm5
icust1_0_39	raw1640	sdld6	stok_0_477	raw478	sdmo5
icust1_0_4	raw1605	sdhv6	stok_0_478	raw479	sdmq5
icust1_0_40	raw1641	sdlh6	stok_0_479	raw480	sdms5
icust1_0_41	raw1642	sdlj6	stok_0_48	raw49	sdhg1
icust1_0_42	raw1643	sdll6	stok_0_480	raw1765	sdcc8
icust1_0_43	raw1644	sdln6	stok_0_49	raw50	sdhi1
icust1_0_44	raw1645	sdlr6	stok_0_5	raw6	sdl1
icust1_0_45	raw1646	sdlt6	stok_0_50	raw51	sdhk1
icust1_0_46	raw1647	sdlv6	stok_0_51	raw52	sdhm1
icust1_0_47	raw1648	sdlx6	stok_0_52	raw53	sdho1
icust1_0_5	raw1606	sdhx6	stok_0_53	raw54	sdhq1
icust1_0_6	raw1607	sdhz6	stok_0_54	raw55	sdhs1
icust1_0_7	raw1608	sdib6	stok_0_55	raw56	sdhu1
icust1_0_8	raw1609	sdif6	stok_0_56	raw57	sdhw1
icust1_0_9	raw1610	sdih6	stok_0_57	raw58	sdhy1
icust2_0_0	raw1177	sdc1	stok_0_58	raw59	sdia1
icust2_0_1	raw1178	sde1	stok_0_59	raw60	sdic1
icust2_0_10	raw1187	sdbw1	stok_0_6	raw7	sdn1
icust2_0_11	raw1188	sdby1	stok_0_60	raw61	sdie1
icust2_0_12	raw1189	sdcc1	stok_0_61	raw62	sdig1
icust2_0_13	raw1190	sdce1	stok_0_62	raw63	sdii1
icust2_0_14	raw1191	sdcg1	stok_0_63	raw64	sdik1
icust2_0_15	raw1192	sdci1	stok_0_64	raw65	sdim1
icust2_0_16	raw1193	sdcm1	stok_0_65	raw66	sdio1
icust2_0_17	raw1194	sdco1	stok_0_66	raw67	sdiq1
icust2_0_18	raw1195	sdcq1	stok_0_67	raw68	sdis1
icust2_0_19	raw1196	sdcs1	stok_0_68	raw69	sdiu1
icust2_0_2	raw1179	sdg1	stok_0_69	raw70	sdiw1
icust2_0_20	raw1197	sdac1	stok_0_7	raw8	sdp1
icust2_0_21	raw1198	sdae1	stok_0_70	raw71	sdiy1
icust2_0_22	raw1199	sdag1	stok_0_71	raw72	sdja1
icust2_0_23	raw1200	sdai1	stok_0_72	raw73	sdjc1
icust2_0_24	raw1201	sdam1	stok_0_73	raw74	sdje1
icust2_0_25	raw1202	sdao1	stok_0_74	raw75	sdjg1
icust2_0_26	raw1203	sdaq1	stok_0_75	raw76	sdji1
icust2_0_27	raw1204	sdas1	stok_0_76	raw77	sdjk1
icust2_0_28	raw1205	sdaw1	stok_0_77	raw78	sdjm1
icust2_0_29	raw1206	sday1	stok_0_78	raw79	sdjo1
icust2_0_3	raw1180	sdi1	stok_0_79	raw80	sdjq1
icust2_0_30	raw1207	sdba1	stok_0_8	raw9	sdr1
icust2_0_31	raw1208	sdbc1	stok_0_80	raw81	sdjs1
icust2_0_32	raw1209	sdbg1	stok_0_81	raw82	sdju1
icust2_0_33	raw1210	sdbi1	stok_0_82	raw83	sdjw1
icust2_0_34	raw1211	sdbk1	stok_0_83	raw84	sdjy1
icust2_0_35	raw1212	sdbm1	stok_0_84	raw85	sdka1
icust2_0_36	raw1213	sdbq1	stok_0_85	raw86	sdkc1
icust2_0_37	raw1214	sdbs1	stok_0_86	raw87	sdke1
icust2_0_38	raw1215	sdbu1	stok_0_87	raw88	sdkg1
icust2_0_39	raw1216	sdhh1	stok_0_88	raw89	sdki1
icust2_0_4	raw1181	sdm1	stok_0_89	raw90	sdkk1
icust2_0_40	raw1217	sdhl1	stok_0_9	raw10	sdt1
icust2_0_41	raw1218	sdhn1	stok_0_90	raw91	sdkm1
icust2_0_42	raw1219	sdhp1	stok_0_91	raw92	sdko1

datafile name	raw number	dev name	datafile name	raw number	dev name
icust2_0_43	raw1220	sdhr1	stok_0_92	raw93	sdkq1
icust2_0_44	raw1221	sdhv1	stok_0_93	raw94	sdks1
icust2_0_45	raw1222	sdhx1	stok_0_94	raw95	sdku1
icust2_0_46	raw1223	sdhz1	stok_0_95	raw96	sdkw1
icust2_0_47	raw1224	sdib1	stok_0_96	raw97	sdky1
icust2_0_48	raw1225	sdif1	stok_0_97	raw98	sdla1
icust2_0_49	raw1226	sdih1	stok_0_98	raw99	sdlc1
icust2_0_5	raw1182	sdo1	stok_0_99	raw100	sdle1
icust2_0_50	raw1227	sdij1	system_1	raw1759	sdq8
icust2_0_51	raw1228	sdil1	temp_0_0	raw1657	sdc7
icust2_0_52	raw1229	sdip1	temp_0_1	raw1658	sde7
icust2_0_53	raw1230	sdir1	temp_0_10	raw1667	sdbw7
icust2_0_54	raw1231	sdit1	temp_0_11	raw1668	sdby7
icust2_0_55	raw1232	sdiv1	temp_0_12	raw1669	sdcc7
icust2_0_56	raw1233	sdiz1	temp_0_13	raw1670	sdce7
icust2_0_57	raw1234	sdjb1	temp_0_14	raw1671	sdcg7
icust2_0_58	raw1235	sdjd1	temp_0_15	raw1672	sdci7
icust2_0_59	raw1236	sdjf1	temp_0_16	raw1673	sdcm7
icust2_0_6	raw1183	sdq1	temp_0_17	raw1674	sdco7
icust2_0_60	raw1237	sdjj1	temp_0_18	raw1675	sdcq7
icust2_0_61	raw1238	sdjl1	temp_0_19	raw1676	sdcs7
icust2_0_62	raw1239	sdjn1	temp_0_2	raw1659	sdg7
icust2_0_63	raw1240	sdjp1	temp_0_20	raw1677	sdac7
icust2_0_64	raw1241	sdjt1	temp_0_21	raw1678	sdae7
icust2_0_65	raw1242	sdjv1	temp_0_22	raw1679	sdag7
icust2_0_66	raw1243	sdjx1	temp_0_23	raw1680	sdai7
icust2_0_67	raw1244	sdjz1	temp_0_24	raw1681	sdam7
icust2_0_68	raw1245	sdkd1	temp_0_25	raw1682	sdao7
icust2_0_69	raw1246	sdkf1	temp_0_26	raw1683	sdaq7
icust2_0_7	raw1184	sds1	temp_0_27	raw1684	sdas7
icust2_0_70	raw1247	sdkh1	temp_0_28	raw1685	sdaw7
icust2_0_71	raw1248	sdkj1	temp_0_29	raw1686	sday7
icust2_0_72	raw1249	sdkn1	temp_0_3	raw1660	sdi7
icust2_0_73	raw1250	sdkp1	temp_0_30	raw1687	sdba7
icust2_0_74	raw1251	sdkr1	temp_0_31	raw1688	sdbc7
icust2_0_75	raw1252	sdkt1	temp_0_32	raw1689	sdbg7
icust2_0_76	raw1253	sdkx1	temp_0_33	raw1690	sdbi7
icust2_0_77	raw1254	sdkz1	temp_0_34	raw1691	sdbk7
icust2_0_78	raw1255	sdlb1	temp_0_35	raw1692	sdbm7
icust2_0_79	raw1256	sdld1	temp_0_36	raw1693	sdbq7
icust2_0_8	raw1185	sdw1	temp_0_37	raw1694	sdbs7
icust2_0_80	raw1257	sdlh1	temp_0_38	raw1695	sdbu7
icust2_0_81	raw1258	sdlj1	temp_0_39	raw1696	sdhh7
icust2_0_82	raw1259	sdll1	temp_0_4	raw1661	sdm7
icust2_0_83	raw1260	sdln1	temp_0_40	raw1697	sdhl7
icust2_0_84	raw1261	sdlr1	temp_0_41	raw1698	sdhn7
icust2_0_85	raw1262	sdlt1	temp_0_42	raw1699	sdhp7
icust2_0_86	raw1263	sdlv1	temp_0_43	raw1700	sdhr7
icust2_0_87	raw1264	sdlx1	temp_0_44	raw1701	sdhv7
icust2_0_88	raw1265	sdmb1	temp_0_45	raw1702	sdhx7
icust2_0_89	raw1266	sdmd1	temp_0_46	raw1703	sdhz7
icust2_0_9	raw1186	sdy1	temp_0_47	raw1704	sdib7
icust2_0_90	raw1267	sdmf1	temp_0_48	raw1705	sdif7
icust2_0_91	raw1268	sdmh1	temp_0_49	raw1706	sdih7

datafile name	raw number	dev name	datafile name	raw number	dev name
icust2_0_92	raw1269	sdml1	temp_0_5	raw1662	sdo7
icust2_0_93	raw1270	sdmn1	temp_0_50	raw1707	sdij7
icust2_0_94	raw1271	sdmp1	temp_0_51	raw1708	sdil7
icust2_0_95	raw1272	sdmr1	temp_0_52	raw1709	sdip7
idist_0_0	raw1757	sdm8	temp_0_53	raw1710	sdir7
iitem_0_0	raw1758	sdo8	temp_0_54	raw1711	sdit7
iordr2_0_0	raw1273	sdc2	temp_0_55	raw1712	sdiv7
iordr2_0_1	raw1274	sde2	temp_0_56	raw1713	sdiz7
iordr2_0_10	raw1283	sdbw2	temp_0_57	raw1714	sdjb7
iordr2_0_11	raw1284	sdby2	temp_0_58	raw1715	sdjd7
iordr2_0_12	raw1285	sdcc2	temp_0_59	raw1716	sdjf7
iordr2_0_13	raw1286	sdce2	temp_0_6	raw1663	sdq7
iordr2_0_14	raw1287	sdcg2	temp_0_60	raw1717	sdjj7
iordr2_0_15	raw1288	sdci2	temp_0_61	raw1718	sdjl7
iordr2_0_16	raw1289	sdcm2	temp_0_62	raw1719	sdjn7
iordr2_0_17	raw1290	sdco2	temp_0_63	raw1720	sdjp7
iordr2_0_18	raw1291	sdcq2	temp_0_64	raw1721	sdjt7
iordr2_0_19	raw1292	sdcs2	temp_0_65	raw1722	sdjv7
iordr2_0_2	raw1275	sdg2	temp_0_66	raw1723	sdjx7
iordr2_0_20	raw1293	sdac2	temp_0_67	raw1724	sdjz7
iordr2_0_21	raw1294	sdae2	temp_0_68	raw1725	sdkd7
iordr2_0_22	raw1295	sdag2	temp_0_69	raw1726	sdkf7
iordr2_0_23	raw1296	sdai2	temp_0_7	raw1664	sds7
iordr2_0_24	raw1297	sdam2	temp_0_70	raw1727	sdkh7
iordr2_0_25	raw1298	sdao2	temp_0_71	raw1728	sdkj7
iordr2_0_26	raw1299	sdaq2	temp_0_72	raw1729	sdkn7
iordr2_0_27	raw1300	sdas2	temp_0_73	raw1730	sdkp7
iordr2_0_28	raw1301	sdaw2	temp_0_74	raw1731	sdkr7
iordr2_0_29	raw1302	sday2	temp_0_75	raw1732	sdkt7
iordr2_0_3	raw1276	sdi2	temp_0_76	raw1733	sdkx7
iordr2_0_30	raw1303	sdba2	temp_0_77	raw1734	sdkz7
iordr2_0_31	raw1304	sdbc2	temp_0_78	raw1735	sdlb7
iordr2_0_32	raw1305	sdbg2	temp_0_79	raw1736	sdld7
iordr2_0_33	raw1306	sdbi2	temp_0_8	raw1665	sdw7
iordr2_0_34	raw1307	sdbk2	temp_0_80	raw1737	sdlh7
iordr2_0_35	raw1308	sdbm2	temp_0_81	raw1738	sdlj7
iordr2_0_36	raw1309	sdbq2	temp_0_82	raw1739	sdll7
iordr2_0_37	raw1310	sdbs2	temp_0_83	raw1740	sdln7
iordr2_0_38	raw1311	sdbu2	temp_0_84	raw1741	sdlr7
iordr2_0_39	raw1312	sdhh2	temp_0_85	raw1742	sdlt7
iordr2_0_4	raw1277	sdm2	temp_0_86	raw1743	sdlv7
iordr2_0_40	raw1313	sdhl2	temp_0_87	raw1744	sdlx7
iordr2_0_41	raw1314	sdhn2	temp_0_88	raw1745	sdmb7
iordr2_0_42	raw1315	sdhp2	temp_0_89	raw1746	sdmd7
iordr2_0_43	raw1316	sdhr2	temp_0_9	raw1666	sdy7
iordr2_0_44	raw1317	sdhv2	temp_0_90	raw1747	sdmf7
iordr2_0_45	raw1318	sdhx2	temp_0_91	raw1748	sdmh7
iordr2_0_46	raw1319	sdhz2	temp_0_92	raw1749	sdml7
iordr2_0_47	raw1320	sdib2	temp_0_93	raw1750	sdmn7
iordr2_0_48	raw1321	sdif2	temp_0_94	raw1751	sdmp7
iordr2_0_49	raw1322	sdih2	temp_0_95	raw1752	sdmr7
iordr2_0_5	raw1278	sdo2	tpccaux	raw1760	sds8
iordr2_0_50	raw1323	sdij2	ware_0_0	raw1754	sde8
iordr2_0_51	raw1324	sdil2			

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/1, 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.

The database was neither partitioned nor replicated.

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:	1,354,086 tpmC
Price per tpmC:	$3.25 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	0.508	0.992	6.389
Payment	0.499	0.983	6.401
Order-Status	0.507	0.992	6.132
Interactive Delivery	0.103	0.104	0.324
Deferred Delivery	0.404	0.887	4.988
Stock-Level	0.491	0.976	6.137
Menu	0.103	0.104	0.358

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	18.034
Payment	3.002	3.012	3.036
Order-Status	2.003	2.012	2.034
Interactive Delivery	2.003	2.012	2.033
Stock-Level	2.003	2.012	2.030

Table 5.3 Think Times

Type	Minimum	Average	Maximum
New-Order	0.000	12.016	120.198
Payment	0.000	12.015	120.196
Order-Status	0.000	10.016	100.187
Interactive Delivery	0.000	5.017	50.184
Stock-Level	0.000	5.018	50.193

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

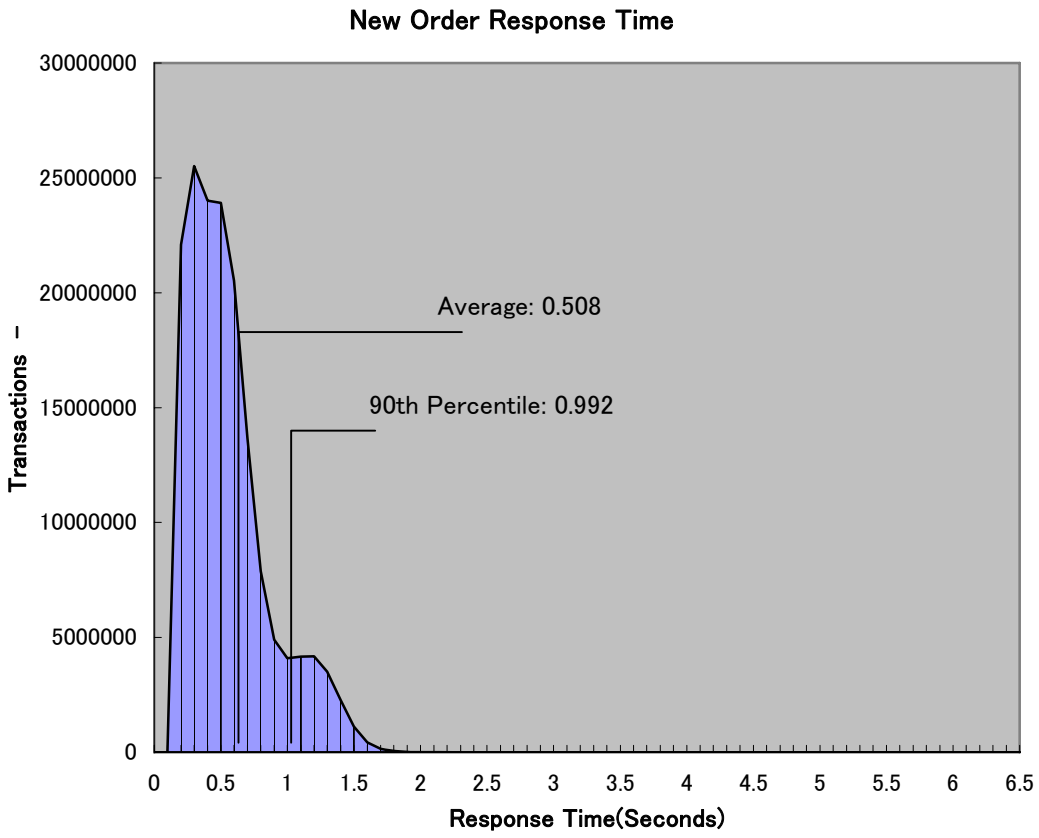


Figure 5.2: Payment Response Time Distribution

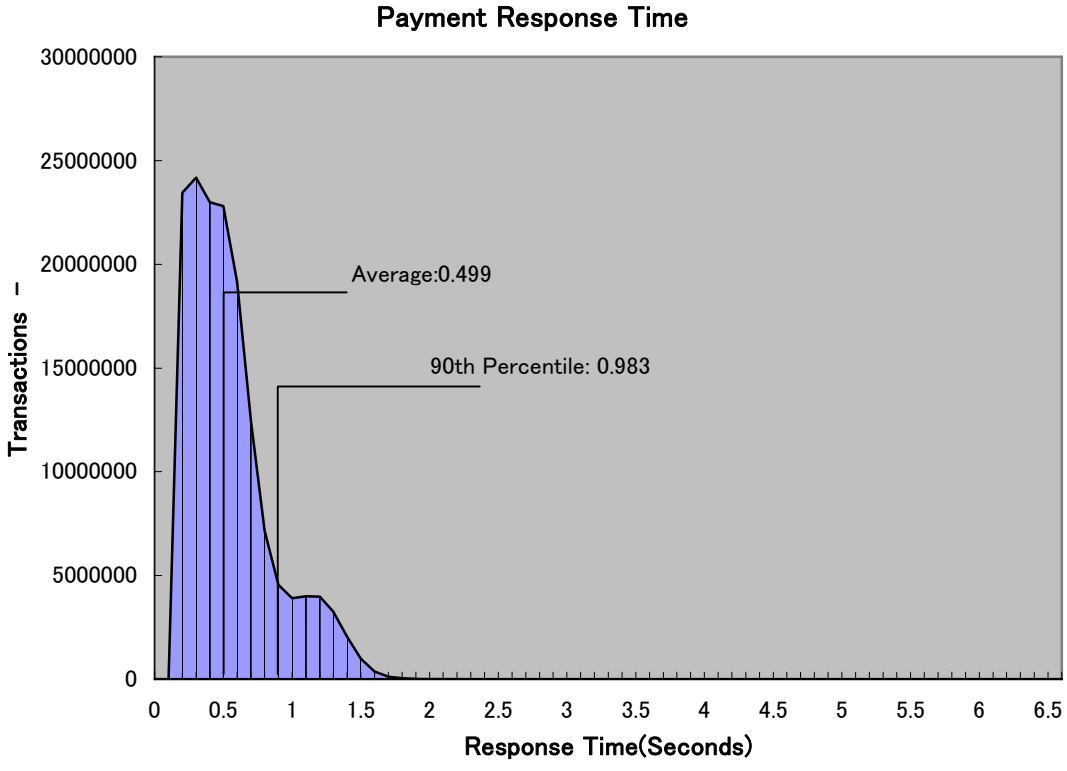


Figure 5.3: Order Status Response Time Distribution

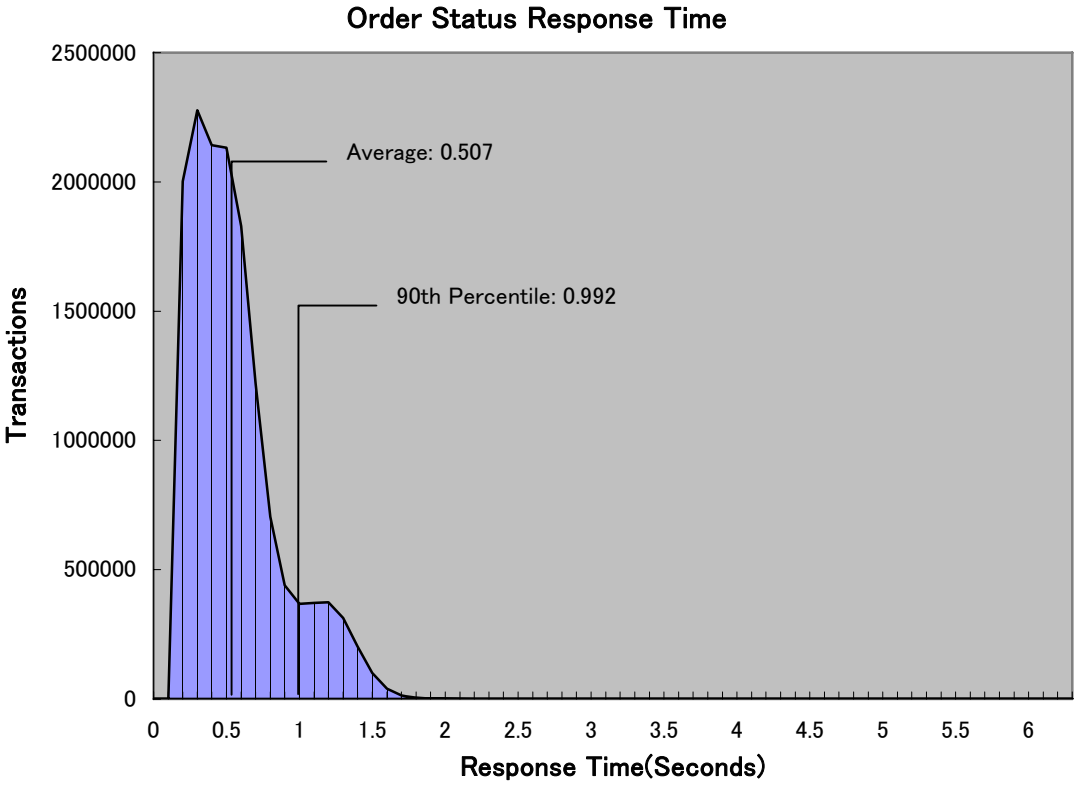


Figure 5.4: Delivery Response Time Distribution

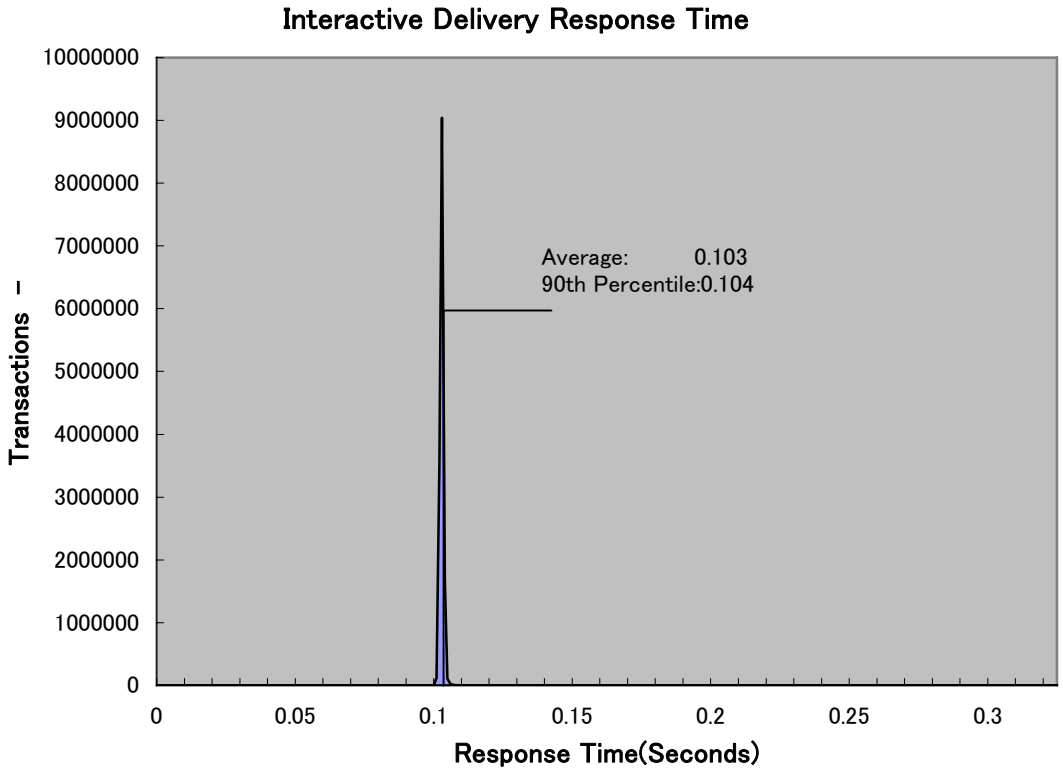


Figure 5.5: Stock Level Response Time Distribution

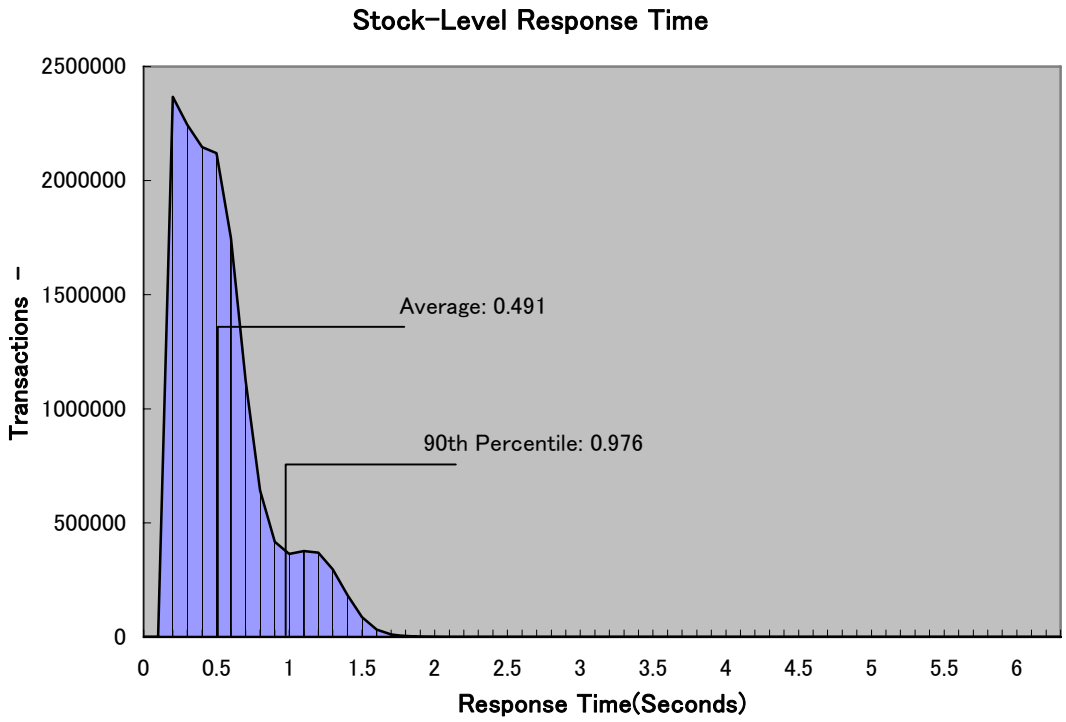


Figure 5.6: New Order Think Time Frequency Distribution

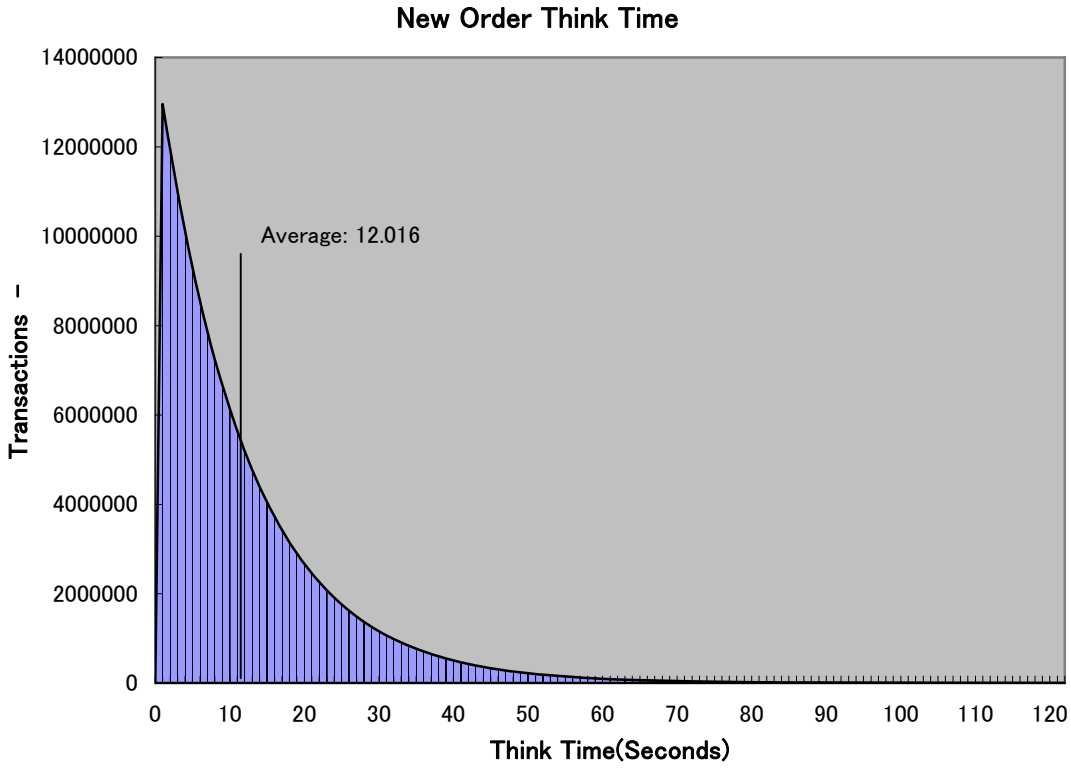


Figure 5.7: New-Order Response time vs. Throughput

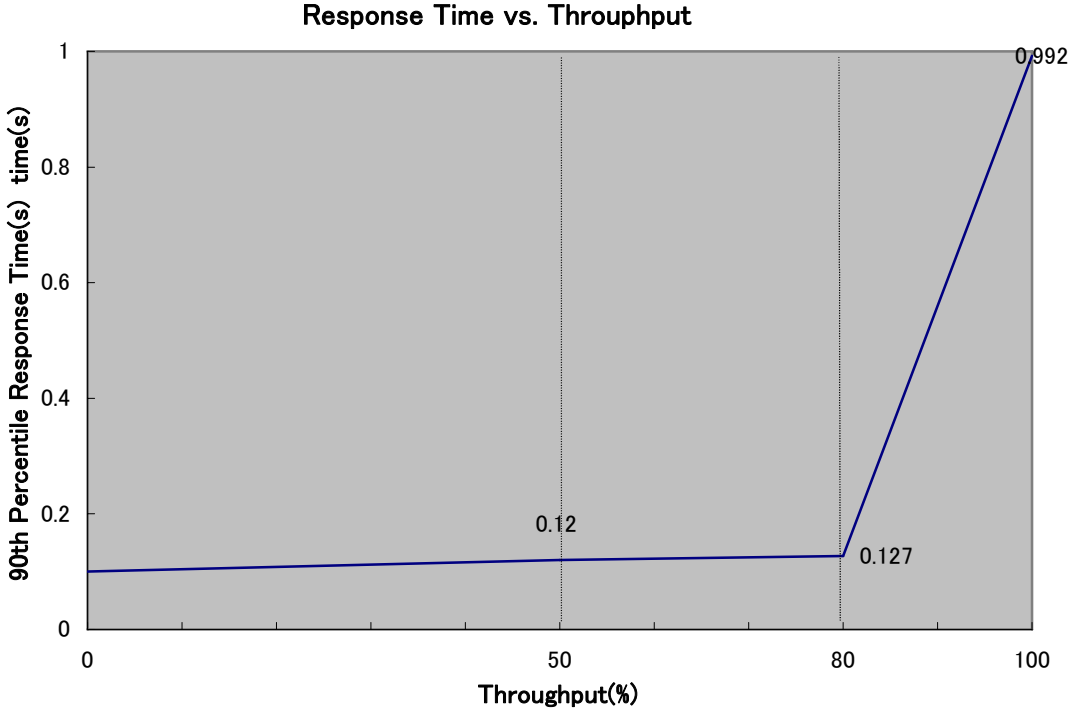
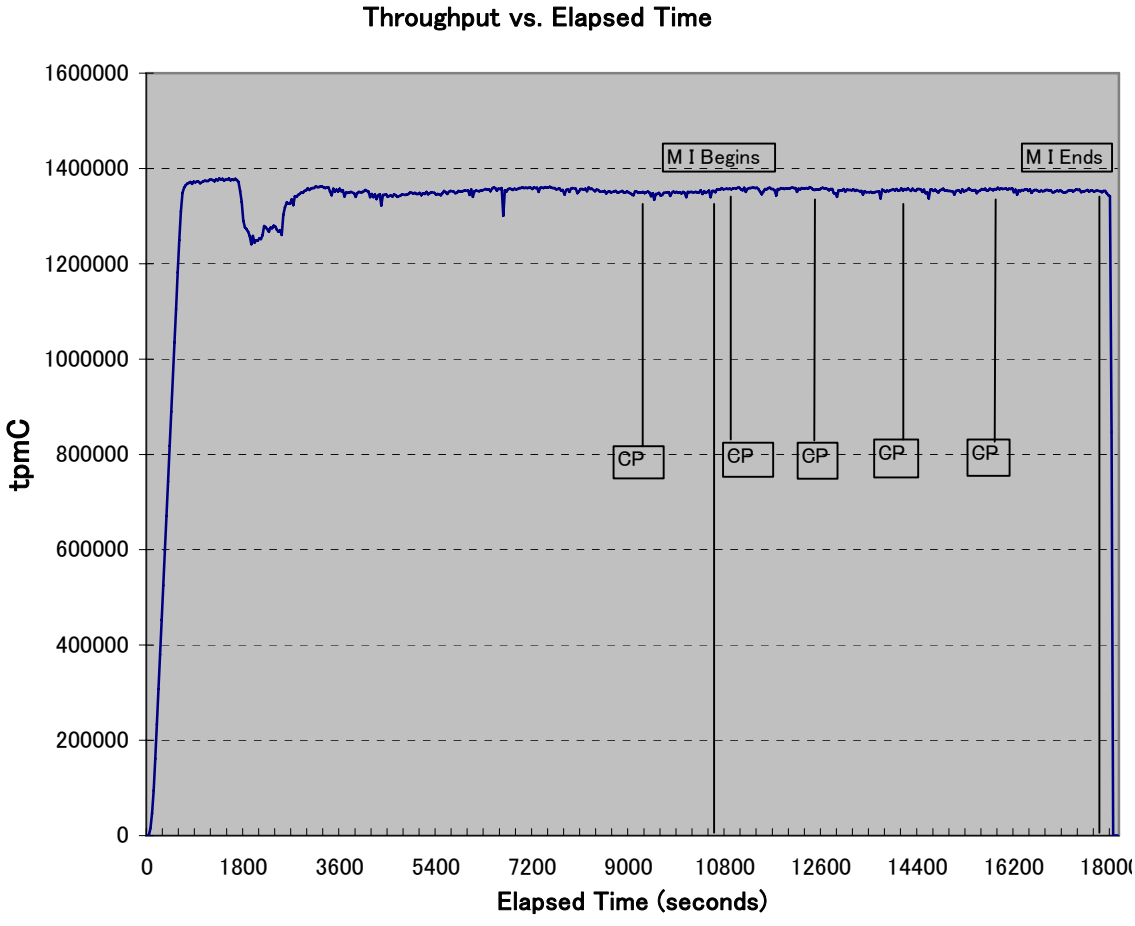


Figure 5.8: Throughput versus Elapsed Time



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 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 it's 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	44.96%
	Payment	43.01%
	Order status	4.01%
	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	59.99%
Order Status	Accessed by last name	60.00%
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	21:27:35	23:27:35	120 minutes
	start	End	duration
checkpoint 0	21:05:50	21:33:19	27:29
checkpoint 1	21:34:17	22:01:44	27:27
checkpoint 2	22:02:51	22:30:12	27:21
checkpoint 3	22:31:28	22:58:50	27:22
checkpoint 4	23:00:00	23:27:13	27:13

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 replace 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.

Four 1Gbps ethernet LAN connections were used between the server and four switches, to which clients were connected by 48(forty-eight) 100Mbps ethernet LAN connections. Another 48(forty-eight) 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 November 22, 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 :	1,354,086 tpmC
Price per tpmC :	$3.25 USD per tpmC
Three-year cost of ownership :	$4,397,010 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:

- 16 Oracle Database 10g Enterprise Edition, Per Processor, Unlimited Users for 3 years
- 1 Red Hat Enterprise Linux AS (Standard)
- 48 Red Hat Enterprise Linux ES (Standard)
- 48 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 linr 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
cogiguration file.   *
 * Return Value                                                  *
 *      Get string length                                        *

*****************************************************************
****/
int    GetPrivateProfileString(char*
section_name,    /* Section name
*/
                        char*    key_name,       /* Key
name                        */
                        char*    delault_str,    /* Default
string, if kye 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 DEFALT_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 DEFALT_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 DEFALT_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 DEFALT_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);

DEFALT_STRING_FCLOSE:
    fclose(prof_file);

DEFALT_STRING:
    strncpy(key_data, delault_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
COMOBJS    = log.o sema.o
GetPrivateProfileString.o shmem.o
COMLIB  = libcom.a

INCFILES = log.h sema.h forlinux.h shmem.h

$(COMLIB) : $(COMOBJS)
    $(AR) $(ARFLAGS) $(COMLIB) $(COMOBJS)

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

$(COMOBJS) : $(INCFILES)

clean:
    rm  $(COMLIB) $(COMOBJS)
```

```
::::::::::::::::::::::::::::::::::
 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 TpccUserLog(char* file_name, int line_no,
char* type_name, char* fmtp, ...)
{
    FILE*       fp;
    pid_t       pid;
    pthread_t    tId;
    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. */
    tId = pthread_self();

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

    va_start(va, fmtp);

    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)tId, fname, line_no,
type_name);
    vfprintf(fp, fmtp, va);

    if (*(fmtp + strlen(fmtp) - 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 TpccUserLog (char *file_name, int line_no,
char* type_name, char* fmtp, ...);

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;

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

    if ((sid = GetSem(path, projectId)) == -1) {
      TpccUserLog(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_PERFOREMANCE
(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 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 ((shmid=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_PERFOREMANCE
(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 monitorring 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 ...                             *
 *   usetlog.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>\r\n\
<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
informtion</TITLE></HEAD><BODY>\
<PRE>\
Information of  %s \r\n\
\r\n\
TpApl performance \r\n\
                        New     Pay     Odr     Del
Sto\r\n\
    Num of waiting process  %-7d  %-7d  %-
7d  %-7d  %-7d\r\n\
    Answer to RTE (ms)      %-7d  %-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\
    Paymant 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"


#if 0
#define CLIENT_SAMPLE "\
<HTML><HEAD><TITLE>Client Sampling
informtion</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\
    Paymant 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,"\

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

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

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

"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_semid);
   if ((cur_sec - GLBSMP_shared_mem-
>CsvOutTime) < GLBSMP_shared_mem-
>SamplingInterval) {
      /* No output timming */
      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.NumQueNewOrder);
    fprintf(csv_file, "%d,",
sampling_data.NumQuePayment);
    fprintf(csv_file, "%d,",
sampling_data.NumQueOrderStatus);
    fprintf(csv_file, "%d,",
sampling_data.NumQueDelivery);
    fprintf(csv_file, "%d,",
sampling_data.NumQueStockLevel);

    /* Repsponse time & number of prosessing
trasaction */
    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(GLBSMP_semid);
    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 ; 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 editting area is 7byts    */
   /* 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-
%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 :mum 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 ... Exsample:
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_SvrAplErr                                       *
 *   (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_oraerr :
!   this function make error message of the
Oracle application program.
!*/
!int set_oraerr (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
!int 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_Numwh;
            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:          /* I_ID data is uninput */
            sprintf(errmsg,errstrings[13],err_no-
12,errstrings[15],errstrings[18]);
            sub_inf2 = 100000;
            break;
        case -6:          /* I_ID data is abnormal */
            sprintf(errmsg,errstrings[13],err_no-
12,errstrings[15],errstrings[20]);
            sub_inf2 = 100000;
            break;
        case -16:         /* I_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;
            }

            sprintf(buf, errhtml, errmsg, SOPATH,
user);
//            printf("%s", buf);
        }

//    DBGR(fprintf (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* delault_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() {

    /* Check INI file exist */
    if (access(GLB_ConfigFilePath, 0x00) != 0) {
        /* INI file no exist, using defalt value */
        TpccUserLog(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;
    }
    TpccUserLog(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 gettiing */
            continue;
        }
        break;
    }
#ifdef PUT_INF_LOG
    TpccUserLog(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 gettiing */
            continue;
        }
        break;
    }
#ifdef PUT_INF_LOG
    TpccUserLog(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
*
 *                                                          *
 *  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
*
 *                                                          *
 *  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    TPCONTEXT_T
GLB_TpContext;
#endif
extern    TPCONTEXT_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                                 *
 *   (1) GetThreadKey                               *
 *   (2) CreateTuxEnv                               *
 *   (3) DestroyThread                              *
 *   (4) FreeThreadKey                              *
 *   (5) GetThreadCntl                              *
 *   (6) RegistTuxApl                               *
 *   (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
   TpccUserLog (LOG_INF, "Thread key
creating start [GetThreadKey]\n");
#endif

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

#ifdef PUT_INF_LOG
   TpccUserLog (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*             itf_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
      TpccUserLog(LOG_INF, "Thread initialize
started \n");
#endif

      /* First execution in this thread */
#ifndef SCRTEST
      /* Regist context */
/*
 * 2006.03.29 T.Motoo: Modified because child
process came to have one or more
 *                  contexts.
 */
#if 0
!      if (tpsetctxt(GLB_TpContext, 0) == -1) {
#endif
      if (tpsetctxt(GLB_TpContext[(id %
GLB_ThreadLimit) /
                        TUXCDPERCTXT], 0) ==
-1) {
         TpccUserLog(LOG_ERR, "tpsetctxt()
failed\n");
         return(0);
      }
#endif

      /* Get query data area */
#ifndef USEPOOL_QUERY
      if ((query_str =
(char*)malloc(QUERY_STR_SIZE)) == NULL) {
         TpccUserLog(LOG_ERR, "malloc() failed
for query string buffer (size=%d)\n",
QUERY_STR_SIZE);
         return(0);
      }
#else
      query_str = NULL;
#endif
      /* Get response editting area */
      if ((resp_buf =
(char*)malloc(RESP_BUF_SIZE)) == NULL) {
         TpccUserLog(LOG_ERR, "malloc() failed
for response editting 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) {
         TpccUserLog(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) &
0xfffffff0;

#ifdef CONST_TUX_BUF

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

#else
      itf_buf = 0;
#endif
      /* Set each pointer */
      ThreadCntlInfo->TrxData        = itf_buf;
      ThreadCntlInfo->TrxDataLeng  = buf_leng;
      ThreadCntlInfo->QueryData     = query_str;
      ThreadCntlInfo->RespBuf        = resp_buf;

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

#ifndef SCRTEST
         tpfree(itf_buf);
#else
         free(itf_buf);
#endif

#endif
         return(0);
      }

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

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

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

#ifndef 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) {
            TpccUserLog(LOG_ERR,
"pthread_setspecific() failed for Thread
destroied\n");
        }

    }

#ifdef PUT_INF_LOG
    TpccUserLog(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) {
        TpccUserLog(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) {
        TpccUserLog(LOG_ERR, "Thread cntrol
information is not allocated.\n");
        return 0;
    }

#ifdef CONST_TUX_BUF
    /* Nothing to do */

#else
#ifndef SCRTEST
    if ((ThreadCntlInfo->TrxData = (char
*)tpalloc("CARRAY", NULL, ThreadCntlInfo-
>TrxDataLeng)) == NULL ) {
        TpccUserLog(LOG_ERR, "tpalloc() failed
for interface data buffer (size=%d)\n",
ThreadCntlInfo->TrxDataLeng);
        return(0);
    }
#else
    if ((ThreadCntlInfo->TrxData = (char
*)calloc( ThreadCntlInfo->TrxDataLeng, 1)) ==
NULL ) {
        TpccUserLog(LOG_ERR, "calloc() failed for
interface data buffer (size=%d)\n",
ThreadCntlInfo->TrxDataLeng);
        return(0);
    }
#endif
#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->TrxData != 0) {
#ifndef SCRTEST
        tpfree(ThreadCntlInfo->TrxData);
#else
        free(ThreadCntlInfo->TrxData);
#endif
        ThreadCntlInfo->TrxData = 0;
    }
#endif

    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
!TPCONTEXT_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) {
            TpccUserLog(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 */
!           TpccUserLog(LOG_ERR, "tpinit() faild
(%s)\n", tpstrerror(tperrno));
!           return 0;
!       }
!   }
!
!   /* Get my context */
!   if (tpgetctxt(&ctx, 0) == -1) {
!       TpccUserLog(LOG_ERR, "Failed to get
Tuxedo context (%s)\n", tpstrerror(tperrno));
!       return 0;
!   }
#endif
        /*
         * Gets "ThreadsPerChild" and
"ThreadLimit".
         */

ap_mpm_query(AP_MPMQ_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) {
         TpccUserLog(LOG_ERR, "ap_palloc
faild for contexts\n");
         return 0;
      }

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

         /*
          * Gets the context.
          */
         if (tpgetctxt((ctx + i), 0) == -1) {
            TpccUserLog(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
   TpccUserLog(LOG_INF, "Child process
terminated start. \n");
#endif
   /* Leave from TUXEDO aplication */
   if (GLB_TpContext != 0) {
      if (tpterm() == -1) {
         TpccUserLog(LOG_ERR, "tpterm() failed
for Thread destroied\n");
      }
      GLB_TpContext = 0;
   }

   /* Delete TSD key */
   FreeThreadKey();

#ifdef PUT_INF_LOG
   TpccUserLog(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


# Make library
cd /home/tpc/client_apl/tpapl
echo "=="                    >> make_result.txt
echo "====== Remake library"  >>
make_result.txt
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
APA0DIR  = /usr/include/apr-0
APLDIR  = $(TOPDIR)/tpapl


# include directory
COM_INC = -I$(TOPDIR)/common
TUX_INC = -I$(TUXDIR)/include
APA_INC = -I$(APADIR)
APA0_INC = -I$(APA0DIR)
APL_INC = -I$(APLDIR)


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

INCLUDE = $(APL_INC) $(COM_INC) -
I$(APA0DIR) $(APA_INC) $(TUX_INC)
INCFILE    = $(APLDIR)/delpage.h \
          $(APLDIR)/GlobalArea.h \
          $(APLDIR)/log_level.h \
          $(APLDIR)/menupage.h \
          $(APLDIR)/newpage.h \
          $(APLDIR)/odrpage.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
OBJS = 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) : $(OBJS) $(INCFILE)
   $(AR) $(ARFLAGS) $(ARCH_LIB) $(OBJS)

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

$(OBJS) : $(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_semid = 0; \
              SAMPLING_DATA*
GLBSMP_shared_mem = 0;

extern    int              GLBSMP_semid;
extern    SAMPLING_DATA*
GLBSMP_shared_mem;


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

#define       MAC_SampleInitChild \
              GLBSMP_semid =
GetSem(SAMPLING_SEMPATH,
SEM_SAMPLING_PERFOREMANCE); \
              GLBSMP_shared_mem =
(SAMPLING_DATA*)GetShmem(SAMPLING_S
HMPATH,
SHMEM_SAMPLING_PERFOREMANCE,
sizeof(SAMPLING_DATA));

#define       MAC_SampleInitPerformance \
              ClientSampleInit();

/* Fanctions 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_SampleQueUp(count_area) \
            if (GLBSMP_shared_mem-
>DataSampling == DATASAMPLE_ENABLE) { \
              LockSem(GLBSMP_semid);    \
              GLBSMP_shared_mem-
>count_area++; \
              UnlockSem(GLBSMP_semid); \
              /*sleep(10);*/ \
            }

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

/* 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_semid);    \
              GLBSMP_shared_mem->resp_time
+= el_time; \

MAC_SampleRespMax(max_resp_time,
smp_max_resp_time); \
              GLBSMP_shared_mem-
>proc_trans++; \
              UnlockSem(GLBSMP_semid); \
```

```
        }

/* For onliy 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_SampleQueUp
 *  (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*    TrxData;
   int      TrxDataLeng;
   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_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 & analize
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 );

   /* Termimal 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:reruest 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);
   GetConfFileInfo();

   /* Initialize TPAPL semafore 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() faile for
TpApl log\n");
      return NULL;
   }

   /* Initialize SVRAPL semafore 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() faile 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) {

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

    /* Get TSD key */
    GetThreadKey();

    /* Regist TUXEDO aplication */
#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) {
            TpccUserLog(LOG_ERR,
"RegistTuxApl() faild\n");
    }

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

    /* Initialize performance sampling */
    MAC_SampleInitChild;

    TpccUserLog(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);

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

    return (void*)conf;
}


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

/************************************************************
*****
 *                                                *
 *       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_trn ( RTE_INPUT_DATA
*in_data, char *s_buf, int cookie );

int                 GetThreadKey();
#if 0 /* 2006.03.29 T.Motoo: The argument was
changed. */
!THREAD_CNTL_INFO*    CreateThreadEnv();
#endif
THREAD_CNTL_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
*
**********************************/

/* ----------------------------------------------------------
----------
   delpage.h
   data of delivery transaction result screen
(HTML format)
   ----------------------------------------------------------
--------- */

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

/* Screen data*/
#define h_del2 "\
Warehouse:       \r\n\
\r\n\
Carrier Number:   \r\n\
\r\n\
Execution Status:                    \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\r\n\r\n\r\n\
"
/* Tailer data */
#define h_del3 "\
</PRE><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\"
VALUE=%d>\r\n\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"New order\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Order Status\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Stock Level\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</FORM></BODY></HTML>\n"

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


::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::
 tpapl/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)
TpccUserLog(LOG_INF, ">>>>> "func" start
>>>>>");
#else
#define    MAC_PutFncEntryLog(func)    ;
#endif

/* Function exit point log    */
#ifdef    PUT_FNC_EXIT_LOG
#define    MAC_PutFncExitLog(func)
TpccUserLog(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>\r\n\
<BODY><CENTER>Transaction
Menu<BR></CENTER>\r\n\
<PRE>\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\r\n\r\n\r\n\r\n\r\n\r\n</PRE>\
<FORM ACTION=\"%s\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"c\"
VALUE=%d>\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"New order\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Order Status\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Stock Level\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</FORM></BODY></HTML>\r\n"

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

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

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

/* If connect client by WWW browser then use
this format */
/* Terminal verification use olny */
#define loginpage "\
<HTML><HEAD><TITLE>Welcome to TPC-
C : %s</TITLE></HEAD><BODY>\r\n\
<P>Please identify your terminal number for this
session.</P>\r\n\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"l\">\r\n\
Your Terminal Number:<INPUT NAME=\"c\"
SIZE=6><BR><HR>\r\n\
<INPUT TYPE=\"submit\">\r\n\
</FORM></BODY></HTML>\r\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>\r\n\
<P>Please identify your terminal number for this
session.</P>\r\n\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"l\">\r\n\
Your Terminal Number:<INPUT NAME=\"c\"
SIZE=5><BR><HR>\r\n\
<INPUT TYPE=\"submit\">\r\n\
</FORM></BODY></HTML>\r\n"

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

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

#define cntlerr "\
<HTML><HEAD><TITLE>ERROR: Can't
execute
transaction</TITLE></HEAD><BODY>\r\n\
<P>Basic control was
failed.</P></BODY></HTML>\r\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
*/
};
```

```
::::::::::::::::::::::::::::::::
 tpapl/modules.mk
::::::::::::::::::::::::::::::::

mod_tpapl.la: mod_tpapl.slo
    $(SH_LINK) $(LIBS) -rpath $(libexecdir) -
module -avoid-version  mod_tpapl.lo
DISTCLEAN_TARGETS = modules.mk
shared =  mod_tpapl.la


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

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

/* ---------------------------------------------------------------
----------
   newpage.h
   data of neworder transaction result screen
(HTML form)
   ---------------------------------------------------------------
---------- */

/* Header data */
#define h_new1 "\
<HTML><HEAD><TITLE>TPC-
WINDOW</TITLE></HEAD><BODY>\r\n\
<CENTER>New Order<BR></CENTER>\r\n\
<font size=4>\r\n<PRE>"

/* Screen data*/
#define h_new2 "\
Warehouse:       District:                    Date:
- -      : :  \r\n\
Customer:        Name:
Credit:     %Disc:   .  \r\n\
Order Number:          Number of Lines:
W_tax:   .     D_tax:   .  \r\n\r\n\
Supp-W   Item-Id  Item Name              Qty
Stock  B/G  Price    Amount\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\
                                        $  .   $   .
\r\n\
                                        $  .   $   .
\r\n\
Execution Status:
Total:  $     .  \r\n"

/* Tailer data */
#define h_new3 "\
</PRE>\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\"
VALUE=%d>\r\n\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"New order\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Order Status\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Stock Level\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</FORM></BODY></HTML>"

/* Offset to field which should set data */
int newp[] = {
0xa, 0x1c, 0x3c,
0x5c, 0x69, 0x84, 0x90,
0xa5, 0xc0, 0xd1, 0xe0,
0x138, 0x140, 0x148, 0x163, 0x169, 0x170,
0x175, 0x17e,
0x5f9, 0x62d
};



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

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

/* ---------------------------------------------------------------
----------
   odrpage.h
   data of OrderStatus transaction result screen
(HTML form)
   ---------------------------------------------------------------
---------- */

/* Header data */
#define h_order1 "\
<HTML><HEAD><TITLE>TPC-
WINDOW</TITLE></HEAD>\r\n\
<BODY><CENTER>Order-
Status<BR></CENTER>\r\n\
<font size=4>\r\n<PRE>"

/* Screen data*/
#define h_order2 "\
Warehouse:       District:   \r\n\
Customer:        Name:                          \r\n\
Cust-Balance: $      .  \r\n\r\n\
Order-Number:          Entry-Date:   - -      : :
Carrier-Number:   \r\n\
Supply-W     Item-Id    Qty     Amount
Delivery-Date\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\
                        $    .        - -   \r\n\
                        $    .        - -   \r\n\
"

/* Tailer data */
#define h_order3 \
"</PRE>\r\n\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\"
VALUE=%d>\r\n\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"New order\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Order Status\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Stock Level\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</FORM></BODY></HTML>"

/* Offset to field which should set data */
int orderp[] = {
0x0A, 0x1C,
0x2a, 0x37, 0x49, 0x4c,
0x6c,
0x87,  0x9e,  0xc4,
0x105, 0x110,  0x11b, 0x123, 0x131};   /* offset
0x39 */



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

/**********************************
*
*  COPYRIGHT FUJITSU LIMITED 2002
*  CREATE:1999.05.25 FJH
*
**********************************/

/* ---------------------------------------------------------------
----------
   paypage.h
   data of Payment transaction result screen
(HTML form)
   ---------------------------------------------------------------
---------- */

/* Header data */
#define h_pay1 "\
<HTML><HEAD><TITLE>TPC-
WINDOW</TITLE></HEAD><BODY>\r\n\
<CENTER>Payment<BR></CENTER><font
size=4>\r\n<PRE>"

/* Screen data*/
#define h_pay2 "\
```

```
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"
*/

/* Tailer data */
#define h_pay3 "\
</PRE>\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\"
VALUE=%d>\r\n\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"New order\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Order Status\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Stock Level\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
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/stpage.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\
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\r\n\r\n\r\n\r\n"

/* Tailer data */
#define h_stock3 "\
</PRE>\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\"
VALUE=%d>\r\n\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"New order\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Order Status\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Stock Level\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
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 editting 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) anly_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 "TpAplDBDependPrototype.h"
#include "sema.h"
#include "shmem.h"
#include "SampleInfo.h"

/*
  anly_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 anly_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_I_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_I_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 form
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 sampleing
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_SampleQueUp(NumQueNewOrder);
                rtn = NewOrder (s_buf, in_data,
cookie);
                MAC_SampleTuxResp(AnsNewOrder,
NumReqNewOrder, NumQueNewOrder);
                break;
```

```
        case 'D':

MAC_SampleQueUp(NumQueDelivery);
            rtn = Delivery(s_buf, in_data, cookie);
            MAC_SampleTuxRespDel;
            break;

        case 'P':

MAC_SampleQueUp(NumQuePayment);
            rtn = Payment (s_buf, in_data, cookie);
            MAC_SampleTuxResp(AnsPayment,
NumReqPayment, NumQuePayment);
            break;

        case 'S':

MAC_SampleQueUp(NumQueStockLevel);
            rtn = StockLevel(s_buf, in_data,
cookie);
            MAC_SampleTuxResp(AnsStockLevel,
NumReqStockLevel, NumQueStockLevel);
            break;

        case 'O':

MAC_SampleQueUp(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:1999.11.19 FJH
*
*  Modified TSL 2003.12.22
************************************/

/*==================================
==============================+
 FILENAME : tpccinf.h
 DESCRIPTION

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

#ifndef TPCCINF_H
#define TPCCINF_H

#define QUERY_STR_SIZE    1024
#define RESP_BUF_SIZE    4096

#define VLDATA "Ver 1.0 Linux & Tuxedo"


#ifdef SCRTEST

 #ifdef DBPRT
#define MDDATA "SCR And DP"
 #else
#define MDDATA "SCR"
 #endif

#else

 #ifdef NOSCR
#define MDDATA "DBG"
 #else
#define MDDATA "REL"
 #endif

#endif

#endif



::::::::::::::::::::::::::::::::::
 tpapl/tpcinweb.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\
```

```
Carrier Number: <INPUT NAME=\"OC\" SIZE=2
maxlength=2>\r\n\
\r\n\
Execution Status:\r\n\
</PRE><INPUT
TYPE=\"submit\"></FORM></BODY></HTML>"

#define in_delpage2 "\
<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=3><PRE>\
Warehouse:%6d\r\n\
\r\n\
Carrier Number:<INPUT NAME=\"OC\" SIZE=2
maxlength=2>\r\n\
\r\n\
Execution Status:\r\n\
</PRE><INPUT
TYPE=\"submit\"></FORM></BODY></HTML>\r
\n"

/* --------------------------
    neworder page
 * --------------------------*/
#define in_newpage "\
<HTML><HEAD><TITLE>TPC-C: New
Order</TITLE></HEAD>\r\n\
<BODY><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"N\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\"
VALUE=%d>\r\n\
<center>New Order<br></center>\r\n\
<PRE><font size=4>Warehouse:%6d  District:
<INPUT NAME=\"D\" SIZE=2 maxlength=2>
Date:\r\n\
Customer:  <INPUT NAME=\"CI\" SIZE=4
maxlength=4>  Name:
Credit:      %%Disc:\r\n\
Order Number:           Number of Lines:
W_tax:          D_tax:\r\n\r\n\
 Supp_W  Item_Id  Item Name                Qty
Stock  B/G  Price    Amount\r\n\
  <INPUT NAME=\"OS01\" SIZE=5
maxlength=6> <INPUT NAME=\"OI01\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ01\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS02\" SIZE=5
maxlength=6> <INPUT NAME=\"OI02\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ02\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS03\" SIZE=5
maxlength=6> <INPUT NAME=\"OI03\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ03\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS04\" SIZE=5
maxlength=6> <INPUT NAME=\"OI04\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ04\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS05\" SIZE=5
maxlength=6> <INPUT NAME=\"OI05\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ05\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS06\" SIZE=5
maxlength=6> <INPUT NAME=\"OI06\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ06\" SIZE=2 maxlength=2>\r\n"

#define in_newpage2 "\
  <INPUT NAME=\"OS07\" SIZE=5
maxlength=6> <INPUT NAME=\"OI07\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ07\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS08\" SIZE=5
maxlength=6> <INPUT NAME=\"OI08\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ08\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS09\" SIZE=5
maxlength=6> <INPUT NAME=\"OI09\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ09\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS10\" SIZE=5
maxlength=6> <INPUT NAME=\"OI10\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ10\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS11\" SIZE=5
maxlength=6> <INPUT NAME=\"OI11\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ11\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS12\" SIZE=5
maxlength=6> <INPUT NAME=\"OI12\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ12\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS13\" SIZE=5
maxlength=6> <INPUT NAME=\"OI13\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ13\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS14\" SIZE=5
maxlength=6> <INPUT NAME=\"OI14\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ14\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS15\" SIZE=5
maxlength=6> <INPUT NAME=\"OI15\" SIZE=6
maxlength=6>                         <INPUT
NAME=\"OQ15\" SIZE=2 maxlength=2>\r\n\
Execution Status:
Total:  $\r\n\
</PRE><INPUT
TYPE=\"submit\"></FORM></BODY></HTML>"


/* --------------------------
    orderstatus page
 * --------------------------*/
#define in_odrpage "\
<HTML><HEAD><TITLE>TPC-C: Order-
Status</TITLE></HEAD>\r\n\
<BODY><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"O\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\"
VALUE=%d>\r\n\
<center>Order-Status<br></center>\r\n\
<font size=4><PRE>Warehouse:%6d  District:
<INPUT NAME=\"D\" SIZE=2 maxlength=2>\r\n\
Customer: <INPUT NAME=\"CI\" SIZE=4
maxlength=4>  Name:                  <INPUT
NAME=\"CL\" SIZE=17 maxlength=16>\r\n\
Cust-Balance: \r\n\
\r\n\
Order-Number:           Entry-Date:
Carrier-Number:\r\n\
Supply-W     Item-Id    Qty     Amount
Delivery-Date\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\
\r\n\
\r\n\
</PRE>\r\n\
<INPUT
TYPE=\"submit\"></FORM></BODY></HTML>"

#define in_odrpage2 "\
<HTML><HEAD><TITLE>TPC-C: Order-
Status</TITLE></HEAD>\r\n\
<BODY><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"O\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\"
VALUE=%d>\r\n\
<center>Order Status<br></center>\r\n\
<font size=3><PRE>\
Warehouse:%6d       District:<INPUT
NAME=\"D\" SIZE=2 maxlength=2>\r\n\
Customer:<INPUT NAME=\"CI\" SIZE=4
maxlength=4>   Name:                 <INPUT
NAME=\"CL\" SIZE=16 maxlength=16>\r\n\
Cust-Balance: \r\n\
\r\n\
Order-Number:            Entry-Date:
Carrier-Number:\r\n\
Supply-W     Item-Id     Qty     Amount
Delivery-Date\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\
\r\n\
\r\n\
</PRE>\r\n\
<INPUT
TYPE=\"submit\"></FORM></BODY></HTML>\r
\n"

/* --------------------------
    payment page
 * --------------------------*/
#define in_paypage "\
<HTML><HEAD><TITLE>TPC-C:
Payment</TITLE></HEAD>\r\n\
<BODY><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"P\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\"
VALUE=%d>\r\n\
<center>Payment<br></center>\r\n\
<font size=4><PRE>Date:\r\n\
\r\n\
Warehouse:%6d
District:<INPUT NAME=\"D\" SIZE=2
maxlength=2>\r\n\
\r\n\
\r\n\
\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>\
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 tuxierr "\
<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 acoording 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_I_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
    fprintf (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 occured. [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
proccessing error [IRRECERR] occurered.\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
OBJS = 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) : $(OBJS)
   $(AR) $(ARFLAGS) $(ARCH_LIB) $(OBJS)

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

$(OBJS) : $(INCFILE)

clean:
#    rm  $(ARCH_LIB) $(OBJS)



::::::::::::::::::::::::::::::::
 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------------------
//

//
// ---------------------------
//

#ifdef SCRTEST

char *get_datetimestr( 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,
"1234567890123456");
   bp->payout.c_balance = 67876;
   bp->payout.c_credit_lim = 77777;
   strcpy( bp->payout.c_since, "11-10-2002" );

   strcpy( bp->payout.c_data,
"Migyamigyamigyamigyamigya"
         "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

//
// ------------------------------------
//

#ifdef 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-
>ol_i_id[i]);
            fprintf(test_fp,"i_name=%s\n",&bp-
>i_name[i][0]);
            fprintf(test_fp,"o_quant=%d ",bp-
>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-
>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 proccesses 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_leng;
    int    h_del2_leng;
    int    h_del3_leng;

    THREAD_CNTL_INFO*    ThreadCntlInfo;

    //SvrAPL return value
#ifndef SCRTEST
    int                  ret_val;
#endif

    MAC_PutFncEntryLog("Delivery");

    /* Create execution environment */
    ThreadCntlInfo = GetThreadCntl();
    if (ThreadCntlInfo == 0) {
        sprintf( S_WORK, "thread contorl
information is not allocated [DEL]\n");
        MAC_errHTML( s_buf, S_WORK, cookie );
        TpccUserLog (LOG_ERR, S_WORK);
        return (-1);
    }
    bp = (DeliveryData*)ThreadCntlInfo->TrxData;
    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);

#ifndef SCRTEST

/* Replaced 2003.12.15 Transaction processeing
interface COM+ --> TUXEDO */
#ifndef 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 ( tperrno ) {
        case TPELIMIT:  /* ------------------------------
---- */
        case TPETIME:   /* ------------------------------
*/
        case TPGOTSIG:  /* ------------------ */
            /* Because it is an executable again error,
               processing is executed again. */
            TpccUserLog (LOG_WRN, "Delivery
retry reason by terrno=%d\n", tperrno);
            goto resend_delivery;
            break;

        default:
            /* The error which was not able to be
executed again occurred */
            sprintf( S_WORK, "tpacall failed in
Delivery: tperrno = %d\n"
                  " svc = '%s' carrier = %d\n", tperrno,
                  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 proccesses 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_S];

#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
#ifndef SCRTEST
    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 ) {
```

```
      TpccUserLog (LOG_ERR, "Input data error
[NEW] (c_id = %s)[Retuen_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 ){
         TpccUserLog (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 ) {
      TpccUserLog (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 SCRTEST
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->TrxData,
sizeof(NewOrderData),
                  (char**)&ThreadCntlInfo-
>TrxData, &olen, 0|TPNOTIME);
   bp = ( NewOrderData * )ThreadCntlInfo-
>TrxData;
   ret_value = CreateTranErrReason(ret_val, bp-
>newout.terror, &tran_errmsg);

   switch(ret_value) {
   case 0:
      /* Success */
      break;

   case 1:
      /* Retry NewOrder transaction */
      TpccUserLog (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 );
      TpccUserLog (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 d_id = %d c_id = %d
lines = %d cookie = %d\n",
            tperrno, ret_value,
            bp->newin.d_id, bp->newin.c_id,
ol_cnt, cookie );

      MAC_errHTML_TUXEDO( s_buf, S_WORK,
cookie );
      TpccUserLog (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
   strcpy(S_WORK, h_new2);
   h_new2_leng = leng_h_new2;
/* Replaced end */

   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) {
         TpccUserLog (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 "odrpage.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 proccesses 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
#ifndef SCRTEST
   int       ret_value;
   int       ret_val;
   char*    tran_errmsg;
#endif

   THREAD_CNTL_INFO*    ThreadCntlInfo;

   MAC_PutFncEntryLog("OrderStatus");

   ThreadCntlInfo = GetThreadCntl();
```

```
    if (ThreadCntlInfo == 0) {
        sprintf( S_WORK, "thread contorl
information is not allocated [ODR]\n");
        MAC_errHTML( s_buf, S_WORK, cookie );
        TpccUserLog (LOG_ERR, S_WORK);
        return (-1);
    }
    bp = ( OrderStatusData * )ThreadCntlInfo-
>TrxData;
    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 ) {
        TpccUserLog (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) {
            TpccUserLog (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) {
        TpccUserLog (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 */
#ifndef SCRTEST
resend_orderstatus:

 /* Replaced 2003.12.15 Transaction
processeing interface COM+ --> TUXEDO */
#ifndef TRNS_BIND
    /* Set transaction type for Warehouse bind */
    bp->retval = 3;
#endif

    ret_val = tpcall(svr_name,
(char*)ThreadCntlInfo->TrxData,
sizeof(NewOrderData),
                    (char**)&ThreadCntlInfo-
>TrxData, &olen, 0|TPNOTIME);
    bp = ( OrderStatusData * )ThreadCntlInfo-
>TrxData;
    ret_value = CreateTranErrReason(ret_val, bp-
>ordout.terror, &tran_errmsg);

    switch(ret_value) {
    case 0:
        /* Success */
        break;

    case 1:
        /* Retry OrderStatus transaction */
        TpccUserLog (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 );
        TpccUserLog (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 d_id = %d c_id = %d
c_last = '%s' cookie = %d\n",
                tperrno, ret_value, bp->ordin.d_id, bp-
>ordin.c_id,
                bp->ordin.c_last, cookie );

        MAC_errHTML_TUXEDO( s_buf, S_WORK,
cookie );
        TpccUserLog (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( strncmp( bp->ordout.ol_delivery_d[i],
"NOT DELIVR", 10 ) != 0 ){

            date2str ((S_WORK+i*0x3a+orderp[14]),
bp->ordout.ol_delivery_d[i]);
        }
    }

/* ------- The execuction 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 proccesses 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_S];

    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
flowting 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 */
#ifndef SCRTEST
resend_payment:

/* Replaced 2003.12.15 Transaction processeing
interface COM+ --> TUXEDO */
#ifndef TRNS_BIND
   /* Set transaction type for Warehouse bind */
   bp->retval = 2;
#endif

   ret_val = tpcall(svr_name,
(char*)ThreadCntlInfo->TrxData,
sizeof(NewOrderData),
                     (char**)&ThreadCntlInfo-
>TrxData, &olen, 0|TPNOTIME);
   bp = ( PaymentData * )ThreadCntlInfo-
>TrxData;
   ret_value = CreateTranErrReason(ret_val, bp-
>payout.terror, &tran_errmsg);

   switch(ret_value) {
   case 0:
      /* Success */
      break;

   case 1:
      /* Retry Paymant 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.(tperrno=%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",
            tperrno, 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 (buffer3, "\r\n");
!   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;
/* Repraced 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;

/* Repalced 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 40.03.10 */
    /*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]);

/* Repleced 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\r\n\r\n");*/

        memcpy(buffer3, "\r\n\r\n\r\n\r\n\r\n", 8+1);
        next_pos = 8;
/* 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 execuction 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 proccesses 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
#ifndef SCRTEST
    int       ret_value;
    int       ret_val;
    char*    tran_errmsg;
#endif

    THREAD_CNTL_INFO*    ThreadCntlInfo;

    MAC_PutFncEntryLog("StockLevel");

    ThreadCntlInfo = GetThreadCntl();
    if (ThreadCntlInfo == 0) {
        sprintf( S_WORK, "thread contorl
information is not allocated [STO]\n");
        MAC_errHTML( s_buf, S_WORK, cookie );
        TpccUserLog (LOG_ERR, S_WORK);
        return (-1);
    }
    bp = ( StockLevelData * )ThreadCntlInfo-
>TrxData;
    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) {
```

```
      TpccUserLog (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 */
#ifndef SCRTEST
resend_stock:

/* Replaced 2003.12.15 Transaction processeing
interface COM+ --> TUXEDO */
#ifndef TRNS_BIND
   /* Set transaction type for Warehouse bind */
   bp->retval = 5;
#endif

   ret_val = tpcall(svr_name,
(char*)ThreadCntlInfo->TrxData,
sizeof(NewOrderData),
                  (char**)&ThreadCntlInfo-
>TrxData, &olen, 0|TPNOTIME);
   bp = ( StockLevelData * )ThreadCntlInfo-
>TrxData;
   ret_value = CreateTranErrReason(ret_val, bp-
>stoout.terror, &tran_errmsg);

   switch(ret_value) {
   case 0:
      /* Success */
      break;

   case 1:
      /* Retry Paymant transaction */
      TpccUserLog (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 );
      TpccUserLog (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 );
      TpccUserLog (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-
>stoout.low_stock);

/* ------- The execuittion 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)
TpccUserLog(LOG_INF, ">>>>> "func" start
>>>>>");
#else
#define    MAC_PutFncEntryLog(func)    ;
#endif

/* Function exit point log    */
#ifdef    PUT_FNC_EXIT_LOG
#define    MAC_PutFncExitLog(func)
TpccUserLog(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;
```

```
::::::::::::::::::::::::::::::::
 blocks/tkvcinin.sql
::::::::::::::::::::::::::::::::

-- The initnew package for storing variables used
in the
-- New Order anonymous block

CREATE OR REPLACE PACKAGE inittpcc
AS
 TYPE intarray IS TABLE OF INTEGER INDEX
BY BINARY_INTEGER;
 TYPE distarray IS TABLE OF VARCHAR(24)
INDEX BY BINARY_INTEGER;
 nulldate      DATE;
 TYPE rowidarray IS TABLE OF ROWID INDEX
BY PLS_INTEGER;
 s_dist          distarray;
 idx1arr       intarray;
 s_remote       intarray;
 dist             intarray;
 row_id           rowidarray;
 cust_rowid       rowid;
 dist_name        VARCHAR2(11);
 ware_name        VARCHAR2(11);
 c_num            PLS_INTEGER;

 PROCEDURE init_no(idxarr intarray);
 PROCEDURE init_del;
 PROCEDURE init_pay;
END inittpcc;
/
show errors;

CREATE OR REPLACE PACKAGE BODY
inittpcc AS
  PROCEDURE init_no (idxarr  intarray)
  IS
  BEGIN
       -- initialize null date
   nulldate := TO_DATE('01-01-1811', 'MM-DD-
YYYY');
    idx1arr := idxarr;
  END init_no;

  PROCEDURE init_del
  IS
  BEGIN
    FOR i IN 1 .. 10 LOOP
      dist(i) := i;
    END LOOP;
  END init_del;

  PROCEDURE init_pay IS
  BEGIN
    NULL;
  END init_pay;

END inittpcc;
/
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 = inittpcc.dist(d)
         AND no_w_id = :w_id
         AND no_o_id = (select min (no_o_id)
                        from nord
                        where no_d_id = N.no_d_id
                              and no_w_id = N.no_w_id)
       RETURNING no_d_id, no_o_id BULK
COLLECT INTO :d_id, :order_id;

    :ordcnt := SQL%ROWCOUNT;

    FORALL o in 1.. :ordcnt
      UPDATE ordr SET o_carrier_id = :carrier_id
      WHERE o_id = :order_id (o)
        AND o_d_id = :d_id(o)
        AND o_w_id = :w_id
      RETURNING o_c_id BULK COLLECT
INTO :o_c_id;

    FORALL o in 1.. :ordcnt
      UPDATE ordl SET ol_delivery_d = :now
      WHERE ol_w_id = :w_id
        AND ol_d_id = :d_id(o)
        AND ol_o_id = :order_id(o)
      RETURNING sum(ol_amount) BULK
COLLECT INTO  :sums;

    FORALL c IN 1.. :ordcnt
      UPDATE cust
        SET c_balance = c_balance + :sums(c),
                    c_delivery_cnt = c_delivery_cnt +
1
       WHERE c_w_id = :w_id
         AND c_d_id = :d_id(c)
         AND c_id = :o_c_id(c);
    COMMIT;
    EXIT;
    EXCEPTION
      WHEN not_serializable OR deadlock OR
snapshot_too_old
      THEN
        ROLLBACK;
        :retry := :retry + 1;
    END;

  END LOOP; -- for retry
END;


::::::::::::::::::::::::::::::::
 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,
inittpcc.s_dist,
                    :ol_amount,:brand_generic;
   END u1;

   PROCEDURE u2 IS
   BEGIN
       FORALL idx IN 1 .. cache_ol_cnt
         UPDATE  stock_item
         SET s_order_cnt = s_order_cnt + 1,
         s_ytd = s_ytd + :ol_quantity(idx),
         s_remote_cnt = s_remote_cnt
+ :s_remote(idx),
         s_quantity = (CASE WHEN s_quantity
< :ol_quantity (idx) + 10
                       THEN s_quantity +91
                       ELSE s_quantity
                  END) - :ol_quantity(idx)
         WHERE i_id = :ol_i_id(idx)
         AND s_w_id = :ol_supply_w_id(idx)
         RETURNING i_price, i_name, s_quantity,
s_dist_02,
                   i_price*:ol_quantity(idx),
          CASE WHEN i_data NOT LIKE
'%ORIGINAL%'
               THEN 'G'
               ELSE (CASE WHEN s_data NOT
LIKE '%ORIGINAL%'
                     THEN 'G'
                     ELSE 'B'
                     END)
          END
       BULK COLLECT
INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                    :ol_amount,:brand_generic;
   END u2;
```

```
  PROCEDURE u3 IS
  BEGIN
      FORALL idx IN 1 .. cache_ol_cnt
        UPDATE  stock_item
        SET s_order_cnt = s_order_cnt + 1,
        s_ytd = s_ytd + :ol_quantity(idx),
        s_remote_cnt = s_remote_cnt
+ :s_remote(idx),
        s_quantity = (CASE WHEN s_quantity
< :ol_quantity (idx) + 10
                      THEN s_quantity +91
                      ELSE s_quantity
                    END) - :ol_quantity(idx)
        WHERE i_id = :ol_i_id(idx)
        AND s_w_id = :ol_supply_w_id(idx)
        RETURNING i_price, i_name, s_quantity,
s_dist_03,
                  i_price*:ol_quantity(idx),
          CASE WHEN i_data NOT LIKE
'%ORIGINAL%'
               THEN 'G'
               ELSE (CASE WHEN s_data NOT
LIKE '%ORIGINAL%'
                     THEN 'G'
                     ELSE 'B'
                     END)
          END
       BULK COLLECT
INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                    :ol_amount,:brand_generic;
  END u3;

  PROCEDURE u4 IS
  BEGIN
      FORALL idx IN 1 .. cache_ol_cnt
        UPDATE  stock_item
        SET s_order_cnt = s_order_cnt + 1,
        s_ytd = s_ytd + :ol_quantity(idx),
        s_remote_cnt = s_remote_cnt
+ :s_remote(idx),
        s_quantity = (CASE WHEN s_quantity
< :ol_quantity (idx) + 10
                      THEN s_quantity +91
                      ELSE s_quantity
                    END) - :ol_quantity(idx)
        WHERE i_id = :ol_i_id(idx)
        AND s_w_id = :ol_supply_w_id(idx)
        RETURNING i_price, i_name, s_quantity,
s_dist_04,
                  i_price*:ol_quantity(idx),
          CASE WHEN i_data NOT LIKE
'%ORIGINAL%'
               THEN 'G'
               ELSE (CASE WHEN s_data NOT
LIKE '%ORIGINAL%'
                     THEN 'G'
                     ELSE 'B'
                     END)
          END
       BULK COLLECT
INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                    :ol_amount,:brand_generic;
  END u4;

  PROCEDURE u5 IS
  BEGIN
      FORALL idx IN 1 .. cache_ol_cnt
        UPDATE  stock_item
        SET s_order_cnt = s_order_cnt + 1,
        s_ytd = s_ytd + :ol_quantity(idx),
        s_remote_cnt = s_remote_cnt
+ :s_remote(idx),
        s_quantity = (CASE WHEN s_quantity
< :ol_quantity (idx) + 10
                      THEN s_quantity +91
                      ELSE s_quantity
                    END) - :ol_quantity(idx)
        WHERE i_id = :ol_i_id(idx)
        AND s_w_id = :ol_supply_w_id(idx)
        RETURNING i_price, i_name, s_quantity,
s_dist_05,
                  i_price*:ol_quantity(idx),
          CASE WHEN i_data NOT LIKE
'%ORIGINAL%'
               THEN 'G'
               ELSE (CASE WHEN s_data NOT
LIKE '%ORIGINAL%'
                     THEN 'G'
                     ELSE 'B'
                     END)
          END
       BULK COLLECT
INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                    :ol_amount,:brand_generic;
  END u5;

  PROCEDURE u6 IS
  BEGIN
      FORALL idx IN 1 .. cache_ol_cnt
        UPDATE  stock_item
        SET s_order_cnt = s_order_cnt + 1,
        s_ytd = s_ytd + :ol_quantity(idx),
        s_remote_cnt = s_remote_cnt
+ :s_remote(idx),
        s_quantity = (CASE WHEN s_quantity
< :ol_quantity (idx) + 10
                      THEN s_quantity +91
                      ELSE s_quantity
                    END) - :ol_quantity(idx)
        WHERE i_id = :ol_i_id(idx)
        AND s_w_id = :ol_supply_w_id(idx)
        RETURNING i_price, i_name, s_quantity,
s_dist_06,
                  i_price*:ol_quantity(idx),
          CASE WHEN i_data NOT LIKE
'%ORIGINAL%'
               THEN 'G'
               ELSE (CASE WHEN s_data NOT
LIKE '%ORIGINAL%'
                     THEN 'G'
                     ELSE 'B'
                     END)
          END
       BULK COLLECT
INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                    :ol_amount,:brand_generic;
  END u6;

  PROCEDURE u7 IS
  BEGIN
      FORALL idx IN 1 .. cache_ol_cnt
        UPDATE  stock_item
        SET s_order_cnt = s_order_cnt + 1,
        s_ytd = s_ytd + :ol_quantity(idx),
        s_remote_cnt = s_remote_cnt
+ :s_remote(idx),
        s_quantity = (CASE WHEN s_quantity
< :ol_quantity (idx) + 10
                      THEN s_quantity +91
                      ELSE s_quantity
                    END) - :ol_quantity(idx)
        WHERE i_id = :ol_i_id(idx)
        AND s_w_id = :ol_supply_w_id(idx)
        RETURNING i_price, i_name, s_quantity,
s_dist_07,
                  i_price*:ol_quantity(idx),
          CASE WHEN i_data NOT LIKE
'%ORIGINAL%'
               THEN 'G'
               ELSE (CASE WHEN s_data NOT
LIKE '%ORIGINAL%'
                     THEN 'G'
                     ELSE 'B'
                     END)
          END
       BULK COLLECT
INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                    :ol_amount,:brand_generic;
  END u7;

  PROCEDURE u8 IS
  BEGIN
      FORALL idx IN 1 .. cache_ol_cnt
        UPDATE  stock_item
        SET s_order_cnt = s_order_cnt + 1,
        s_ytd = s_ytd + :ol_quantity(idx),
        s_remote_cnt = s_remote_cnt
+ :s_remote(idx),
        s_quantity = (CASE WHEN s_quantity
< :ol_quantity (idx) + 10
                      THEN s_quantity +91
                      ELSE s_quantity
                    END) - :ol_quantity(idx)
        WHERE i_id = :ol_i_id(idx)
        AND s_w_id = :ol_supply_w_id(idx)
        RETURNING i_price, i_name, s_quantity,
s_dist_08,
                  i_price*:ol_quantity(idx),
          CASE WHEN i_data NOT LIKE
'%ORIGINAL%'
               THEN 'G'
               ELSE (CASE WHEN s_data NOT
LIKE '%ORIGINAL%'
                     THEN 'G'
                     ELSE 'B'
                     END)
          END
       BULK COLLECT
INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                    :ol_amount,:brand_generic;
  END u8;

  PROCEDURE u9 IS
  BEGIN
      FORALL idx IN 1 .. cache_ol_cnt
        UPDATE  stock_item
        SET s_order_cnt = s_order_cnt + 1,
        s_ytd = s_ytd + :ol_quantity(idx),
        s_remote_cnt = s_remote_cnt
+ :s_remote(idx),
        s_quantity = (CASE WHEN s_quantity
< :ol_quantity (idx) + 10
                      THEN s_quantity +91
                      ELSE s_quantity
                    END) - :ol_quantity(idx)
        WHERE i_id = :ol_i_id(idx)
        AND s_w_id = :ol_supply_w_id(idx)
        RETURNING i_price, i_name, s_quantity,
s_dist_09,
                  i_price*:ol_quantity(idx),
          CASE WHEN i_data NOT LIKE
'%ORIGINAL%'
               THEN 'G'
               ELSE (CASE WHEN s_data NOT
LIKE '%ORIGINAL%'
                     THEN 'G'
                     ELSE 'B'
                     END)
```

```
        END
      BULK COLLECT
INTO :i_price, :i_name, :s_quantity,
inittpcc.s_dist,
                  :ol_amount,:brand_generic;
  END u9;

  PROCEDURE u10 IS
  BEGIN
     FORALL idx IN 1 .. cache_ol_cnt
       UPDATE  stock_item
       SET s_order_cnt = s_order_cnt + 1,
       s_ytd = s_ytd + :ol_quantity(idx),
       s_remote_cnt = s_remote_cnt
+ :s_remote(idx),
       s_quantity = (CASE WHEN s_quantity
< :ol_quantity (idx) + 10
                    THEN s_quantity +91
                    ELSE s_quantity
                END) - :ol_quantity(idx)
       WHERE i_id = :ol_i_id(idx)
       AND s_w_id = :ol_supply_w_id(idx)
       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 ordr  (o_id,o_d_id, o_w_id,
o_c_id, o_entry_d,
                    o_carrier_id, o_ol_cnt,
o_all_local)
        VALUES (:o_id, :d_id, :w_id, :c_id,
              :cr_date, 11, :o_ol_cnt, :o_all_local);


       dummy_local :=  :d_id;

       IF (dummy_local < 6) THEN
         IF (dummy_local < 3) THEN
           IF (dummy_local = 1) THEN
             u1;
           ELSE
             u2;
           END IF;
         ELSE
           IF (dummy_local = 3) THEN
             u3;
           ELSIF (dummy_local = 4) then
             u4;
           ELSE
             u5;
           END IF;
         END IF;
       ELSE
         IF (dummy_local < 8) THEN
           IF (dummy_local = 6) THEN
             u6;
           ELSE
             u7;
           END IF;
         ELSE
           IF (dummy_local = 8) THEN
             u8;
           ELSIF (dummy_local = 9) then
             u9;
           ELSE
             u10;
           END IF;
         END IF;
       END IF;

       dummy_local := sql%rowcount;

       IF (dummy_local != cache_ol_cnt )  THEN
fix_items; END IF;

       FORALL idx IN 1..dummy_local
        INSERT INTO ordl
          (ol_o_id, ol_d_id, ol_w_id, ol_number,
ol_delivery_d, ol_i_id,
             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
cogiguration file.   *
```

```
 * Return Value                                    *
 *     Get string length                           *

*************************************************************
****/
int    GetPrivateProfileString(char*
section_name,    /* Section name
*/
                        char*    key_name,       /* Key
name                        */
                        char*    delault_str,    /* Default
string, if kye 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 DEFALT_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 DEFALT_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 DEFALT_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 DEFALT_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);

DEFALT_STRING_FCLOSE:
    fclose(prof_file);

DEFALT_STRING:
    strncpy(key_data, delault_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
COMOBJS    = log.o sema.o
GetPrivateProfileString.o shmem.o
COMLIB  = libcom.a

INCFILES = log.h sema.h forlinux.h shmem.h

$(COMLIB) : $(COMOBJS)
    $(AR) $(ARFLAGS) $(COMLIB) $(COMOBJS)

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

$(COMOBJS) : $(INCFILES)

clean:
    rm  $(COMLIB) $(COMOBJS)
```

```
::::::::::::::::::::::::::::::::
 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 TpccUserLog(char* file_name, int line_no,
char* type_name, char* fmtp, ...)
{
   FILE*       fp;
   pid_t       pid;
   pthread_t   tId;
   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. */
   tId = pthread_self();

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

   va_start(va, fmtp);

   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)tId, fname, line_no,
type_name);
   vfprintf(fp, fmtp, va);

   if (*(fmtp + strlen(fmtp) - 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 TpccUserLog (char *file_name, int line_no,
char* type_name, char* fmtp, ...);

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;

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

    if ((sid = GetSem(path, projectId)) == -1) {
       TpccUserLog(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_PERFOREMANCE
(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 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 ((shmid=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_PERFOREMANCE
(int)'P'


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



::::::::::::::::::::::::::::::::
```

```
 svrapl/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 "TrnCntlInfo.h"

char    GLB_LogFilePath[MAX_PATH];
char    GLB_ConfigFilePath[MAX_PATH];
int       GLB_LogSemId;

/* Global area for Oracle interfase. */
/* ------------------------------ */
/*    Delivery (pldel.cpp)      */
/* ------------------------------ */
pldelctx *pldctx;
delctx *dctx;
#ifdef DMLRETDEL
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)        */
/* ------------------------------ */
paydtx *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_quanti10[15];
int nol_quanti91[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;


#ifndef AVOID_DEADLOCK
int indx[NITEMS], ordl_cnt;
#endif



::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::
 svrapl/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 "TrnCntlInfo.h"



extern char       GLB_LogFilePath[MAX_PATH];
extern char
GLB_ConfigFilePath[MAX_PATH];
extern int        GLB_LogSemId;


#define    TPCC_CONF_FILE
"/home/tpc/conf/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_ORA8I
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 OCIError *errhp;
extern OCISvcCtx *tpcsvc;
extern OCISession *tpcusr;
extern OCIStmt *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_quanti10[15];
extern int nol_quanti91[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 *xmem;
/* ---------------------------- */
/*    (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;



#ifndef AVOID_DEADLOCK
int indx[NITEMS], ordl_cnt;
#endif



::::::::::::::::::::::::::::::::::
 svrapl/MakeShell
::::::::::::::::::::::::::::::::::

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


::::::::::::::::::::::::::::::::::
 svrapl/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/svrapl
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
COMOBJS = tpccsvr.o GlobalArea.o
initsvrconfig.o
ALLOBJS =  $(COMOBJS) $(MAIN_WHBOBJ)
$(MAIN_NEWOBJ) $(MAIN_PAYOBJ)
$(MAIN_DELOBJ) \
        $(MAIN_STOOBJ) $(MAIN_ORDOBJ)
3TIERLIB = $(3TIERDIR)/libtier.a
COMLIB   = $(COMDIR)/libcom.a

# depend on include file.
INCFILE = $(SVRDIR)/tpcc.h
$(SVRDIR)/GlobalArea.h $(SVRDIR)/prototype.h
\
        $(SVRDIR)/tpccflags.h
$(SVRDIR)/tpcc_info.h $(SVRDIR)/TrnCntlInfo.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/libbuft.a $(TUXDIR)/lib/libfml.a \
        $(TUXDIR)/lib/libfml32.a
$(TUXDIR)/lib/libengine.a -lpthread -ldl
#TUXLIBS    = -L$(TUXDIR)/lib/ -ltux -lbuft -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_tpccfmld
TARGET_STO_3TIER       =
$(OBJDIR)/3tier_tpccfmls
TARGET_ORD_3TIER       =
$(OBJDIR)/3tier_tpccfmlo

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/ \
#    $(ORACLE_HOME)/rdbms/lib/defopt.o  -
lclntsh \
#    -ldl -lm -lpthread -lnsl
LDFLAGS=-L$(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 -lpthread -lnsl


$(TARGETS) : $(ALLOBJS) $(3TIERLIB)
$(COMLIB)
    $(LD) -o $(TARGET_WHB_3TIER)
$(MAIN_WHBOBJ) $(COMOBJS) $(COMLIB)
$(3TIERLIB) $(TUXLIBS) $(ORALIBS)
$(LDFLAGS)
    $(LD) -o $(TARGET_NEW_3TIER)
$(MAIN_NEWOBJ) $(COMOBJS) $(COMLIB)
$(3TIERLIB) $(TUXLIBS) $(ORALIBS)
$(LDFLAGS)
    $(LD) -o $(TARGET_PAY_3TIER)
$(MAIN_PAYOBJ) $(COMOBJS) $(COMLIB)
$(3TIERLIB) $(TUXLIBS) $(ORALIBS)
$(LDFLAGS)
    $(LD) -o $(TARGET_DEL_3TIER)
$(MAIN_DELOBJ) $(COMOBJS) $(COMLIB)
$(3TIERLIB) $(TUXLIBS) $(ORALIBS)
$(LDFLAGS)
    $(LD) -o $(TARGET_STO_3TIER)
$(MAIN_DELOBJ) $(COMOBJS) $(COMLIB)
$(3TIERLIB) $(TUXLIBS) $(ORALIBS)
$(LDFLAGS)
    $(LD) -o $(TARGET_ORD_3TIER)
$(MAIN_ORDOBJ) $(COMOBJS) $(COMLIB)
$(3TIERLIB) $(TUXLIBS) $(ORALIBS)
$(LDFLAGS)

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

$(ALLOBJS) : $(INCFILE)
$(ALLOBJS) : Makefile

clean:
    rm  $(ALLOBJS) $(TARGETS)



::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::
 svrapl/TrnCntlInfo.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];
    OCIStmt *curd0;
    OCIStmt *curd1;
    OCIStmt *curd2;
    OCIStmt *curd3;
    OCIStmt *curd4;
    OCIStmt *curd5;
    OCIStmt *curd6;
    OCIStmt *curdtest;

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

```
  OCIDefine *del_o_id_dp;
  OCIDefine *no_rowid_dp;
  OCIDefine *c_id_dp;
  OCIDefine *o_rowid_dp;
  OCIDefine *cons_dp;
  OCIDefine *amt_dp;

  int norow;
};

typedef struct delctx delctx;
struct pldelctx {

  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];*/
!#ifdef 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;
  OCIStmt *curp1;
  OCIStmt *curp2;
  OCIBind *w_id_bp;
  OCIBind *d_id_bp;
  OCIBind *o_id_bp;
  OCIBind *o_c_id_bp;
  OCIBind *ordcnt_bp;
  OCIBind *sums_bp;
  OCIBind *del_date_bp;
  OCIBind *carrier_id_bp;
  OCIBind *retry_bp;

  int norow;

};
typedef struct pldelctx pldelctx;


#ifdef DMLRETDEL
struct amtctx {
  int ol_amt[NITEMS];
  sb2 ol_amt_ind[NITEMS];
  ub4 ol_amt_len[NITEMS];
  ub2 ol_amt_rcode[NITEMS];
  int ol_cnt;
};
typedef struct amtctx amtctx;
#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];
  OCIStmt *curn1;
  OCIBind *ol_i_id_bp;
  OCIBind *ol_supply_w_id_bp;
  OCIBind *i_price_bp;
  OCIBind *i_name_bp;
  OCIBind *s_bg_bp;
  ub4 nol_i_count;
  ub4 nol_s_count;
  ub4 nol_q_count;
  ub4 nol_item_count;
  ub4 nol_name_count;
  ub4 nol_qty_count;
  ub4 nol_bg_count;
  ub4 nol_am_count;
  ub4 s_remote_count;
  OCIStmt *curn2;
  OCIBind *ol_quantity_bp;
  OCIBind *s_remote_bp;
  OCIBind *s_quantity_bp;
  OCIBind *w_id_bp;
  OCIBind *d_id_bp;
  OCIBind *c_id_bp;
  OCIBind *o_all_local_bp;
  OCIBind *o_all_cnt_bp;
  OCIBind *w_tax_bp;
  OCIBind *d_tax_bp;
  OCIBind *o_id_bp;
  OCIBind *c_discount_bp;
  OCIBind *c_credit_bp;
  OCIBind *c_last_bp;
  OCIBind *retries_bp;
  OCIBind *cr_date_bp;
  OCIBind *ol_o_id_bp;
  OCIBind *ol_amount_bp;

/* 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;

  OCIStmt *curo0;
  OCIStmt *curo1;
  OCIStmt *curo2;
  OCIStmt *curo3;
  OCIStmt *curo4;
  OCIBind *c_id_bp;
  OCIBind *w_id_bp[4];
  OCIBind *d_id_bp[4];
  OCIBind *c_last_bp[2];
  OCIBind *o_id_bp;
  OCIBind *c_rowid_bp;
/* 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 *ol_d_d_dp;
  OCIDefine *ol_i_id_dp;
  OCIDefine *ol_supply_w_id_dp;
  OCIDefine *ol_quantity_dp;
  OCIDefine *ol_amount_dp;
  OCIDefine *ol_d_base_dp;
  OCIDefine *c_count_dp;
  OCIRowid *c_rowid_ptr[100];
  OCIRowid *c_rowid_cust;
/* 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 stoctx {
  OCIStmt *curs;
  OCIBind *w_id_bp;
  OCIBind *d_id_bp;
  OCIBind *threshold_bp;
#ifdef PLSQLSTO
  OCIBind *low_stock_bp;
#else
  OCIDefine *low_stock_bp;
#endif
  int norow;
};

typedef struct stoctx stoctx;
```

```
::::::::::::::::::::::::::::::::::
 svrapl/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 tmdsptchtbl_t _tmdsptchtbl[] = {
//    { "OPSTUXSERVER", "OPSTUXSERVER",
(void (*) _((TPSVCINFO *))) OPSTUXSERVER,
0, 0 },
//    { NULL, NULL, NULL, 0, 0 }
//};
static struct tmdsptchtbl_t _tmdsptchtbl[] = {
    { "DELIVERY", "DELIVERY", (void (*)
_((TPSVCINFO *))) DELIVERY, 0, 0 },
    { NULL, NULL, NULL, 0, 0 }
};

#ifndef _TMDLLIMPORT
#define _TMDLLIMPORT
#endif

_TMDLLIMPORT extern struct xa_switch_t
tmnull_switch;

struct tmsvrargs_t tmsvrargs = {
    NULL,
    &_tmdsptchtbl[0],
    0,
    tpsvrinit,
    tpsvrdone,
```

```
    _tmrunserver,    /* PRIVATE  */
    NULL,            /* RESERVED */
    NULL,            /* RESERVED */
    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-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 tmdsptchtbl_t _tmdsptchtbl[] = {
//    { "OPSTUXSERVER", "OPSTUXSERVER",
(void (*) _((TPSVCINFO *))) OPSTUXSERVER,
0, 0 },
//    { NULL, NULL, NULL, 0, 0 }
//};
static struct tmdsptchtbl_t _tmdsptchtbl[] = {
    { "NEWORDER", "NEWORDER", (void (*)
_((TPSVCINFO *))) NEWORDER, 0, 0 },
    { NULL, NULL, NULL, 0, 0 }
};

#ifndef _TMDLLIMPORT
#define _TMDLLIMPORT
#endif

_TMDLLIMPORT extern struct xa_switch_t
tmnull_switch;

struct tmsvrargs_t tmsvrargs = {
    NULL,
    &_tmdsptchtbl[0],
    0,
    tpsvrinit,
    tpsvrdone,
    _tmrunserver,    /* PRIVATE  */
    NULL,            /* RESERVED */
    NULL,            /* RESERVED */
    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-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 tmdsptchtbl_t _tmdsptchtbl[] = {
//    { "OPSTUXSERVER", "OPSTUXSERVER",
(void (*) _((TPSVCINFO *))) OPSTUXSERVER,
0, 0 },
//    { NULL, NULL, NULL, 0, 0 }
//};
static struct tmdsptchtbl_t _tmdsptchtbl[] = {
    { "ORDERSTATUS", "ORDERSTATUS", (void
(*) _((TPSVCINFO *))) ORDERSTATUS, 0, 0 },
    { NULL, NULL, NULL, 0, 0 }
};

#ifndef _TMDLLIMPORT
#define _TMDLLIMPORT
#endif

_TMDLLIMPORT extern struct xa_switch_t
tmnull_switch;

struct tmsvrargs_t tmsvrargs = {
    NULL,
    &_tmdsptchtbl[0],
    0,
    tpsvrinit,
    tpsvrdone,
    _tmrunserver,    /* PRIVATE  */
    NULL,            /* RESERVED */
    NULL,            /* RESERVED */
    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-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 tmdsptchtbl_t _tmdsptchtbl[] = {
//    { "OPSTUXSERVER", "OPSTUXSERVER",
(void (*) _((TPSVCINFO *))) OPSTUXSERVER,
0, 0 },
```

```
//    { NULL, NULL, NULL, 0, 0 }
//};
static struct tmdsptchtbl_t _tmdsptchtbl[] = {
    { "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,
    &_tmdsptchtbl[0],
    0,
    tpsvrinit,
    tpsvrdone,
    _tmrunserver,    /* PRIVATE  */
    NULL,           /* RESERVED */
    NULL,           /* RESERVED */
    NULL,           /* RESERVED */
    NULL            /* RESERVED */
};

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

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

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


::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::
 svrapl/bs-sto.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 STOCKLVL _((TPSVCINFO *));
#if defined(__cplusplus)
}
#endif

//static struct tmdsptchtbl_t _tmdsptchtbl[] = {
//    { "OPSTUXSERVER", "OPSTUXSERVER",
(void (*) _((TPSVCINFO *))) OPSTUXSERVER,
0, 0 },
//    { NULL, NULL, NULL, 0, 0 }
//};
static struct tmdsptchtbl_t _tmdsptchtbl[] = {
    { "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,
    &_tmdsptchtbl[0],
    0,
    tpsvrinit,
    tpsvrdone,
    _tmrunserver,    /* PRIVATE  */
    NULL,           /* RESERVED */
    NULL,           /* RESERVED */
    NULL,           /* RESERVED */
    NULL            /* RESERVED */
};

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

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

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


::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::
 svrapl/bs-whb.c
::::::::::::::::::::::::::::::::

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

#include <string.h>
#include "forlinux.h"
#include "log.h"

#if defined(__cplusplus)
extern "C" {
#endif
extern int _tmrunserver _((int));
extern void OPSTUXSERVER _((TPSVCINFO
*));
#if defined(__cplusplus)
}
#endif

static struct tmdsptchtbl_t _tmdsptchtbl[] = {
    { "OPSTUXSERVER", "OPSTUXSERVER",
(void (*) _((TPSVCINFO *))) OPSTUXSERVER,
0, 0 },
    { NULL, NULL, NULL, 0, 0 }
};

#ifndef _TMDLLIMPORT
#define _TMDLLIMPORT
#endif

_TMDLLIMPORT extern struct xa_switch_t
tmnull_switch;

struct tmsvrargs_t tmsvrargs = {
    NULL,
    &_tmdsptchtbl[0],
    0,
    tpsvrinit,
    tpsvrdone,
    _tmrunserver,    /* PRIVATE  */
    NULL,           /* RESERVED */
    NULL,           /* RESERVED */
    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 "shmem.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 defalt value */
        TpccUserLog(LOG_LCK, "INI file nothing,
using default value");
        strcpy(GLB_LogFilePath,
DEFAULT_SVRAPL_LOG_PATH);
        return;
    }

    /* Get execution informations
*/
    /* If undefined key and illigal 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 gettiing */
            continue;
        }
        break;
    }
#ifdef PUT_INF_LOG
    TpccUserLog(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 informaion (set to global
area) */
    strcpy(GLB_ConfigFilePath, path);

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

    GetConfFileInfo();

    TpccUserLog(LOG_LCK, "InitSvrConfig start
\n");

    /* Initialize SVRAPL semafore 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) {
        TpccUserLog(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_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)
TpccUserLog(LOG_INF, ">>>>> "func" start
>>>>>");
#else
#define    MAC_PutFncEntryLog(func)    ;
#endif

/* Function exit point log    */
#ifdef    PUT_FNC_EXIT_LOG
#define    MAC_PutFncExitLog(func)
TpccUserLog(LOG_INF, "<<<<< "func" end
<<<<<");
#else
#define    MAC_PutFncExitLog(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 tkvcdinit ();
int tkvcninit ();
int tkvcoinit ();
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    tkvcdinit (int plsqlflag);
int    tkvcninit ();
int    tkvcoinit ();
int tkvcpinit (void);
int    tkvcsinit ();
int    tkvcd (int plsqlflag);
int    tkvcn ();
int    tkvcs ();
int    tkvcp ();
int    tkvco ();
void tkvcddone (int plsqlflag);
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* fnam, 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* delault_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 ldadef;
#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 multitranx;
#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 errrpt ();
#endif
/* Deleted end */

/* Added T.Kato 2003.03.25 for debug */
extern    void DbgLog(char* form_dat, int arg);
#ifdef DGLDEF
#define    DBGLOG(format_data, arg)
TpccUserLog(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, progv,
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)),\
           (progv), (progvl),
(ftype),0,0,0,0,0,OCI_DEFAULT));



/* bind arrays for sql */
#define
OCIBNDRA(stmp,bndp,errp,sqlvar,progv,progvl,f
type,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)),\

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


/* use with callback data */
#define
OCIBNDRAD(stmp,bndp,errp,sqlvar,progvl,ftype
,indp,ctxp,\
               cbf_nodata,cbf_data) \
      DISCARD ocierror(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,progv,progvl,f
type,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)),
(dvoid*)(progv),(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,progv,progvl,ft
ype,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)),\

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

/* bind in/out for plsql arrays witout indicator and
rcode */
#define
OCIBNDPLA(stmp,bndp,errp,sqlvar,progv,progvl
,ftype,alen,ms,cu) \
      DISCARD ocierror(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 *)(progv), \

(progvl),(ftype),NULL,(alen),NULL,(ms),(cu),OCI
_DEFAULT));

/* bind in/out values for plsql with indicator and
rcode */
#define
OCIBNDRAA(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode,\
                ms,cu) \
      ocierror(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)),\

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

#define
OCIDEFINE(stmp,dfnp,errp,pos,progv,progvl,fty
pe)\

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


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

OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HT
YPE_DEFINE,0,\
                (dvoid**)0);\

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(pro
gv),(progvl),\

(ftype),NULL,NULL,NULL,OCI_DEFAULT); \


#define
OCIDFNRA(stmp,dfnp,errp,pos,progv,progvl,ftyp
e,indp,alen,arcode) \

OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HT
YPE_DEFINE,0,\
                (dvoid**)0);\

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(pro
gv),\
                (progvl),(ftype),(indp),(alen),\
                (arcode),OCI_DEFAULT);

#define
OCIDFNDYN(stmp,dfnp,errp,pos,progv,progvl,ft
ype,indp,ctxp,cbf_data) \
      ocierror(LOG_FILE_LINE,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HT
YPE_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 Overraped
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 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 plsqlflag;
!};
!
!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
#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 <tmenv.h>
  #include <xa.h>
  #include <userlog.h>

/* set up pointers for type casting */
struct newstruct *newinfo;
struct paystruct *payinfo;
struct ordstruct *ordinfo;
struct delstruct *delinfo;
struct stostruct *stoinfo;

//extern void TMlog();

#endif



#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)_vsnprintf( 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 inited */
     /* fix uid an 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 tpreturn success - let client
side poll retval for actual error
  tpreturn (TPSUCCESS, 0, (char *) payinfo,
sizeof (struct paystruct), 0);

#endif

}


/* FUNCTION: int
ORDERSTATUS(CLIENTDATA *jobData,
OrderStatusData *orddata, int deadlock)
 *
 * PURPOSE:     This function handles the new
order transaction.
 *
 * ARGUMENTS:   deadlock :   count of
deadlocks encountered during txn
 *            jobData:     pointer to entire block of
user data
 *            orddata:     pointer to datastructure in
jobData that contains the new order data
 * RETURNS:           int     TRUE    transaction
committed
 *                            FALSE   item number
not valid
 *                            -1            deadlock
max retry reached
 *
 *
 * COMMENTS:    None
 *
 */

#ifdef TOP
int ORDERSTATUS(CLIENTDATA *jobData,
OrderStatusData *orddata, int deadlock)
#else
void ORDERSTATUS (TPSVCINFO *msg)
#endif

{

#ifdef TOP
   int result;

   result = TPCord(orddata);

   return result;

#else
  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 tpreturn success - let client
side poll retval for actual error
  tpreturn (TPSUCCESS, 0, (char *) ordinfo,
sizeof (struct ordstruct), 0);


#endif

}

/* FUNCTION: int DELIVERY(CLIENTDATA
*jobData, DeliveryData *deldata, int deadlock)
 *
 * PURPOSE:     This function handles the new
order transaction.
 *
 * ARGUMENTS:   deadlock :   count of
deadlocks encountered during txn
 *            jobData:     pointer to entire block of
user data
 *            stodata:     pointer to datastructure in
jobData that contains the new order data
 * RETURNS:           int     TRUE    transaction
committed
 *                            FALSE   item number
not valid
 *                            -1            deadlock
max retry reached
 *
 *
 * COMMENTS:    None
 *
 */

#ifdef TOP
int DELIVERY(CLIENTDATA *jobData,
DeliveryData *deldata, int deadlock)
#else
void DELIVERY (TPSVCINFO *msg)
#endif

{

#ifdef TOP
   int result;

   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 tpreturn success - let client
side poll retval for actual error
  tpreturn (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
```

```
!void STOCKLEVEL (TPSVCINFO *msg)
!#endif
#endif

#ifdef 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 tpreturn success - let client
side poll retval for actual error
   tpreturn (TPSUCCESS, 0, (char *) stoinfo,
sizeof (struct stostruct), 0);


#endif
}

/* FUNCTION: int
OPSTUXSERVER(CLIENTDATA *jobData,
NewOrderData *neword, int deadlock)
 *
 * PURPOSE:     This function handles all
transactions.
 *
 * ARGUMENTS:   deadlock :  count of
deadlocks encountered during txn
 *              jobData:    pointer to entire block of
user data
 *              neword:     pointer to datastructure in
jobData that contains the new order data
 * RETURNS:             int     TRUE    transaction
committed
 *                              FALSE   item number
not valid
 *                              -1              deadlock
max retry reached
 *
 *
 * COMMENTS:    None
 *
 */
#ifdef TOP
int OPSTUXSERVER(CLIENTDATA *jobData,
NewOrderData *neword, int deadlock)
#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 tpreturn success - let client
side poll retval for actual error
!    tpreturn (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 tpreturn success - let client
side poll retval for actual error
!      tpreturn (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 tpreturn success - let client
side poll retval for actual error
!       tpreturn (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 tpreturn success
- let client side poll retval for actual error
!                      tpreturn (TPSUCCESS, 0, (char
*) delinfo, sizeof (struct delstruct), 0);
!                  }
!                  else {  // assume rest is stock level
!
!              stoinfo = (struct stostruct *) msg-
>data;
!              stoinfo->retval = TPCsto (stoinfo);  //
set return value to 0 or -1
!
!              // always return tpreturn success - let
client side poll retval for actual error
!              tpreturn (TPSUCCESS, 0, (char *)
stoinfo, sizeof (struct stostruct), 0);
!               }
#endif


   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 tpreturn success - let client
side poll retval for actual error

      tpreturn (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 tpreturn success - let client
side poll retval for actual error
      tpreturn (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 tpreturn success - let client
side poll retval for actual error
      tpreturn (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 = -I$(ORADIR)/rdbms/demo -
I$(ORADIR)/rdbms/public
COM_INC = -I$(COMDIR)
SRV_COM_INC = -I$(SVRDIR)
TUX_INC = -I$(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)/TrnCntlInfo.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 inittpcc.init_del ; END;"

#define SQLTXT1 "DELETE FROM nord
WHERE no_d_id = :d_id \
            AND no_w_id = :w_id and rownum <=
1 \
          RETURNING no_o_id into :o_id "

#define SQLTXT3 "UPDATE ordr SET
o_carrier_id = :carrier_id \
            WHERE o_id = :o_id and o_d_id
= :d_id and o_w_id = :w_id \
            returning o_c_id into :o_c_id"


#define SQLTXT4 "UPDATE ordl \
  SET ol_delivery_d = :cr_date \
  WHERE ol_w_id = :w_id AND ol_d_id = :d_id
AND ol_o_id = :o_id \
  RETURNING sum(ol_amount) into :ol_amount
"


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

#define NDISTS 10
#define ROWIDLEN 20

#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

#endif


int tkvcdinit (int plsqlflag)
{
   text stmbuf[SQL_BUF_SIZE];

   if (plsqlflag)
   {
     pldctx = (pldelctx *) malloc (sizeof(pldelctx));
     DISCARD
memset(pldctx,(char)0,(ub4)sizeof(pldelctx));
     /* Initialize */
     DISCARD OCIHandleAlloc(tpcenv,
(dvoid**)&pldctx->curp1, OCI_HTYPE_STMT, 0,
                        (dvoid**)0);
     DISCARD sprintf ((char *) stmbuf, SQLTXT);
     DISCARD OCIStmtPrepare(pldctx->curp1,
errhp, stmbuf,
                        (ub4) strlen((char *)stmbuf),
                   OCI_NTV_SYNTAX,
OCI_DEFAULT);
     DISCARD OCIERROR(errhp,
       OCIStmtExecute(tpcsvc,pldctx-
>curp1,errhp,1,0,NULLP(OCISnapshot),
                     NULLP(OCISnapshot),
OCI_DEFAULT));


     DISCARD OCIHandleAlloc(tpcenv,(dvoid**)
&pldctx->curp2, OCI_HTYPE_STMT,
                  0, (dvoid**)0);
#if defined(ISO5) || defined(ISO6) ||
defined(ISO8)
  #if defined(ISO5)
    sqlfile("../blocks/tkvcpdel_iso5.sql",stmbuf);
  #endif
  #if defined(ISO6)
    sqlfile("../blocks/tkvcpdel_iso6.sql",stmbuf);
  #endif
  #if defined(ISO8)
    sqlfile("../blocks/tkvcpdel_iso8.sql",stmbuf);
  #endif
#else

/* Replaced 04.01.20 TUXEDO Client */

#if 0
!     sqlfile("../blocks/tkvcpdel.sql",stmbuf);
#endif

sqlfile("/home/tpc/blocks/tkvcpdel.sql",stmbuf);
/* Replaced end */
#endif
     DISCARD OCIStmtPrepare(pldctx->curp2,
errhp, stmbuf,
                  (ub4)strlen((char *)stmbuf),
OCI_NTV_SYNTAX, OCI_DEFAULT);
     OCIBNDPL(pldctx->curp2, pldctx->w_id_bp ,
errhp,":w_id",
              ADR(w_id), SIZ(int),
SQLT_INT,&pldctx->w_id_len);
     OCIBNDPL(pldctx->curp2, pldctx->ordcnt_bp ,
errhp,":ordcnt",
              ADR(pldctx->ordcnt), SIZ(int),
SQLT_INT,&pldctx->ordcnt_len);

/* 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),
SQLT_ODT,&pldctx->del_date_len);
#endif

#ifdef TSL
     OCIBNDPL(pldctx->curp2, pldctx-
>del_date_bp,errhp,":now",
              dctx->del_date, SIZ(OCIDate),
SQLT_ODT,&pldctx->del_date_len);
#else
     OCIBNDPL(pldctx->curp2, pldctx-
>del_date_bp,errhp,":now",
              ADR(pldctx->del_date), SIZ(OCIDate),
SQLT_ODT,&pldctx->del_date_len);
#endif
/* Replaced end */

     OCIBNDPL(pldctx->curp2, pldctx-
>carrier_id_bp , errhp,
            ":carrier_id", ADR(o_carrier_id), SIZ(int),
            SQLT_INT, &pldctx->carrier_id_len);

     OCIBNDPLA(pldctx->curp2, pldctx->d_id_bp,
errhp,":d_id",
              pldctx->del_d_id, SIZ(int),SQLT_INT,
pldctx->del_d_id_len,
              NDISTS, &pldctx->del_d_id_rcnt);
     OCIBNDPLA(pldctx->curp2, pldctx->o_id_bp,
errhp,":order_id",
              pldctx->del_o_id,SIZ(int),SQLT_INT,
pldctx->del_o_id_len,NDISTS,
              &pldctx->del_o_id_rcnt);


/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
!/* Replaced T.kato 03.07.18 New Oracle10i tool
kit */
!#if 0
!!     OCIBNDPLA(pldctx->curp2, pldctx-
>sums_bp, errhp,":sums",
!!               pldctx->sums,SIZ(int),SQLT_INT,
pldctx->sums_len,NDISTS,
!!               &pldctx->sums_rcnt);
!#endif
!
!#ifdef TSL
!     OCIBNDPLA(pldctx->curp2, pldctx-
>sums_bp, errhp,":sums",
!               pldctx->sums,SIZ(int),SQLT_INT,
pldctx->sums_len,NDISTS,
!               &pldctx->sums_rcnt);
!#else
!     OCIBNDPLA(pldctx->curp2, pldctx-
>sums_bp, errhp,":sums",
!               pldctx-
>sums,SIZ(float),SQLT_BFLOAT, pldctx-
>sums_len,NDISTS,
!               &pldctx->sums_rcnt);
!#endif
!/* Replaced end */
#endif

#ifdef USE_IEEE_NUMBER
     OCIBNDPLA(pldctx->curp2, pldctx-
>sums_bp, errhp,":sums",
               pldctx-
>sums,SIZ(float),SQLT_BFLOAT, pldctx-
>sums_len,NDISTS,
               &pldctx->sums_rcnt);
#else
     OCIBNDPLA(pldctx->curp2, pldctx-
>sums_bp, errhp,":sums",
               pldctx->sums,SIZ(int),SQLT_INT,
pldctx->sums_len,NDISTS,
               &pldctx->sums_rcnt);
#endif
/* Replaced end */




     OCIBNDPLA(pldctx->curp2, pldctx-
>o_c_id_bp, errhp,":o_c_id",
               pldctx->o_c_id,SIZ(int),SQLT_INT,
pldctx->o_c_id_len,NDISTS,
               &pldctx->o_c_id_rcnt);
     OCIBND(pldctx->curp2, pldctx->retry_bp ,
errhp,":retry",
            ADR(pldctx->retry), SIZ(int),SQLT_INT);

   }
   else
   {
```

```
    dctx = (delctx *) malloc (sizeof(delctx));
    memset(dctx,(char)0,sizeof(delctx));
    dctx->norow = 0;
    actx = (amtctx *) malloc (sizeof(amtctx));
    memset(actx,(char)0,sizeof(amtctx));

    OCIHandleAlloc(tpcenv, (dvoid **)(&dctx-
>curd1), OCI_HTYPE_STMT, 0,
              (dvoid**)0);
    DISCARD sprintf ((char *) stmbuf, "%s",
SQLTXT1);
    DISCARD OCIStmtPrepare(dctx->curd1,
errhp, stmbuf,
                strlen((char
*)stmbuf),OCI_NTV_SYNTAX, OCI_DEFAULT);

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

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

  /* open third cursor */

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


  /* bind variables */

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

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

  /* open fourth cursor */


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

  /* bind variables */

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

  /* open sixth cursor */

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

  /* bind variables */


    OCIBND(dctx->curd6,dctx-
>amt_bp,errhp,":amt",dctx->amt,SIZ(int),
         SQLT_INT);
    OCIBND(dctx->curd6,dctx-
>w_id_bp6,errhp,":w_id",dctx->w_id,SIZ(int),
          SQLT_INT);
    OCIBND(dctx->curd6,dctx-
>d_id_bp6,errhp,":d_id",dctx->d_id,SIZ(int),
          SQLT_INT);
    OCIBND(dctx->curd6,dctx-
>c_id_bp,errhp,":c_id",dctx->c_id,SIZ(int),
          SQLT_INT);
  }
  return (0);

}


void shiftdata(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 plsqlflag)
{
  //int i, j;
  int i;
  //int rpc,rcount,count;
  int rpc,rcount;
  int invalid;

  if (plsqlflag)
  {

    pldctx->w_id_len = sizeof (int);
    pldctx->carrier_id_len = sizeof (int);
    for (i = 0; i < NDISTS; i++)
    {
      pldctx->del_o_id_len[i] = sizeof(int);
      del_o_id[i] = 0;
    }
    pldctx->del_date_len = DEL_DATE_LEN;
    DISCARD memcpy(&pldctx-
>del_date,&cr_date,sizeof(OCIDate));

    pldctx->retry=0;

    DISCARD OCIERROR(errhp,
      OCIStmtExecute(tpcsvc,pldctx-
>curp2,errhp,1,0,NULLP(CONST OCISnapshot),

NULLP(OCISnapshot),OCI_DEFAULT));
    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_d_id[i] - 1] = pldctx-
>del_o_id[i];
  }
  else
  {


retry:

    invalid = 0;

  /* initialization for array operations */

    for (i = 0; i < NDISTS; i++)
    {
      dctx->del_o_id_ind[i] = TRUE;

      dctx->d_id_ind[i] = TRUE;
      dctx->c_id_ind[i] = TRUE;
      dctx->del_date_ind[i] = TRUE;
      dctx->carrier_id_ind[i] = TRUE;
      dctx->amt_ind[i] = TRUE;

      dctx->del_o_id_len[i] = SIZ(dctx-
>del_o_id[0]);
      dctx->w_id_len[i] = SIZ(dctx->w_id[0]);
      dctx->d_id_len[i] = SIZ(dctx->d_id[0]);
      dctx->c_id_len[i] = SIZ(dctx->c_id[0]);
      dctx->del_date_len[i] = DEL_DATE_LEN;
      dctx->carrier_id_len[i] = SIZ(dctx-
>carrier_id[0]);
      dctx->amt_len[i] = SIZ(dctx->amt[0]);

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

    memset(actx,(char)0,sizeof(amtctx));

  /* array select from new_order and orders
tables */

    execstatus=OCIStmtExecute(tpcsvc,dctx-
>curd1,errhp,NDISTS,0,
            NULLP(CONST
OCISnapshot),NULLP(OCISnapshot),OCI_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 plsqlflag)
{

  if (plsqlflag)
  {
    if (pldctx)
    {
      DISCARD OCIHandleFree((dvoid *)dctx-
>curd0,OCI_HTYPE_STMT);
      DISCARD free(pldctx);
    }
```

```
  }
  else
  {
    if (dctx)
    {
      OCIHandleFree((dvoid *)dctx-
>curd1,OCI_HTYPE_STMT);
      OCIHandleFree((dvoid *)dctx-
>curd2,OCI_HTYPE_STMT);
      OCIHandleFree((dvoid *)dctx-
>curd3,OCI_HTYPE_STMT);
      OCIHandleFree((dvoid *)dctx-
>curd4,OCI_HTYPE_STMT);
      OCIHandleFree((dvoid *)dctx-
>curd5,OCI_HTYPE_STMT);
      OCIHandleFree((dvoid *)dctx-
>curd6,OCI_HTYPE_STMT);
      DISCARD free (dctx);
    }
  }

}


::::::::::::::::::::::::::::::::::
 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"

#ifndef 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->curn1),
          OCI_HTYPE_STMT, 0, (dvoid**)0));
/* Replaced T.kato 03.03.19 Replaced Oracle
10i tool kit */
/*  sqlfile("../blocks/tkvcpnew.sql",stmbuf);*/
#if defined(ISO)
  sqlfile("../blocks/tkvcpnew_iso.sql",stmbuf);
#else
#if defined(ISO7)
  sqlfile("../blocks/tkvcpnew_iso7.sql",stmbuf);
#else
/* Replaced 04.01.20 TUXEDO Client */
#if 0
!  sqlfile("../blocks/tkvcpnew.sql",stmbuf);
#endif
  sqlfile("/home/tpc/blocks/tkvcpnew.sql",stmbuf);
/* Replaced end */
#endif
#endif
/* Replaced end */

  DISCARD
OCIERROR(errhp,OCIStmtPrepare(nctx->curn1,
errhp, stmbuf,
          strlen((char *)stmbuf),
OCI_NTV_SYNTAX, OCI_DEFAULT));

  /* bind variables */

  OCIBNDPL(nctx->curn1, nctx->w_id_bp, errhp,
":w_id",ADR(w_id),SIZ(w_id),
          SQLT_INT, &nctx->w_id_len);
  OCIBNDPL(nctx->curn1, nctx->d_id_bp, errhp,
":d_id",ADR(d_id),SIZ(d_id),
          SQLT_INT, &nctx->d_id_len);
  OCIBNDPL(nctx->curn1, nctx->c_id_bp, errhp,
":c_id",ADR(c_id),SIZ(c_id),
          SQLT_INT, &nctx->c_id_len);
  OCIBNDPL(nctx->curn1, nctx->o_all_local_bp,
errhp, ":o_all_local",
          ADR(o_all_local),
SIZ(o_all_local),SQLT_INT, &nctx-
>o_all_local_len);
  OCIBNDPL(nctx->curn1, nctx->o_all_cnt_bp,
errhp, ":o_ol_cnt",ADR(o_ol_cnt),
          SIZ(o_ol_cnt),SQLT_INT, &nctx-
>o_ol_cnt_len);
  OCIBNDPL(nctx->curn1, nctx->w_tax_bp,
errhp, ":w_tax",ADR(w_tax),SIZ(w_tax),
          SQLT_FLT, &nctx->w_tax_len);
  OCIBNDPL(nctx->curn1, nctx->d_tax_bp, errhp,
":d_tax",ADR(d_tax),SIZ(d_tax),
          SQLT_FLT, &nctx->d_tax_len);
  OCIBNDPL(nctx->curn1, nctx->o_id_bp, errhp,
":o_id",ADR(o_id),SIZ(o_id),
          SQLT_INT, &nctx->o_id_len);
  OCIBNDPL(nctx->curn1, nctx->c_discount_bp,
errhp, ":c_discount",
          ADR(c_discount),
SIZ(c_discount),SQLT_FLT, &nctx-
>c_discount_len);
  OCIBNDPL(nctx->curn1, nctx->c_credit_bp,
errhp, ":c_credit",c_credit,
          SIZ(c_credit),SQLT_CHR, &nctx-
>c_credit_len);
  OCIBNDPL(nctx->curn1, nctx->c_last_bp,
errhp, ":c_last",c_last,SIZ(c_last),
          SQLT_STR, &nctx->c_last_len);
  OCIBNDPL(nctx->curn1, nctx->retries_bp,
errhp, ":retry",ADR(retries),
          SIZ(retries),SQLT_INT, &nctx-
>retries_len);
  OCIBNDPL(nctx->curn1, nctx->cr_date_bp,
errhp, ":cr_date",&cr_date,
          SIZ(OCIDate), SQLT_ODT, &nctx-
>cr_date_len);

  OCIBNDPLA(nctx->curn1, nctx-
>ol_i_id_bp,errhp,":ol_i_id",nol_i_id,
          SIZ(int), SQLT_INT, nctx-
>nol_i_id_len,NITEMS,&nctx->nol_i_count);
  OCIBNDPLA(nctx->curn1, nctx-
>ol_supply_w_id_bp, errhp, ":ol_supply_w_id",
          nol_supply_w_id,SIZ(int),SQLT_INT,
nctx->nol_supply_w_id_len,
          NITEMS, &nctx->nol_s_count);


/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
!  OCIBNDPLA(nctx->curn1, nctx-
>ol_quantity_bp,errhp,":ol_quantity",
!          nol_quantity, SIZ(int),SQLT_INT,nctx-
>nol_quantity_len,
!          NITEMS,&nctx->nol_q_count);
!  OCIBNDPLA(nctx->curn1, nctx-
>i_price_bp,errhp,":i_price",i_price,SIZ(int),
!          SQLT_INT, nctx->i_price_len, NITEMS,
&nctx->nol_item_count);
#endif

#ifdef USE_IEEE_NUMBER
  OCIBNDPLA(nctx->curn1, nctx-
>ol_quantity_bp,errhp,":ol_quantity",
          nol_quantity,
SIZ(float),SQLT_BFLOAT,nctx-
>nol_quantity_len,
          NITEMS,&nctx->nol_q_count);

  OCIBNDPLA(nctx->curn1, nctx-
>i_price_bp,errhp,":i_price",i_price,SIZ(float),
          SQLT_BFLOAT, nctx->i_price_len,
NITEMS, &nctx->nol_item_count);
#else
```

```
  OCIBNDPLA(nctx->curn1, nctx-
>ol_quantity_bp,errhp,":ol_quantity",
        nol_quantity, SIZ(int),SQLT_INT,nctx-
>nol_quantity_len,
        NITEMS,&nctx->nol_q_count);

  OCIBNDPLA(nctx->curn1, nctx-
>i_price_bp,errhp,":i_price",i_price,SIZ(int),
        SQLT_INT, nctx->i_price_len, NITEMS,
&nctx->nol_item_count);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

  OCIBNDPLA(nctx->curn1, nctx-
>i_name_bp,errhp,":i_name",i_name,
        SIZ(i_name[0]),SQLT_STR, nctx-
>i_name_len,NITEMS,
        &nctx->nol_name_count);

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
!  OCIBNDPLA(nctx->curn1, nctx-
>s_quantity_bp,errhp,":s_quantity",s_quantity,
!         SIZ(int), SQLT_INT,nctx-
>s_quant_len,NITEMS,&nctx->nol_qty_count);
#endif

#ifdef USE_IEEE_NUMBER
  OCIBNDPLA(nctx->curn1, nctx-
>s_quantity_bp,errhp,":s_quantity",s_quantity,
        SIZ(float), SQLT_BFLOAT,nctx-
>s_quant_len,NITEMS,&nctx->nol_qty_count);
#else
  OCIBNDPLA(nctx->curn1, nctx-
>s_quantity_bp,errhp,":s_quantity",s_quantity,
        SIZ(int), SQLT_INT,nctx-
>s_quant_len,NITEMS,&nctx->nol_qty_count);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

  OCIBNDPLA(nctx->curn1, nctx-
>s_bg_bp,errhp,":brand_generic",brand_generic,
        SIZ(char), SQLT_CHR,nctx-
>s_bg_len,NITEMS,&nctx->nol_bg_count);

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
!  OCIBNDPLA(nctx->curn1, nctx-
>ol_amount_bp,errhp,":ol_amount",nol_amount,
!         SIZ(int),SQLT_INT, nctx-
>nol_amount_len,NITEMS,&nctx-
>nol_am_count);
!  OCIBNDPLA(nctx->curn1, nctx-
>s_remote_bp,errhp,":s_remote",nctx-
>s_remote,
!         SIZ(int),SQLT_INT, nctx-
>s_remote_len,NITEMS,&nctx-
>s_remote_count);
#endif

#ifdef USE_IEEE_NUMBER
  OCIBNDPLA(nctx->curn1, nctx-
>ol_amount_bp,errhp,":ol_amount",nol_amount,
        SIZ(float),SQLT_BFLOAT, nctx-
>nol_amount_len,NITEMS,&nctx-
>nol_am_count);

  OCIBNDPLA(nctx->curn1, nctx-
>s_remote_bp,errhp,":s_remote",nctx-
>s_remote,
        SIZ(float),SQLT_BFLOAT, nctx-
>s_remote_len,NITEMS,&nctx-
>s_remote_count);
#else
  OCIBNDPLA(nctx->curn1, nctx-
>ol_amount_bp,errhp,":ol_amount",nol_amount,
        SIZ(int),SQLT_INT, nctx-
>nol_amount_len,NITEMS,&nctx-
>nol_am_count);

  OCIBNDPLA(nctx->curn1, nctx-
>s_remote_bp,errhp,":s_remote",nctx-
>s_remote,
        SIZ(int),SQLT_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->curn2),
              OCI_HTYPE_STMT, 0, (dvoid**)0));
  DISCARD sprintf ((char *) stmbuf, SQLTXT2);
  DISCARD
OCIERROR(errhp,OCIStmtPrepare(nctx->curn2,
errhp, stmbuf,
              strlen((char *)stmbuf),
OCI_NTV_SYNTAX, OCI_DEFAULT));


  /* execute second cursor to init newinit
package */
  {
    int idx1arr[NITEMS];
    OCIBind *idx1arr_bp;
    ub2 idx1arr_len[NITEMS];
/* 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_ol_cnt = NITEMS;


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

    DBGLOG("NEW:[1]Start",0);
     execstatus = OCIStmtExecute(tpcsvc,nctx-
>curn2,errhp,1,0,
              NULLP(CONST
OCISnapshot),NULLP(OCISnapshot),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_ol_cnt = NITEMS;
  o_all_local = 1;
  for (i = 0; i < NITEMS; i++) {
    if (nol_i_id[i] == 0) {
      o_ol_cnt = i;
      break;
    }
    if (nol_supply_w_id[i] != w_id) {

/* 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_ol_cnt_len = sizeof(o_ol_cnt);
  nctx->w_tax_len = 0;
  nctx->d_tax_len = 0;
  nctx->o_id_len = sizeof(o_id);
  nctx->c_discount_len = 0;
  nctx->c_credit_len = 0;
  nctx->c_last_len = 0;
  nctx->retries_len = sizeof(retries);
  nctx->cr_date_len = sizeof(cr_date);
  /* this is the row count */
  rcount = o_ol_cnt;
  nctx->nol_i_count = o_ol_cnt;
  nctx->nol_q_count = o_ol_cnt;
  nctx->nol_s_count = o_ol_cnt;
  nctx->s_remote_count = o_ol_cnt;

  nctx->nol_qty_count  = 0;
  nctx->nol_bg_count = 0;
  nctx->nol_item_count = 0;
  nctx->nol_name_count = 0;
  nctx->nol_am_count   = 0;

  /* initialization for array operations */
  for (i = 0; i < o_ol_cnt; i++) {
    nctx->ol_number[i] = i + 1;
    nctx->nol_i_id_len[i] = sizeof(int);
    nctx->nol_supply_w_id_len[i] = sizeof(int);
    nctx->nol_quantity_len[i] = sizeof(int);
```

```
    nctx->nol_amount_len[i] = sizeof(int);
    nctx->ol_o_id_len[i] = sizeof(int);
    nctx->ol_number_len[i] = sizeof(int);
    nctx->ol_dist_info_len[i] = nctx-
>s_dist_info_len[i];
    nctx->s_remote_len[i] = sizeof(int);
    nctx->s_quant_len[i] = sizeof(int);
    nctx->i_name_len[i]=0;
    nctx->s_bg_len[i] = 0;
  }
  for (i = o_ol_cnt; i < NITEMS; i++) {

    nctx->nol_i_id_len[i] = 0;
    nctx->nol_supply_w_id_len[i] = 0;
    nctx->nol_quantity_len[i] = 0;
    nctx->nol_amount_len[i] = 0;
    nctx->ol_o_id_len[i] = 0;
    nctx->ol_number_len[i] = 0;
    nctx->ol_dist_info_len[i] = 0;
    nctx->s_remote_len[i] = 0;
    nctx->s_quant_len[i] = 0;
    nctx->i_name_len[i]=0;
    nctx->s_bg_len[i] = 0;
  }

  DBGLOG("NEW:[2]Start",0);
  execstatus = OCIStmtExecute(tpcsvc,nctx-
>curn1,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 == RECOVERR) {
   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-
>curn1,OCI_HTYPE_STMT);
    DISCARD OCIHandleFree((dvoid *)nctx-
>curn2,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_tpccitrc
  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_ol_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 tkvcoinit ()
{
  int i;
  text stmbuf[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 *) stmbuf, SQLCUR0);
  DISCARD OCIERROR(errhp,
    OCIStmtPrepare(octx-
>curo0,errhp,stmbuf,(ub4)strlen((char *)stmbuf),

OCI_NTV_SYNTAX,OCI_DEFAULT));
  DISCARD OCIERROR(errhp,
    OCIAttrSet(octx-
>curo0,OCI_HTYPE_STMT,&octx->norow,0,

OCI_ATTR_PREFETCH_ROWS,errhp));
/* get order/customer info back based on rowid */
  DISCARD sprintf((char *) stmbuf, SQLCUR1);
  DISCARD OCIERROR(errhp,
    OCIStmtPrepare(octx-
>curo1,errhp,stmbuf,(ub4)strlen((char *)stmbuf),

OCI_NTV_SYNTAX,OCI_DEFAULT));
  DISCARD OCIERROR(errhp,
    OCIAttrSet(octx-
>curo1,OCI_HTYPE_STMT,&octx->norow,0,

OCI_ATTR_PREFETCH_ROWS,errhp));

/*  c_id == 0, use lastname to find customer */
  DISCARD sprintf((char *) stmbuf, SQLCUR2);
  DISCARD OCIERROR(errhp,
    OCIStmtPrepare(octx-
>curo2,errhp,stmbuf,(ub4)strlen((char *)stmbuf),

OCI_NTV_SYNTAX,OCI_DEFAULT));
  DISCARD OCIERROR(errhp,
    OCIAttrSet(octx-
>curo2,OCI_HTYPE_STMT,&octx->norow,0,

OCI_ATTR_PREFETCH_ROWS,errhp));

  DISCARD sprintf((char *) stmbuf, SQLCUR3);
  DISCARD OCIERROR(errhp,
    OCIStmtPrepare(octx-
>curo3,errhp,stmbuf,(ub4)strlen((char *)stmbuf),

OCI_NTV_SYNTAX,OCI_DEFAULT));
  DISCARD OCIERROR(errhp,
    OCIAttrSet(octx-
>curo3,OCI_HTYPE_STMT,&octx->norow,0,

OCI_ATTR_PREFETCH_ROWS,errhp));

  DISCARD sprintf((char *) stmbuf, SQLCUR4);
  DISCARD OCIERROR(errhp,
    OCIStmtPrepare(octx-
>curo4,errhp,stmbuf,(ub4)strlen((char *)stmbuf),

OCI_NTV_SYNTAX,OCI_DEFAULT));
  DISCARD OCIERROR(errhp,
    OCIAttrSet(octx-
>curo4,OCI_HTYPE_STMT,&octx->norow,0,

OCI_ATTR_PREFETCH_ROWS,errhp));

  for (i = 0; i < NITEMS; i++) {

    octx->ol_supply_w_id_len[i] = sizeof(int);
    octx->ol_i_id_len[i] = sizeof(int);
    octx->ol_quantity_len[i] = sizeof(int);
    octx->ol_amount_len[i] = sizeof(int);
    octx->ol_delivery_d_len[i] =
sizeof(ol_d_base[0]);
  }
  octx->ol_supply_w_id_csize = NITEMS;
  octx->ol_i_id_csize = NITEMS;
  octx->ol_quantity_csize = NITEMS;
  octx->ol_amount_csize = NITEMS;
  octx->ol_delivery_d_csize = NITEMS;
  octx->ol_w_id_csize = NITEMS;
  octx->ol_o_id_csize = NITEMS;
  octx->ol_d_id_csize = NITEMS;
  octx->ol_w_id_len = sizeof(int);
  octx->ol_d_id_len = sizeof(int);
  octx->ol_o_id_len = sizeof(int);

  /* bind variables */

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

  OCIBND(octx->curo1,octx-
>c_rowid_bp,errhp,":cust_rowid", &octx-
>c_rowid_cust,
       sizeof( octx->c_rowid_ptr[0]),SQLT_RDD);
  OCIDEF(octx->curo1,octx-
>c_id_dp,errhp,1,ADR(c_id),SIZ(int),SQLT_INT);

/* 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),SQLT_FLT);
#endif

#ifdef USE_IEEE_NUMBER
  OCIDEF(octx->curo1,octx-
>c_balance_dp[0],errhp,2,ADR(c_balance),
       SIZ(double),SQLT_BDOUBLE);
#else
  OCIDEF(octx->curo1,octx-
>c_balance_dp[0],errhp,2,ADR(c_balance),
       SIZ(double),SQLT_FLT);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

  OCIDEF(octx->curo1,octx-
>c_first_dp[0],errhp,3,c_first,SIZ(c_first)-1,
       SQLT_CHR);
  OCIDEF(octx->curo1,octx-
>c_middle_dp[0],errhp,4,c_middle,
       SIZ(c_middle)-1,SQLT_AFC);
```

```
  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;

#if defined(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
OCISnapshot),NULLP(OCISnapshot),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 == RECOVERR))
      {
        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)
!       {
!          TpccUserLog(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
!       TpccUserLog(LOG_FILE_INF,
"ORDERSTATUS  rcount=%d\n",rcount);
!       return (-1);
#endif

      TpccUserLog(LOG_FILE_INF,
"ORDERSTATUS  rcount=%d\n",rcount);
      TpccUserLog(LOG_FILE_INF, "
w_id  =%d\n",w_id);
      TpccUserLog(LOG_FILE_INF, "
d_id  =%d\n",d_id);
      TpccUserLog(LOG_FILE_INF, "
c_last='%s'\n",c_last);
      TpccUserLog(LOG_FILE_INF, "
retries=%d\n",retries);
      TpccUserLog(LOG_FILE_INF, "
errcode=%d\n",errcode);
      TpccUserLog(LOG_FILE_INF, "
execstatus=%d\n",execstatus);
      TpccUserLog(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
OCISnapshot),NULLP(OCISnapshot),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 == RECOVERR))
       {
         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
OCISnapshot),
!
NULLP(OCISnapshot),OCI_DEFAULT);
!        DBGLOG("ORD:[3]End
>%d",execstatus);
!        /* will get OCI_NO_DATA if <100 found */
!        if (cbctx.count > 0)
!        {
!          TpccUserLog (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 == RECOVERR))
!          {
!            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=OCIStmtExecute(tpcsvc,octx->curo0,errhp,cbctx.count,
                                0,NULLP(CONST OCISnapshot),

NULLP(OCISnapshot),OCI_DEFAULT);

      DISCARD OCIAttrGet(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)
      {
        TpccUserLog (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=OCIStmtExecute(tpcsvc,octx->curo1,errhp,1,0,
                NULLP(CONST OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);
    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=OCIStmtExecute(tpcsvc,octx->curo2,errhp,1,0,
                NULLP(CONST OCISnapshot),NULLP(OCISnapshot),
                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 */
#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_entry_d_base,(text*)FULLDATE,SIZ(FULLDATE),(text*
)0,0,&datelen,o_entry_d));
    printf ("o_entry_d = %s\n", o_entry_d);
    printf ("o_carrier_id = %d\n", o_carrier_id);
    printf ("o_ol_cnt = %d\n", o_ol_cnt);
    printf ("------------------------------------------------\n\n", sdate);

if (!secondread) {
    printf ("Sleep before re-read order at: %s\n", sdate);
    sleep (30);
    sysdate (sdate);
    printf ("Wake up and reread at: %s\n", sdate);
    secondread = 1;
    goto retry;
}
#endif /* ISO9 */
  }
  octx->ol_w_id_len = sizeof(int);
  octx->ol_d_id_len = sizeof(int);
  octx->ol_o_id_len = sizeof(int);

  DBGLOG("ORD:[6]Start",0);
  execstatus = OCIStmtExecute(tpcsvc,octx->curo3,errhp,o_ol_cnt,0,
                NULLP(CONST OCISnapshot),NULLP(OCISnapshot),
                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);
!!TpccUserLog(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);
!!TpccUserLog(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);
!!TpccUserLog(LOG_FILE_INF, "DBG_LOG cur2  : row_id=[%s]\n", b_row_id);
!!TpccUserLog(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++)
  {
    ol_del_len[i]=sizeof(ol_delivery_d[i]);
    DISCARD OCIERROR(errhp,OCIDateToText(errhp,&ol_d_base[i],
      (const text*)SHORTDATE,(ub1)strlen(SHORTDATE),(text*)0,0,
      &ol_del_len[i], ol_delivery_d[i]));
/*
  cvtdmy(ol_d_base[i],ol_delivery_d[i]);
*/
  }

/* 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(tpcsvc, 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(tpcsvc, 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 inittpcc.init_pay;
END;"


int tkvcpinit (void)
{
  text stmbuf[SQL_BUF_SIZE];

  pctx = (payctx *)malloc(sizeof(payctx));
  memset(pctx,(char)0,sizeof(payctx));

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

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

  /*  build the init statement  and execute it */

  sprintf ((char*)stmbuf, SQLTXT_INIT);
  DISCARD
OCIERROR(errhp,OCIStmtPrepare(pctx->curpi,
errhp, stmbuf,
        strlen((char *)stmbuf),
OCI_NTV_SYNTAX, OCI_DEFAULT));
  DBGLOG("PAY:[1]Start",0);
```

```
  DISCARD OCIERROR(errhp,
OCIStmtExecute(tpcsvc,pctx->curpi,errhp,1,0,
              NULLP(CONST
OCISnapshot),NULLP(OCISnapshot),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,
            NULLP(CONST OCISnapshot),NULLP(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,
            NULLP(CONST OCISnapshot),NULLP(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 == RECOVERR) {
      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
```

```
::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::

#ifdef RCSID
static char *RCSid =
  "$Header: plsto.c 7010000.3 95/02/14
12:48:03 plai Generic<base> $ Copyr (c) 1994
Oracle";
#endif /* RCSID */

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

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

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


#include "forlinux.h"
#include "log.h"
#include "tpcc.h"
#include "GlobalArea.h"
#include "prototype.h"

#ifdef 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 = (stoctx *)malloc(sizeof(stoctx));
  memset(sctx,(char)0,sizeof(stoctx));

  sctx->norow=0;

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

OCI_ATTR_PREFETCH_ROWS,errhp));
#endif

  /* bind variables */

  OCIBND(sctx->curs,sctx->w_id_bp,errhp,
":w_id", ADR(w_id),sizeof(int),
       SQLT_INT);
  OCIBND(sctx->curs,sctx->d_id_bp,errhp,
":d_id", ADR(d_id),sizeof(int),
    SQLT_INT);

/* 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 == RECOVERR)
       || (errcode == SNAPSHOT_TOO_OLD))
    {
      retries++;
      goto retry;
    } else {
      return -1;
    }
   }

  return (0);
}


void tkvcsdone ()
{
  if(sctx) free(sctx);
}


::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::
 svrapl/3tier/tpccpl.c
::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::

#ifdef RCSID
static char *RCSid =
  "$Header: tpccpl.c 7030100.2 96/04/02
17:51:34 plai Generic<base> $ Copyr (c) 1994
Oracle";
#endif /* RCSID */


/*================================
==============================+
|       Copyright (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_EXECUTE\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, "tkvuopen() 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: %#x\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 *oradt)
{
 struct tm *loctime;
 time_t  int_time;

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

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

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

/* Convert the current date and time into local
time */
```

```
loctime = localtime( &int_time);

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

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


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

 return;

}
void cvtdmy (unsigned char *oradt, char
*outdate)
{

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


      int day,month,year;

      memcpy(&Date,oradt,7);

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

      return;
}

void cvtdmyhms (unsigned char *oradt, char
*outdate)
{

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


      int day,month,year;
      int hour,min,sec;

      memcpy(&Date,oradt,7);

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

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

      return;
}
#endif


void TPCexit (void)
{

  TpccUserLog(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 normaly 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;
  }
 TpccUserLog(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
!     TpccUserLog ("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
**)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);


/* Replaced T.Kato 02.10.24 Retry until
successfully
!  OCIServerAttach(tpcsrv, errhp, (text
*)0,0,OCI_DEFAULT);
*/
  for (;;) {
    rval = OCIServerAttach(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 *)tpcsvc,
OCI_HTYPE_SVCCTX, (dvoid *)tpcsrv,
(ub4)0,OCI_ATTR_SERVER, errhp);
  OCIHandleAlloc((dvoid *)tpcenv, (dvoid
**)&tpcusr, OCI_HTYPE_SESSION, 0 , (dvoid
**)0);
  OCIAttrSet((dvoid *)tpcusr,
OCI_HTYPE_SESSION, (dvoid *)uid,
(ub4)strlen(uid),OCI_ATTR_USERNAME,
errhp);
  OCIAttrSet((dvoid *)tpcusr,
OCI_HTYPE_SESSION, (dvoid *)pwd,
(ub4)strlen(pwd),
       OCI_ATTR_PASSWORD, errhp);
  OCIERROR(errhp, OCISessionBegin(tpcsvc,
errhp, tpcusr, OCI_CRED_RDBMS,
OCI_DEFAULT));

  OCIAttrSet(tpcsvc, OCI_HTYPE_SVCCTX,
tpcusr, 0, OCI_ATTR_SESSION, errhp);

  /* run all transaction in serializable mode */

  OCIHandleAlloc(tpcenv, (dvoid **)&curi,
OCI_HTYPE_STMT, 0, (dvoid**)0);
  sprintf ((char *) stmbuf, SQLTXT);
  OCIStmtPrepare(curi, errhp, stmbuf,
strlen((char *)stmbuf), OCI_NTV_SYNTAX,
OCI_DEFAULT);
  DBGLOG("INI:[1]Start",0);
  OCIERROR(errhp,OCIStmtExecute(tpcsvc,
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(tpcsvc,
curi, errhp,1,0,0,0,OCI_DEFAULT));
    OCIHandleFree((dvoid *)curi,
OCI_HTYPE_STMT);
  }
*/
  if (tracelevel == 3) {
    OCIHandleAlloc(tpcenv, (dvoid **)&curi,
OCI_HTYPE_STMT, 0, (dvoid**)0);
    memset(stmbuf,0,100);
    sprintf ((char *) stmbuf, SQLTXTTIM);
    OCIStmtPrepare(curi, errhp, stmbuf,
strlen((char *)stmbuf),
        OCI_NTV_SYNTAX, OCI_DEFAULT);
    DBGLOG("INI:[2]Start",0);
    OCIERROR(errhp, OCIStmtExecute(tpcsvc,
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 ()) {  /* new order */
    TPCexit ();
    return (-1);
  }
  else
    new_init = 1;

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

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

#ifdef DEL_ORA8I
  if (tkvcdinit ()) {  /* 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 ()) {  /* stock level */
    TPCexit ();
    return (-1);
  }
  else
    sto_init = 1;

  return (0);

}



int TPCnew (struct newstruct *str)
{
/* Added T.Kato 02.11.25 */
#ifndef AVOID_DEADLOCK
  static int
init_value_index[NITEMS]={0,1,2,3,4,5,6,7,8,9,1
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 /* USE_IEEE_NUMBER */
/* Replaced end */

  }
```

```
  retries = 0;

#ifndef AVOID_DEADLOCK

  for (i = NITEMS; i > 0; i--) {
    if (nol_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(nol_i_id, str, 0, ordl_cnt-1);

#endif

/*
  vgetdate(cr_date); */


OCIERROR(errhp,OCIDateSysDate(errhp,&cr_d
ate));

   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*)FULLDAT
E,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)(nol_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] =
nol_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)(nol_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_d
ate));

  if (bylastname) {
    c_id = 0;
    strncpy (c_last, str->payin.c_last, 17);
  }
  else {
    c_id = str->payin.c_id;
    strcpy (c_last, " ");
  }
  retries = 0;

  if ((str->payout.terror = tkvcp ()) != 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_da
te,

(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_sin
ce,

(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 != RECOVERR)
      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) {
      TpccUserLog (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 != RECOVERR)
      str->delout.terror = IRRECERR;
    return (-1);
  }

/* Replaced T.Kato 02.10.24 Chenged 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) {
       TpccUserLog (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);

}

#ifndef 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_item(str,left,(left+right)/2);
  last = left;
  for(i=left+1;i<=right;i++)
    if(arr[i] < arr[left])
      swap_item(str,last,i);
  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

```
::::::::::::::::::::::::::::::::::
 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 = 565001
         LOGPATH = /w00
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl065a
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 569001
         LOGPATH = /w01
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl065a
         SUTPORT = 80
         SUTTERM = 3550
         BASENO = 572751
         LOGPATH = /w02
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl065b
         SUTPORT = 80
         SUTTERM = 200
         BASENO = 576301
         LOGPATH = /w02
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl065b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 576501
         LOGPATH = /w03
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl065b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 580251
         LOGPATH = /w04
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl065b
         SUTPORT = 80
         SUTTERM = 3600
         BASENO = 584001
         LOGPATH = /w05
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl066a
         SUTPORT = 80
         SUTTERM = 150
         BASENO = 700601
         LOGPATH = /w05
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl066a
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 700751
         LOGPATH = /w06
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl066a
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 704501
         LOGPATH = /w07
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl066a
         SUTPORT = 80
         SUTTERM = 3650
         BASENO = 708251
         LOGPATH = /w08
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl066b
         SUTPORT = 80
         SUTTERM = 100
         BASENO = 711901
         LOGPATH = /w08
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl066b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 712001
         LOGPATH = /w09
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl066b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 715751
         LOGPATH = /w10
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl066b
         SUTPORT = 80
         SUTTERM = 3700
         BASENO = 719501
         LOGPATH = /w11
         LOGLEVEL = 0
      ENDSUT
   ENDRTE
   STARTVARIABLE
      WAREHOUSE = 108480
      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 = 836201
         LOGPATH = /w00
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl067a
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 840201
         LOGPATH = /w01
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl067a
         SUTPORT = 80
         SUTTERM = 3550
         BASENO = 843951
         LOGPATH = /w02
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl067b
         SUTPORT = 80
         SUTTERM = 200
         BASENO = 847501
         LOGPATH = /w02
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl067b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 847701
         LOGPATH = /w03
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl067b
         SUTPORT = 80
```

```
    SUTTERM = 3750
    BASENO = 851451
    LOGPATH = /w04
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl067b
    SUTPORT = 80
    SUTTERM = 3600
    BASENO = 855201
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl069a
    SUTPORT = 80
    SUTTERM = 150
    BASENO = 971801
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl069a
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 971951
    LOGPATH = /w06
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl069a
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 975701
    LOGPATH = /w07
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl069a
    SUTPORT = 80
    SUTTERM = 3650
    BASENO = 979451
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl069b
    SUTPORT = 80
    SUTTERM = 100
    BASENO = 983101
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl069b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 983201
    LOGPATH = /w09
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl069b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 986951
    LOGPATH = /w10
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl069b
    SUTPORT = 80
    SUTTERM = 3700
    BASENO = 990701
    LOGPATH = /w11
    LOGLEVEL = 0
  ENDSUT
 ENDRTE
 STARTVARIABLE
  WAREHOUSE = 108480
  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 = 1
    LOGPATH = /w00
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl039a
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 4001
    LOGPATH = /w01
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl039a
    SUTPORT = 80
    SUTTERM = 3550
    BASENO = 7751
    LOGPATH = /w02
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl039b
    SUTPORT = 80
    SUTTERM = 200
    BASENO = 11301
    LOGPATH = /w02
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl039b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 11501
    LOGPATH = /w03
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl039b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 15251
    LOGPATH = /w04
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl039b
    SUTPORT = 80
    SUTTERM = 3600
    BASENO = 19001
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl040a
    SUTPORT = 80
    SUTTERM = 150
    BASENO = 135601
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl040a
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 135751
    LOGPATH = /w06
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl040a
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 139501
    LOGPATH = /w07
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl040a
    SUTPORT = 80
    SUTTERM = 3650
    BASENO = 143251
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl040b
    SUTPORT = 80
    SUTTERM = 100
    BASENO = 146901
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl040b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 147001
    LOGPATH = /w09
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl040b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 150751
```

```
        LOGPATH = /w10
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl040b
        SUTPORT = 80
        SUTTERM = 3700
        BASENO = 154501
        LOGPATH = /w11
        LOGLEVEL = 0
      ENDSUT
    ENDRTE
    STARTVARIABLE
      WAREHOUSE = 108480
      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 = 271201
        LOGPATH = /w00
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl041a
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 275201
        LOGPATH = /w01
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl041a
        SUTPORT = 80
        SUTTERM = 3550
        BASENO = 278951
        LOGPATH = /w02
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl041b
        SUTPORT = 80
        SUTTERM = 200
        BASENO = 282501
        LOGPATH = /w02
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl041b
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 282701
        LOGPATH = /w03
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl041b
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 286451
        LOGPATH = /w04
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl041b
        SUTPORT = 80
        SUTTERM = 3600
        BASENO = 290201
        LOGPATH = /w05
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl042a
        SUTPORT = 80
        SUTTERM = 150
        BASENO = 406801
        LOGPATH = /w05
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl042a
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 406951
        LOGPATH = /w06
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl042a
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 410701
        LOGPATH = /w07
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl042a
        SUTPORT = 80
        SUTTERM = 3650
        BASENO = 414451
        LOGPATH = /w08
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl042b
        SUTPORT = 80
        SUTTERM = 100
        BASENO = 418101
        LOGPATH = /w08
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl042b
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 418201
        LOGPATH = /w09
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl042b
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 421951
        LOGPATH = /w10
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl042b
        SUTPORT = 80
        SUTTERM = 3700
        BASENO = 425701
        LOGPATH = /w11
        LOGLEVEL = 0
      ENDSUT
    ENDRTE
    STARTVARIABLE
      WAREHOUSE = 108480
      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 = 542401
        LOGPATH = /w00
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl043a
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 546401
        LOGPATH = /w01
        LOGLEVEL = 0
      ENDSUT
```

```
    STARTSUT
      SUTHOST = cl043a
      SUTPORT = 80
      SUTTERM = 3550
      BASENO = 550151
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl043b
      SUTPORT = 80
      SUTTERM = 200
      BASENO = 553701
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl043b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 553901
      LOGPATH = /w03
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl043b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 557651
      LOGPATH = /w04
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl043b
      SUTPORT = 80
      SUTTERM = 3600
      BASENO = 561401
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl044a
      SUTPORT = 80
      SUTTERM = 150
      BASENO = 678001
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl044a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 678151
      LOGPATH = /w06
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl044a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 681901
      LOGPATH = /w07
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl044a
      SUTPORT = 80
      SUTTERM = 3650
      BASENO = 685651
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl044b
      SUTPORT = 80
      SUTTERM = 100
      BASENO = 689301
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl044b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 689401
      LOGPATH = /w09
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl044b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 693151
      LOGPATH = /w10
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl044b
      SUTPORT = 80
      SUTTERM = 3700
      BASENO = 696901
      LOGPATH = /w11
      LOGLEVEL = 0
    ENDSUT
  ENDRTE
  STARTVARIABLE
    WAREHOUSE = 108480
    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


::::::::::::::::::::::::::::::::::
 rte17.conf
::::::::::::::::::::::::::::::::::

#
#   rte17.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
  STARTRTE
    STARTSUT
      SUTHOST = cl045a
      SUTPORT = 80
      SUTTERM = 4000
      BASENO = 813601
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl045a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 817601
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl045a
      SUTPORT = 80
      SUTTERM = 3550
      BASENO = 821351
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl045b
      SUTPORT = 80
      SUTTERM = 200
      BASENO = 824901
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl045b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 825101
      LOGPATH = /w03
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl045b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 828851
      LOGPATH = /w04
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl045b
      SUTPORT = 80
      SUTTERM = 3600
      BASENO = 832601
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl046a
      SUTPORT = 80
      SUTTERM = 150
      BASENO = 949201
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl046a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 949351
      LOGPATH = /w06
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl046a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 953101
      LOGPATH = /w07
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl046a
```

```
        SUTPORT = 80
        SUTTERM = 3650
        BASENO = 956851
        LOGPATH = /w08
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl046b
        SUTPORT = 80
        SUTTERM = 100
        BASENO = 960501
        LOGPATH = /w08
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl046b
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 960601
        LOGPATH = /w09
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl046b
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 964351
        LOGPATH = /w10
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl046b
        SUTPORT = 80
        SUTTERM = 3700
        BASENO = 968101
        LOGPATH = /w11
        LOGLEVEL = 0
      ENDSUT
    ENDRTE
    STARTVARIABLE
      WAREHOUSE = 108480
      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 = 45201
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl105a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 49201
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl105a
      SUTPORT = 80
      SUTTERM = 3550
      BASENO = 52951
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl105b
      SUTPORT = 80
      SUTTERM = 200
      BASENO = 56501
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl105b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 56701
      LOGPATH = /w03
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl105b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 60451
      LOGPATH = /w04
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl105b
      SUTPORT = 80
      SUTTERM = 3600
      BASENO = 64201
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl106a
      SUTPORT = 80
      SUTTERM = 150
      BASENO = 180801
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl106a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 180951
      LOGPATH = /w06
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl106a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 184701
      LOGPATH = /w07
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl106a
      SUTPORT = 80
      SUTTERM = 3650
      BASENO = 188451
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl106b
      SUTPORT = 80
      SUTTERM = 100
      BASENO = 192101
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl106b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 192201
      LOGPATH = /w09
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl106b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 195951
      LOGPATH = /w10
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl106b
      SUTPORT = 80
      SUTTERM = 3700
      BASENO = 199701
      LOGPATH = /w11
      LOGLEVEL = 0
    ENDSUT
  ENDRTE
  STARTVARIABLE
    WAREHOUSE = 108480
    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 = cl108a
      SUTPORT = 80
      SUTTERM = 2000
      BASENO = 316401
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl108a
      SUTPORT = 80
      SUTTERM = 2000
      BASENO = 318401
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl108a
      SUTPORT = 80
      SUTTERM = 2000
      BASENO = 320401
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl108a
      SUTPORT = 80
      SUTTERM = 2000
      BASENO = 322401
      LOGPATH = /w03
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl108a
      SUTPORT = 80
      SUTTERM = 2000
      BASENO = 324401
      LOGPATH = /w04
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl108a
      SUTPORT = 80
      SUTTERM = 1300
      BASENO = 326401
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl108b
      SUTPORT = 80
      SUTTERM = 700
      BASENO = 327701
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl108b
      SUTPORT = 80
      SUTTERM = 2000
      BASENO = 328401
      LOGPATH = /w06
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl108b
      SUTPORT = 80
      SUTTERM = 1750
      BASENO = 330401
      LOGPATH = /w07
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl108b
      SUTPORT = 80
      SUTTERM = 1750
      BASENO = 332151
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl108b
      SUTPORT = 80
      SUTTERM = 1750
      BASENO = 333901
      LOGPATH = /w09
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl108b
      SUTPORT = 80
      SUTTERM = 1750
      BASENO = 335651
      LOGPATH = /w10
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl108b
      SUTPORT = 80
      SUTTERM = 1600
      BASENO = 337401
      LOGPATH = /w11
      LOGLEVEL = 0
    ENDSUT
  ENDRTE
  STARTVARIABLE
    WAREHOUSE = 108480
    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 = 452001
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl109a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 456001
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl109a
      SUTPORT = 80
      SUTTERM = 3550
      BASENO = 459751
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl109b
      SUTPORT = 80
      SUTTERM = 200
      BASENO = 463301
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl109b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 463501
      LOGPATH = /w03
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl109b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 467251
      LOGPATH = /w04
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl109b
      SUTPORT = 80
      SUTTERM = 3600
      BASENO = 471001
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl110a
      SUTPORT = 80
      SUTTERM = 150
      BASENO = 587601
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl110a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 587751
      LOGPATH = /w06
      LOGLEVEL = 0
```

```
    ENDSUT
    STARTSUT
      SUTHOST = cl110a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 591501
      LOGPATH = /w07
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl110a
      SUTPORT = 80
      SUTTERM = 3650
      BASENO = 595251
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl110b
      SUTPORT = 80
      SUTTERM = 100
      BASENO = 598901
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl110b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 599001
      LOGPATH = /w09
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl110b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 602751
      LOGPATH = /w10
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl110b
      SUTPORT = 80
      SUTTERM = 3700
      BASENO = 606501
      LOGPATH = /w11
      LOGLEVEL = 0
    ENDSUT
  ENDRTE
  STARTVARIABLE
    WAREHOUSE = 108480
    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 = 22601
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl061a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 26601
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl061a
      SUTPORT = 80
      SUTTERM = 3550
      BASENO = 30351
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl061b
      SUTPORT = 80
      SUTTERM = 200
      BASENO = 33901
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl061b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 34101
      LOGPATH = /w03
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl061b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 37851
      LOGPATH = /w04
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl061b
      SUTPORT = 80
      SUTTERM = 3600
      BASENO = 41601
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl062a
      SUTPORT = 80
      SUTTERM = 150
      BASENO = 158201
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl062a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 158351
      LOGPATH = /w06
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl062a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 162101
      LOGPATH = /w07
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl062a
      SUTPORT = 80
      SUTTERM = 3650
      BASENO = 165851
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl062b
      SUTPORT = 80
      SUTTERM = 100
      BASENO = 169501
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl062b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 169601
      LOGPATH = /w09
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl062b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 173351
      LOGPATH = /w10
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl062b
      SUTPORT = 80
      SUTTERM = 3700
      BASENO = 177101
      LOGPATH = /w11
      LOGLEVEL = 0
    ENDSUT
  ENDRTE
  STARTVARIABLE
    WAREHOUSE = 108480
    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 = 293801
         LOGPATH = /w00
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl063a
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 297801
         LOGPATH = /w01
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl063a
         SUTPORT = 80
         SUTTERM = 3550
         BASENO = 301551
         LOGPATH = /w02
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl063b
         SUTPORT = 80
         SUTTERM = 200
         BASENO = 305101
         LOGPATH = /w02
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl063b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 305301
         LOGPATH = /w03
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl063b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 309051
         LOGPATH = /w04
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl063b
         SUTPORT = 80
         SUTTERM = 3600
         BASENO = 312801
         LOGPATH = /w05
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl064a
         SUTPORT = 80
         SUTTERM = 150
         BASENO = 429401
         LOGPATH = /w05
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl064a
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 429551
         LOGPATH = /w06
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl064a
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 433301
         LOGPATH = /w07
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl064a
         SUTPORT = 80
         SUTTERM = 3650
         BASENO = 437051
         LOGPATH = /w08
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl064b
         SUTPORT = 80
         SUTTERM = 100
         BASENO = 440701
         LOGPATH = /w08
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl064b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 440801
         LOGPATH = /w09
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl064b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 444551
         LOGPATH = /w10
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl064b
         SUTPORT = 80
         SUTTERM = 3700
         BASENO = 448301
         LOGPATH = /w11
         LOGLEVEL = 0
      ENDSUT
   ENDRTE
   STARTVARIABLE
      WAREHOUSE = 108480
      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 = 723201
         LOGPATH = /w00
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl111a
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 727201
         LOGPATH = /w01
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl111a
         SUTPORT = 80
         SUTTERM = 3550
         BASENO = 730951
         LOGPATH = /w02
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl111b
         SUTPORT = 80
         SUTTERM = 200
         BASENO = 734501
         LOGPATH = /w02
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl111b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 734701
         LOGPATH = /w03
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl111b
         SUTPORT = 80
         SUTTERM = 3750
```

```
    BASENO = 738451
    LOGPATH = /w04
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl111b
    SUTPORT = 80
    SUTTERM = 3600
    BASENO = 742201
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl112a
    SUTPORT = 80
    SUTTERM = 150
    BASENO = 858801
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl112a
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 858951
    LOGPATH = /w06
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl112a
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 862701
    LOGPATH = /w07
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl112a
    SUTPORT = 80
    SUTTERM = 3650
    BASENO = 866451
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl112b
    SUTPORT = 80
    SUTTERM = 100
    BASENO = 870101
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl112b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 870201
    LOGPATH = /w09
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl112b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 873951
    LOGPATH = /w10
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl112b
    SUTPORT = 80
    SUTTERM = 3700
    BASENO = 877701
    LOGPATH = /w11
    LOGLEVEL = 0
    ENDSUT
  ENDRTE
  STARTVARIABLE
    WAREHOUSE = 108480
    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


::::::::::::::::::::::::::::::::::::
 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 = 994401
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl113a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 998401
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl113a
      SUTPORT = 80
      SUTTERM = 3550
      BASENO = 1002151
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl113b
      SUTPORT = 80
      SUTTERM = 200
      BASENO = 1005701
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl113b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 1005901
      LOGPATH = /w03
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl113b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 1009651
      LOGPATH = /w04
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl113b
      SUTPORT = 80
      SUTTERM = 3600
      BASENO = 1013401
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl114a
      SUTPORT = 80
      SUTTERM = 150
      BASENO = 67801
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl114a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 67951
      LOGPATH = /w06
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl114a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 71701
      LOGPATH = /w07
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl114a
      SUTPORT = 80
      SUTTERM = 3650
      BASENO = 75451
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl114b
      SUTPORT = 80
      SUTTERM = 100
      BASENO = 79101
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl114b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 79201
      LOGPATH = /w09
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl114b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 82951
      LOGPATH = /w10
```

```
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl114b
        SUTPORT = 80
        SUTTERM = 3700
        BASENO = 86701
        LOGPATH = /w11
        LOGLEVEL = 0
      ENDSUT
    ENDRTE
    STARTVARIABLE
      WAREHOUSE = 108480
      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 = 203401
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl115a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 207401
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl115a
      SUTPORT = 80
      SUTTERM = 3550
      BASENO = 211151
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl115b
      SUTPORT = 80
      SUTTERM = 200
      BASENO = 214701
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl115b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 214901
      LOGPATH = /w03
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl115b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 218651
      LOGPATH = /w04
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl115b
      SUTPORT = 80
      SUTTERM = 3600
      BASENO = 222401
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl116a
      SUTPORT = 80
      SUTTERM = 150
      BASENO = 339001
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl116a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 339151
      LOGPATH = /w06
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl116a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 342901
      LOGPATH = /w07
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl116a
      SUTPORT = 80
      SUTTERM = 3650
      BASENO = 346651
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl116b
      SUTPORT = 80
      SUTTERM = 100
      BASENO = 350301
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl116b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 350401
      LOGPATH = /w09
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl116b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 354151
      LOGPATH = /w10
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl116b
      SUTPORT = 80
      SUTTERM = 3700
      BASENO = 357901
      LOGPATH = /w11
      LOGLEVEL = 0
    ENDSUT
  ENDRTE
  STARTVARIABLE
    WAREHOUSE = 108480
    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 = 474601
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl117a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 478601
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
```

```
    SUTHOST = cl117a
    SUTPORT = 80
    SUTTERM = 3550
    BASENO = 482351
    LOGPATH = /w02
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl117b
    SUTPORT = 80
    SUTTERM = 200
    BASENO = 485901
    LOGPATH = /w02
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl117b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 486101
    LOGPATH = /w03
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl117b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 489851
    LOGPATH = /w04
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl117b
    SUTPORT = 80
    SUTTERM = 3600
    BASENO = 493601
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl118a
    SUTPORT = 80
    SUTTERM = 150
    BASENO = 610201
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl118a
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 610351
    LOGPATH = /w06
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl118a
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 614101
    LOGPATH = /w07
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl118a
    SUTPORT = 80
    SUTTERM = 3650
    BASENO = 617851
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl118b
    SUTPORT = 80
    SUTTERM = 100
    BASENO = 621501
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl118b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 621601
    LOGPATH = /w09
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl118b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 625351
    LOGPATH = /w10
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl118b
    SUTPORT = 80
    SUTTERM = 3700
    BASENO = 629101
    LOGPATH = /w11
    LOGLEVEL = 0
  ENDSUT
 ENDRTE
 STARTVARIABLE
  WAREHOUSE = 108480
  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 = 2000
    BASENO = 745801
    LOGPATH = /w00
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl119a
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 747801
    LOGPATH = /w01
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl119a
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 749801
    LOGPATH = /w02
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl119a
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 751801
    LOGPATH = /w03
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl119a
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 753801
    LOGPATH = /w04
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl119a
    SUTPORT = 80
    SUTTERM = 1300
    BASENO = 755801
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl119b
    SUTPORT = 80
    SUTTERM = 700
    BASENO = 757101
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl119b
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 757801
    LOGPATH = /w06
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl119b
    SUTPORT = 80
    SUTTERM = 1750
    BASENO = 759801
    LOGPATH = /w07
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl119b
    SUTPORT = 80
    SUTTERM = 1750
    BASENO = 761551
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl119b
    SUTPORT = 80
```

```
        SUTTERM = 1750
        BASENO = 763301
        LOGPATH = /w09
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl119b
        SUTPORT = 80
        SUTTERM = 1750
        BASENO = 765051
        LOGPATH = /w10
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl119b
        SUTPORT = 80
        SUTTERM = 1600
        BASENO = 766801
        LOGPATH = /w11
        LOGLEVEL = 0
      ENDSUT
    ENDRTE
    STARTVARIABLE
      WAREHOUSE = 108480
      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


::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::
 rte39.conf
::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::


#
#   rte39.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
   STARTRTE
      STARTSUT
        SUTHOST = cl121a
        SUTPORT = 80
        SUTTERM = 4000
        BASENO = 881401
        LOGPATH = /w00
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl121a
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 885401
        LOGPATH = /w01
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl121a
        SUTPORT = 80
        SUTTERM = 3550
        BASENO = 889151
        LOGPATH = /w02
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl121b
        SUTPORT = 80
        SUTTERM = 200
        BASENO = 892701
        LOGPATH = /w02
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl121b
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 892901
        LOGPATH = /w03
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl121b
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 896651
        LOGPATH = /w04
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl121b
        SUTPORT = 80
        SUTTERM = 3600
        BASENO = 900401
        LOGPATH = /w05
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl122a
        SUTPORT = 80
        SUTTERM = 150
        BASENO = 1017001
        LOGPATH = /w05
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl122a
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 1017151
        LOGPATH = /w06
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl122a
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 1020901
        LOGPATH = /w07
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl122a
        SUTPORT = 80
        SUTTERM = 3650
        BASENO = 1024651
        LOGPATH = /w08
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl122b
        SUTPORT = 80
        SUTTERM = 100
        BASENO = 1028301
        LOGPATH = /w08
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl122b
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 1028401
        LOGPATH = /w09
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl122b
        SUTPORT = 80
        SUTTERM = 3750
        BASENO = 1032151
        LOGPATH = /w10
        LOGLEVEL = 0
      ENDSUT
      STARTSUT
        SUTHOST = cl122b
        SUTPORT = 80
        SUTTERM = 3700
        BASENO = 1035901
        LOGPATH = /w11
        LOGLEVEL = 0
      ENDSUT
    ENDRTE
    STARTVARIABLE
      WAREHOUSE = 108480
      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 = cl124a
        SUTPORT = 80
        SUTTERM = 2000
        BASENO = 90401
        LOGPATH = /w00
```

```
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl124a
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 92401
    LOGPATH = /w01
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl124a
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 94401
    LOGPATH = /w02
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl124a
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 96401
    LOGPATH = /w03
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl124a
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 98401
    LOGPATH = /w04
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl124a
    SUTPORT = 80
    SUTTERM = 1300
    BASENO = 100401
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl124b
    SUTPORT = 80
    SUTTERM = 700
    BASENO = 101701
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl124b
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 102401
    LOGPATH = /w06
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl124b
    SUTPORT = 80
    SUTTERM = 1750
    BASENO = 104401
    LOGPATH = /w07
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl124b
    SUTPORT = 80
    SUTTERM = 1750
    BASENO = 106151
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl124b
    SUTPORT = 80
    SUTTERM = 1750
    BASENO = 107901
    LOGPATH = /w09
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl124b
    SUTPORT = 80
    SUTTERM = 1750
    BASENO = 109651
    LOGPATH = /w10
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl124b
    SUTPORT = 80
    SUTTERM = 1600
    BASENO = 111401
    LOGPATH = /w11
    LOGLEVEL = 0
  ENDSUT
 ENDRTE
 STARTVARIABLE
  WAREHOUSE = 108480
  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 = 226001
    LOGPATH = /w00
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl125a
    SUTPORT = 80
    SUTTERM = 1750
    BASENO = 230001
    LOGPATH = /w01
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl125a
    SUTPORT = 80
    SUTTERM = 3550
    BASENO = 233751
    LOGPATH = /w02
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl125b
    SUTPORT = 80
    SUTTERM = 200
    BASENO = 237301
    LOGPATH = /w02
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl125b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 237501
    LOGPATH = /w03
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl125b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 241251
    LOGPATH = /w04
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl125b
    SUTPORT = 80
    SUTTERM = 3600
    BASENO = 245001
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl126a
    SUTPORT = 80
    SUTTERM = 150
    BASENO = 361601
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl126a
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 361751
    LOGPATH = /w06
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl126a
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 365501
    LOGPATH = /w07
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl126a
    SUTPORT = 80
    SUTTERM = 3650
    BASENO = 369251
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
```

```
    STARTSUT
      SUTHOST = cl126b
      SUTPORT = 80
      SUTTERM = 100
      BASENO = 372901
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl126b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 373001
      LOGPATH = /w09
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl126b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 376751
      LOGPATH = /w10
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl126b
      SUTPORT = 80
      SUTTERM = 3700
      BASENO = 380501
      LOGPATH = /w11
      LOGLEVEL = 0
    ENDSUT
  ENDRTE
  STARTVARIABLE
    WAREHOUSE = 108480
    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 = 497201
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl127a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 501201
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl127a
      SUTPORT = 80
      SUTTERM = 3550
      BASENO = 504951
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl127b
      SUTPORT = 80
      SUTTERM = 200
      BASENO = 508501
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl127b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 508701
      LOGPATH = /w03
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl127b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 512451
      LOGPATH = /w04
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl127b
      SUTPORT = 80
      SUTTERM = 3600
      BASENO = 516201
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl128a
      SUTPORT = 80
      SUTTERM = 150
      BASENO = 632801
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl128a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 632951
      LOGPATH = /w06
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl128a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 636701
      LOGPATH = /w07
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl128a
      SUTPORT = 80
      SUTTERM = 3650
      BASENO = 640451
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl128b
      SUTPORT = 80
      SUTTERM = 100
      BASENO = 644101
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl128b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 644201
      LOGPATH = /w09
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl128b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 647951
      LOGPATH = /w10
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl128b
      SUTPORT = 80
      SUTTERM = 3700
      BASENO = 651701
      LOGPATH = /w11
      LOGLEVEL = 0
    ENDSUT
  ENDRTE
  STARTVARIABLE
    WAREHOUSE = 108480
    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 = 768401
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl129a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 772401
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl129a
      SUTPORT = 80
      SUTTERM = 3550
      BASENO = 776151
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl129b
      SUTPORT = 80
      SUTTERM = 200
      BASENO = 779701
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl129b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 779901
      LOGPATH = /w03
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl129b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 783651
      LOGPATH = /w04
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl129b
      SUTPORT = 80
      SUTTERM = 3600
      BASENO = 787401
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl130a
      SUTPORT = 80
      SUTTERM = 150
      BASENO = 904001
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl130a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 904151
      LOGPATH = /w06
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl130a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 907901
      LOGPATH = /w07
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl130a
      SUTPORT = 80
      SUTTERM = 3650
      BASENO = 911651
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl130b
      SUTPORT = 80
      SUTTERM = 100
      BASENO = 915301
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl130b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 915401
      LOGPATH = /w09
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl130b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 919151
      LOGPATH = /w10
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl130b
      SUTPORT = 80
      SUTTERM = 3700
      BASENO = 922901
      LOGPATH = /w11
      LOGLEVEL = 0
    ENDSUT
  ENDRTE
  STARTVARIABLE
    WAREHOUSE = 108480
    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 = 1039601
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl131a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 1043601
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl131a
      SUTPORT = 80
      SUTTERM = 3550
      BASENO = 1047351
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl131b
      SUTPORT = 80
      SUTTERM = 200
      BASENO = 1050901
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl131b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 1051101
      LOGPATH = /w03
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl131b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 1054851
      LOGPATH = /w04
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl131b
      SUTPORT = 80
      SUTTERM = 3600
      BASENO = 1058601
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl132a
      SUTPORT = 80
      SUTTERM = 150
      BASENO = 113001
```

```
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl132a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 113151
      LOGPATH = /w06
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl132a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 116901
      LOGPATH = /w07
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl132a
      SUTPORT = 80
      SUTTERM = 3650
      BASENO = 120651
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl132b
      SUTPORT = 80
      SUTTERM = 100
      BASENO = 124301
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl132b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 124401
      LOGPATH = /w09
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl132b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 128151
      LOGPATH = /w10
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl132b
      SUTPORT = 80
      SUTTERM = 3700
      BASENO = 131901
      LOGPATH = /w11
      LOGLEVEL = 0
    ENDSUT
  ENDRTE
  STARTVARIABLE
    WAREHOUSE = 108480
    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 = 248601
      LOGPATH = /w00
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl133a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 252601
      LOGPATH = /w01
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl133a
      SUTPORT = 80
      SUTTERM = 3550
      BASENO = 256351
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl133b
      SUTPORT = 80
      SUTTERM = 200
      BASENO = 259901
      LOGPATH = /w02
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl133b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 260101
      LOGPATH = /w03
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl133b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 263851
      LOGPATH = /w04
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl133b
      SUTPORT = 80
      SUTTERM = 3600
      BASENO = 267601
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl134a
      SUTPORT = 80
      SUTTERM = 150
      BASENO = 384201
      LOGPATH = /w05
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl134a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 384351
      LOGPATH = /w06
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl134a
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 388101
      LOGPATH = /w07
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl134a
      SUTPORT = 80
      SUTTERM = 3650
      BASENO = 391851
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl134b
      SUTPORT = 80
      SUTTERM = 100
      BASENO = 395501
      LOGPATH = /w08
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl134b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 395601
      LOGPATH = /w09
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl134b
      SUTPORT = 80
      SUTTERM = 3750
      BASENO = 399351
      LOGPATH = /w10
      LOGLEVEL = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl134b
      SUTPORT = 80
      SUTTERM = 3700
      BASENO = 403101
      LOGPATH = /w11
      LOGLEVEL = 0
    ENDSUT
  ENDRTE
  STARTVARIABLE
    WAREHOUSE = 108480
    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 = 519801
         LOGPATH = /w00
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl135a
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 523801
         LOGPATH = /w01
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl135a
         SUTPORT = 80
         SUTTERM = 3550
         BASENO = 527551
         LOGPATH = /w02
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl135b
         SUTPORT = 80
         SUTTERM = 200
         BASENO = 531101
         LOGPATH = /w02
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl135b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 531301
         LOGPATH = /w03
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl135b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 535051
         LOGPATH = /w04
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl135b
         SUTPORT = 80
         SUTTERM = 3600
         BASENO = 538801
         LOGPATH = /w05
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl136a
         SUTPORT = 80
         SUTTERM = 150
         BASENO = 655401
         LOGPATH = /w05
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl136a
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 655551
         LOGPATH = /w06
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl136a
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 659301
         LOGPATH = /w07
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl136a
         SUTPORT = 80
         SUTTERM = 3650
         BASENO = 663051
         LOGPATH = /w08
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl136b
         SUTPORT = 80
         SUTTERM = 100
         BASENO = 666701
         LOGPATH = /w08
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl136b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 666801
         LOGPATH = /w09
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl136b
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 670551
         LOGPATH = /w10
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl136b
         SUTPORT = 80
         SUTTERM = 3700
         BASENO = 674301
         LOGPATH = /w11
         LOGLEVEL = 0
      ENDSUT
   ENDRTE
   STARTVARIABLE
      WAREHOUSE = 108480
      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 = 791001
         LOGPATH = /w00
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl137a
         SUTPORT = 80
         SUTTERM = 3750
         BASENO = 795001
         LOGPATH = /w01
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl137a
         SUTPORT = 80
         SUTTERM = 3550
         BASENO = 798751
         LOGPATH = /w02
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl137b
         SUTPORT = 80
         SUTTERM = 200
         BASENO = 802301
         LOGPATH = /w02
         LOGLEVEL = 0
      ENDSUT
      STARTSUT
         SUTHOST = cl137b
```

```
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 802501
    LOGPATH = /w03
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl137b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 806251
    LOGPATH = /w04
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl137b
    SUTPORT = 80
    SUTTERM = 3600
    BASENO = 810001
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl138a
    SUTPORT = 80
    SUTTERM = 150
    BASENO = 926601
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl138a
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 926751
    LOGPATH = /w06
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl138a
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 930501
    LOGPATH = /w07
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl138a
    SUTPORT = 80
    SUTTERM = 3650
    BASENO = 934251
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl138b
    SUTPORT = 80
    SUTTERM = 100
    BASENO = 937901
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl138b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 938001
    LOGPATH = /w09
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl138b
    SUTPORT = 80
    SUTTERM = 3750
    BASENO = 941751
    LOGPATH = /w10
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl138b
    SUTPORT = 80
    SUTTERM = 3700
    BASENO = 945501
    LOGPATH = /w11
    LOGLEVEL = 0
  ENDSUT
 ENDRTE
 STARTVARIABLE
  WAREHOUSE = 108480
  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 = 2000
    BASENO = 1062201
    LOGPATH = /w00
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl139a
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 1064201
    LOGPATH = /w01
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl139a
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 1066201
    LOGPATH = /w02
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl139a
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 1068201
    LOGPATH = /w03
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl139a
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 1070201
    LOGPATH = /w04
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl139a
    SUTPORT = 80
    SUTTERM = 1300
    BASENO = 1072201
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl139b
    SUTPORT = 80
    SUTTERM = 700
    BASENO = 1073501
    LOGPATH = /w05
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl139b
    SUTPORT = 80
    SUTTERM = 2000
    BASENO = 1074201
    LOGPATH = /w06
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl139b
    SUTPORT = 80
    SUTTERM = 1750
    BASENO = 1076201
    LOGPATH = /w07
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl139b
    SUTPORT = 80
    SUTTERM = 1750
    BASENO = 1077951
    LOGPATH = /w08
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl139b
    SUTPORT = 80
    SUTTERM = 1750
    BASENO = 1079701
    LOGPATH = /w09
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl139b
    SUTPORT = 80
    SUTTERM = 1750
    BASENO = 1081451
    LOGPATH = /w10
    LOGLEVEL = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl139b
    SUTPORT = 80
    SUTTERM = 1600
```

```
        BASENO = 1083201
        LOGPATH = /w11
        LOGLEVEL = 0
      ENDSUT
    ENDRTE
    STARTVARIABLE
      WAREHOUSE = 108480
      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

[/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 = 0x4000000000
kernel.shmall = 0x20000000
fs.aio-max-nr = 5242880

vm.nr_hugepages = 3991

[/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)
- 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

oracle soft memlock 2147483648
oracle hard memlock 2147483648
oracle soft nofile 4096
oracle hard nofile 65536
oracle soft nproc 4095
oracle hard nproc 16384

End of file

[/etc/elilo.conf]
prompt
timeout=20
default=tpc-c
relocatable

image=vmlinuz-2.6.9-42.EL.oralargesmp
label=tpc-c
initrd=initrd-2.6.9-42.EL.oralargesmp.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"

[p_run.ora]
trace_enabled = false
control_files = (/ora_dev/control_001,/ora_dev/control_002)
processes = 2000
sessions = 2000
transactions = 2000
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
max_dump_file_size = 10M
db_cache_size = 10240M
db_keep_cache_size = 600000M
db_recycle_cache_size = 138752M
db_16k_cache_size = 244032M
db_8k_cache_size = 2048M
db_4k_cache_size = 512M
shared_pool_size = 24024M
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 = 1600
log_checkpoints_to_alert = true
timed_statistics = false
statistics_level = basic
query_rewrite_enabled = false
==========
==========
(cl039 configuration)
==========
==========

[OS tunables]

.....
.....
chkconfig
.....
.....

sendmail 0:off 1:off 2:off 3:off 4:off 5:off 6:off
xinetd 0:off 1:off 2:off 3:on 4:on 5:on 6:off
named 0:off 1:off 2:off 3:off 4:off 5:off 6:off
smartd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
syslog 0:off 1:off 2:on 3:on 4:on 5:on 6:off
radiusd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
rwhod 0:off 1:off 2:off 3:off 4:off 5:off 6:off
mdmonitor 0:off 1:off 2:off 3:off 4:off 5:off 6:off
ypbind 0:off 1:off 2:off 3:off 4:off 5:off 6:off

nscd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
isdn 0:off 1:off 2:off 3:off 4:off 5:off 6:off
arptables_jf 0:off 1:off 2:off 3:off 4:off 5:off 6:off
lisa 0:off 1:off 2:off 3:off 4:off 5:off 6:off
rusersd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
dhcp6s 0:off 1:off 2:off 3:off 4:off 5:off 6:off
cyrus-imapd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
winbind 0:off 1:off 2:off 3:off 4:off 5:off 6:off
vncserver 0:off 1:off 2:off 3:off 4:off 5:off 6:off
amd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
gpm 0:off 1:off 2:off 3:off 4:off 5:off 6:off
apmd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
bgpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
readahead 0:off 1:off 2:off 3:off 4:off 5:on 6:off
ypxfrd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
mysqld 0:off 1:off 2:off 3:off 4:off 5:off 6:off
mailman 0:off 1:off 2:off 3:off 4:off 5:off 6:off
rpcgssd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
innd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
pcmcia 0:off 1:off 2:off 3:off 4:off 5:off 6:off
mdmpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
autofs 0:off 1:off 2:off 3:off 4:off 5:off 6:off
rawdevices 0:off 1:off 2:off 3:on 4:on 5:on 6:off
ip6tables 0:off 1:off 2:off 3:off 4:off 5:off 6:off
nfs 0:off 1:off 2:off 3:off 4:off 5:off 6:off
bluetooth 0:off 1:off 2:off 3:off 4:off 5:off 6:off
netdump-server 0:off 1:off 2:off 3:off 4:off 5:off 6:off
ripngd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
iptables 0:off 1:off 2:on 3:on 4:on 5:on 6:off
NetworkManager 0:off 1:off 2:off 3:off 4:off 5:off 6:off
rpcsvcgssd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
dhcrelay 0:off 1:off 2:off 3:off 4:off 5:off 6:off
bootparamd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
squid 0:off 1:off 2:off 3:off 4:off 5:off 6:off
diskdump 0:off 1:off 2:off 3:off 4:off 5:off 6:off
haldaemon 0:off 1:off 2:off 3:off 4:off 5:off 6:off
cups 0:off 1:off 2:off 3:off 4:off 5:off 6:off
yppasswdd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
saslauthd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
netplugd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
snmptrapd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
canna 0:off 1:off 2:off 3:off 4:off 5:off 6:off
readahead_early 0:off 1:off 2:off 3:off 4:off 5:on 6:off
kprop 0:off 1:off 2:off 3:off 4:off 5:off 6:off
ripd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
irqbalance 0:off 1:off 2:off 3:on 4:on 5:on 6:off
messagebus 0:off 1:off 2:off 3:off 4:off 5:off 6:off
kudzu 0:off 1:off 2:off 3:off 4:off 5:off 6:off
ldap 0:off 1:off 2:off 3:off 4:off 5:off 6:off
microcode_ctl 0:off 1:off 2:off 3:off 4:off 5:off 6:off
network 0:off 1:off 2:on 3:on 4:on 5:on 6:off
rstatd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
dhcpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
portmap 0:off 1:off 2:off 3:off 4:off 5:off 6:off
lm_sensors 0:off 1:off 2:off 3:off 4:off 5:off 6:off
atd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
ntpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
krb524 0:off 1:off 2:off 3:off 4:off 5:off 6:off
smb 0:off 1:off 2:off 3:off 4:off 5:off 6:off
httpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
rpcidmapd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
krb5kdc 0:off 1:off 2:off 3:off 4:off 5:off 6:off
anacron 0:off 1:off 2:off 3:off 4:off 5:off 6:off
ospf6d 0:off 1:off 2:off 3:off 4:off 5:off 6:off
cpuspeed 0:off 1:on 2:on 3:on 4:on 5:on 6:off
nfslock 0:off 1:off 2:off 3:off 4:off 5:off 6:off
dc_client 0:off 1:off 2:off 3:off 4:off 5:off 6:off
dovecot 0:off 1:off 2:off 3:off 4:off 5:off 6:off
sshd 0:off 1:off 2:on 3:on 4:on 5:on 6:off
psacct 0:off 1:off 2:off 3:off 4:off 5:off 6:off
hpoj 0:off 1:off 2:off 3:off 4:off 5:off 6:off
radvd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
ypserv 0:off 1:off 2:off 3:off 4:off 5:off 6:off
iiim 0:off 1:off 2:off 3:off 4:off 5:off 6:off
netdump 0:off 1:off 2:off 3:off 4:off 5:off 6:off
ospfd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
cups-config-daemon 0:off 1:off 2:off 3:off 4:off 5:off 6:off
snmpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
acpid 0:off 1:off 2:off 3:off 4:off 5:off 6:off
dc_server 0:off 1:off 2:off 3:off 4:off 5:off 6:off
sysstat 0:off 1:on 2:off 3:off 4:off 5:off 6:off
kadmin 0:off 1:off 2:off 3:off 4:off 5:off 6:off
xfs 0:off 1:off 2:off 3:off 4:off 5:off 6:off
arpwatch 0:off 1:off 2:off 3:off 4:off 5:off 6:off
netfs 0:off 1:off 2:off 3:off 4:off 5:off 6:off
spamassassin 0:off 1:off 2:off 3:off 4:off 5:off 6:off
FreeWnn 0:off 1:off 2:off 3:off 4:off 5:off 6:off
tux 0:off 1:off 2:off 3:off 4:off 5:off 6:off
crond 0:off 1:off 2:on 3:on 4:on 5:on 6:off
vsftpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
rhnsd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
irda 0:off 1:off 2:off 3:off 4:off 5:off 6:off
postgresql 0:off 1:off 2:off 3:off 4:off 5:off 6:off
zebra 0:off 1:off 2:off 3:off 4:off 5:off 6:off

xinetd based services:
- talk: off
- daytime: off
- kshell: off
- amandaidx: off
- amanda: off
- krb5-telnet: off
- auth: on
- telnet: on
- finger: off
- gssftp: off
- amidxtape: off
- dbskkd-cdb: off
- ntalk: off
- ktalk: off
- rsync: off
- time-udp: off
- echo: off
- echo-udp: off
- chargen-udp: off
- eklogin: off
- klogin: off
- rsh: on
- cups-lpd: off
- time: off
- rexec: off
- daytime-udp: off
- rlogin: on
- chargen: off
- swat: off
- tftp: off

::::::::::::::::::::::::::::::::::
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 filediscriptor
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:$ORACLE_
HOME/network/lib:$TUXDIR/lib

ulimit -u 30000
ulimit -s 1536

#/sbin/swapoff -a

# For 3tier tune
SVRAPL=`ps -e | grep tpccfmlw | 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 logfile 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        46
ServerLimit         46
ThreadLimit         500
MaxClients          23000
MinSpareThreads      1
MaxSpareThreads      23000
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 tpapl_module modules/mod_tpapl.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) - Luxembourgeois* (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
char set.
#
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
(SSI):
# (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="108480"
MaxUsers="1084800"
MaxTerm of Client="22600"
CONTROL_Flag="0"
LogPath="/home/tpc/log/userlog.log"

[SVRAPL_INFO]
LogPath="/home/tpc/log/DBDepend_Userlog.log
"



::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

#
# 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]
-------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

#
#   ubbconfig : 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
cl039   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

tpccfmlw    SRVGRP=group1 RQADDR=ware01
SRVID=1 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware02
SRVID=2 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware03
SRVID=3 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware04
SRVID=4 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware05
SRVID=5 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware06
SRVID=6 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware07
SRVID=7 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware08
SRVID=8 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware09
SRVID=9 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
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"
tpccfmlw    SRVGRP=group1 RQADDR=ware19
SRVID=19 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"

*SERVICES
"OPSTUXSERVER"  TRANTIME=0
SRVGRP=group1

*ROUTING


=====================================
======================
 (configuration difference between cl039 and
cl040)
=====================================
======================


[Front-end application tunables]
-------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="1"
---
> Term_Base="135601"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))


[TP monitor tunables]
-------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl039 LMID=SITE1
---
> cl040 LMID=SITE1


=====================================
======================
 (configuration difference between cl039 and
cl041)
=====================================
======================


[Front-end application tunables]
-------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="1"
---
> Term_Base="271201"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))


[TP monitor tunables]
-------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl039 LMID=SITE1
---
> cl041 LMID=SITE1


=====================================
======================
 (configuration difference between cl039 and
cl042)
=====================================
======================


[Front-end application tunables]
-------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="1"
---
> Term_Base="406801"
```

```
::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl039 LMID=SITE1
---
> cl042 LMID=SITE1


====================================
=====================
 (configuration difference between cl039 and
cl043)
====================================
=====================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="1"
---
> Term_Base="542401"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl039 LMID=SITE1
---
> cl043 LMID=SITE1


====================================
=====================
 (configuration difference between cl039 and
cl044)
====================================
=====================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="1"
---
> Term_Base="678001"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl039 LMID=SITE1
---
> cl044 LMID=SITE1


====================================
=====================
 (configuration difference between cl039 and
cl045)
====================================
=====================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="1"
---
> Term_Base="813601"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl039 LMID=SITE1
---
> cl045 LMID=SITE1


====================================
=====================
 (configuration difference between cl039 and
cl046)
====================================
=====================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="1"
---
> Term_Base="949201"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl039 LMID=SITE1
---
> cl046 LMID=SITE1


====================================
=====================
 (cl061 configuration)
====================================
=====================


[OS tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 chkconfig
::::::::::::::::::::::::::::::::
```

```
lisa            0:off  1:off  2:off  3:off  4:off  5:off
6:off
isdn            0:off  1:off  2:off  3:off  4:off  5:off
6:off
cups            0:off  1:off  2:off  3:off  4:off  5:off
6:off
xinetd          0:off  1:off  2:off  3:on   4:on
5:on   6:off
postgresql      0:off  1:off  2:off  3:off  4:off
5:off  6:off
squid           0:off  1:off  2:off  3:off  4:off  5:off
6:off
ospfd           0:off  1:off  2:off  3:off  4:off  5:off
6:off
psacct          0:off  1:off  2:off  3:off  4:off  5:off
6:off
mdmonitor       0:off  1:off  2:off  3:off  4:off
5:off  6:off
iptables        0:off  1:off  2:on   3:on   4:on
5:on   6:off
network         0:off  1:off  2:on   3:on   4:on
5:on   6:off
arptables_jf    0:off  1:off  2:off  3:off  4:off
5:off  6:off
dhcpd           0:off  1:off  2:off  3:off  4:off  5:off
6:off
messagebus      0:off  1:off  2:off  3:off  4:off
5:off  6:off
ripd            0:off  1:off  2:off  3:off  4:off  5:off
6:off
dc_client       0:off  1:off  2:off  3:off  4:off  5:off
6:off
ypserv          0:off  1:off  2:off  3:off  4:off  5:off
6:off
rusersd         0:off  1:off  2:off  3:off  4:off  5:off
6:off
sendmail        0:off  1:off  2:off  3:off  4:off
5:off  6:off
winbind         0:off  1:off  2:off  3:off  4:off  5:off
6:off
sysstat         0:off  1:on   2:off  3:off  4:off  5:off
6:off
vsftpd          0:off  1:off  2:off  3:off  4:off  5:off
6:off
kudzu           0:off  1:off  2:off  3:off  4:off  5:off
6:off
rwhod           0:off  1:off  2:off  3:off  4:off  5:off
6:off
portmap         0:off  1:off  2:off  3:off  4:off
5:off  6:off
smartd          0:off  1:off  2:off  3:off  4:off  5:off
6:off
dhcrelay        0:off  1:off  2:off  3:off  4:off
5:off  6:off
mailman         0:off  1:off  2:off  3:off  4:off
5:off  6:off
mysqld          0:off  1:off  2:off  3:off  4:off
5:off  6:off
anacron         0:off  1:off  2:off  3:off  4:off
5:off  6:off
netdump-server  0:off  1:off  2:off  3:off  4:off
5:off  6:off
atd             0:off  1:off  2:off  3:off  4:off  5:off
6:off
nscd            0:off  1:off  2:off  3:off  4:off  5:off
6:off
kadmin          0:off  1:off  2:off  3:off  4:off
5:off  6:off
sshd            0:off  1:off  2:on   3:on   4:on
5:on   6:off
gpm             0:off  1:off  2:off  3:off  4:off  5:off
6:off
innd            0:off  1:off  2:off  3:off  4:off  5:off
6:off
cpuspeed        0:off  1:on   2:on   3:on   4:on
5:on   6:off
named           0:off  1:off  2:off  3:off  4:off
5:off  6:off
snmptrapd       0:off  1:off  2:off  3:off  4:off
5:off  6:off
dc_server       0:off  1:off  2:off  3:off  4:off
5:off  6:off
acpid           0:off  1:off  2:off  3:off  4:off  5:off
6:off
lm_sensors      0:off  1:off  2:off  3:off  4:off
5:off  6:off
kprop           0:off  1:off  2:off  3:off  4:off  5:off
6:off
ip6tables       0:off  1:off  2:off  3:off  4:off
5:off  6:off
cyrus-imapd     0:off  1:off  2:off  3:off  4:off
5:off  6:off
rstatd          0:off  1:off  2:off  3:off  4:off  5:off
6:off
nfs             0:off  1:off  2:off  3:off  4:off  5:off
6:off
cups-config-daemon 0:off  1:off  2:off  3:off
4:off  5:off  6:off
haldaemon       0:off  1:off  2:off  3:off  4:off
5:off  6:off
krb5kdc         0:off  1:off  2:off  3:off  4:off
5:off  6:off
netfs           0:off  1:off  2:off  3:off  4:off  5:off
6:off
ospf6d          0:off  1:off  2:off  3:off  4:off  5:off
6:off
irda            0:off  1:off  2:off  3:off  4:off  5:off
6:off
spamassassin    0:off  1:off  2:off  3:off  4:off
5:off  6:off
pcmcia          0:off  1:off  2:off  3:off  4:off
5:off  6:off
irqbalance      0:off  1:off  2:off  3:on   4:on
5:on   6:off
FreeWnn         0:off  1:off  2:off  3:off  4:off
5:off  6:off
iiim            0:off  1:off  2:off  3:off  4:off  5:off
6:off
bootparamd      0:off  1:off  2:off  3:off  4:off
5:off  6:off
nfslock         0:off  1:off  2:off  3:off  4:off  5:off
6:off
hpoj            0:off  1:off  2:off  3:off  4:off  5:off
6:off
rhnsd           0:off  1:off  2:off  3:off  4:off  5:off
6:off
zebra           0:off  1:off  2:off  3:off  4:off  5:off
6:off
httpd           0:off  1:off  2:off  3:off  4:off  5:off
6:off
dovecot         0:off  1:off  2:off  3:off  4:off
5:off  6:off
syslog          0:off  1:off  2:on   3:on   4:on
5:on   6:off
bgpd            0:off  1:off  2:off  3:off  4:off  5:off
6:off
bluetooth       0:off  1:off  2:off  3:off  4:off
5:off  6:off
netplugd        0:off  1:off  2:off  3:off  4:off
5:off  6:off
rpcgssd         0:off  1:off  2:off  3:off  4:off
5:off  6:off
saslauthd       0:off  1:off  2:off  3:off  4:off
5:off  6:off
vncserver       0:off  1:off  2:off  3:off  4:off
5:off  6:off
radiusd         0:off  1:off  2:off  3:off  4:off  5:off
6:off
diskdump        0:off  1:off  2:off  3:off  4:off
5:off  6:off
ypxfrd          0:off  1:off  2:off  3:off  4:off  5:off
6:off
autofs          0:off  1:off  2:off  3:off  4:off  5:off
6:off
snmpd           0:off  1:off  2:off  3:off  4:off
5:off  6:off
readahead_early 0:off  1:off  2:off  3:off  4:off
5:on   6:off
tux             0:off  1:off  2:off  3:off  4:off  5:off
6:off
ripngd          0:off  1:off  2:off  3:off  4:off  5:off
6:off
ypbind          0:off  1:off  2:off  3:off  4:off  5:off
6:off
netdump         0:off  1:off  2:off  3:off  4:off
5:off  6:off
ntpd            0:off  1:off  2:off  3:off  4:off  5:off
6:off
crond           0:off  1:off  2:on   3:on   4:on
5:on   6:off
dhcp6s          0:off  1:off  2:off  3:off  4:off
5:off  6:off
smb             0:off  1:off  2:off  3:off  4:off  5:off
6:off
canna           0:off  1:off  2:off  3:off  4:off  5:off
6:off
amd             0:off  1:off  2:off  3:off  4:off  5:off
6:off
rawdevices      0:off  1:off  2:off  3:on   4:on
5:on   6:off
rpcsvcgssd      0:off  1:off  2:off  3:off  4:off
5:off  6:off
xfs             0:off  1:off  2:off  3:off  4:off  5:off
6:off
radvd           0:off  1:off  2:off  3:off  4:off  5:off
6:off
ldap            0:off  1:off  2:off  3:off  4:off  5:off
6:off
krb524          0:off  1:off  2:off  3:off  4:off  5:off
6:off
readahead       0:off  1:off  2:off  3:off  4:off
5:on   6:off
mdmpd           0:off  1:off  2:off  3:off  4:off
5:off  6:off
yppasswdd       0:off  1:off  2:off  3:off  4:off
5:off  6:off
NetworkManager  0:off  1:off  2:off  3:off  4:off
5:off  6:off
rpcidmapd       0:off  1:off  2:off  3:off  4:off
5:off  6:off
arpwatch        0:off  1:off  2:off  3:off  4:off
5:off  6:off
apmd            0:off  1:off  2:off  3:off  4:off  5:off
6:off
microcode_ctl   0:off  1:off  2:off  3:off  4:off
5:off  6:off
xinetd based services:
    amandaidx:  off
    talk:   off
    rsync:  off
    cups-lpd:   off
    time-udp:   off
    krb5-telnet:    off
    echo:   off
    dbskkd-cdb: off
    swat:   off
    auth:   on
    klogin: off
    gssftp: off
    rsh:    on
    kshell: off
    telnet: on
    daytime-udp:    off
```

```
chargen-udp:    off
amidxtape:  off
tftp:   off
rlogin: on
finger: off
daytime:    off
eklogin:    off
ntalk:  off
time:   off
ktalk:  off
rexec:  off
amanda: off
echo-udp:   off
chargen:    off
```

::::::::::::::::::::::::::::::::
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 filediscriptor
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:$ORACLE_
HOME/network/lib:$TUXDIR/lib

ulimit -u 30000
ulimit -s 1536

#/sbin/swapoff -a

# For 3tier tune
SVRAPL=`ps -e | grep tpccfmlw | 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 logfile 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         46
ServerLimit          46
ThreadLimit          500
MaxClients           23000
MinSpareThreads      1
MaxSpareThreads      23000
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 tpapl_module
modules/mod_tpapl.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#forb
idden
#
#<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) - Luxembourgeois* (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
char set.
#
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
(SSI):
# (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="22601"
NumWarehouses="108480"
MaxUsers="1084800"
MaxTerm of Client="22600"
CONTROL_Flag="0"
LogPath="/home/tpc/log/userlog.log"

[SVRAPL_INFO]
LogPath="/home/tpc/log/DBDepend_Userlog.log
"



::::::::::::::::::::::::::::::::
 tnsnames.ora
```

```
::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::

#
# 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]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

#
#   ubbconfig : 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
cl061   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

tpccfmlw    SRVGRP=group1 RQADDR=ware01
SRVID=1 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware02
SRVID=2 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware03
SRVID=3 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware04
SRVID=4 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware05
SRVID=5 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware06
SRVID=6 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware07
SRVID=7 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware08
SRVID=8 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware09
SRVID=9 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
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"
tpccfmlw    SRVGRP=group1 RQADDR=ware19
SRVID=19 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"

*SERVICES
"OPSTUXSERVER"  TRANTIME=0
SRVGRP=group1

*ROUTING


==========
==========
 (configuration difference between cl061 and
cl062)
==========
==========


[Front-end application tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="158201"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))


[TP monitor tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl062 LMID=SITE1


==========
==========
 (configuration difference between cl061 and
cl063)
==========
==========


[Front-end application tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="293801"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))


[TP monitor tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl063 LMID=SITE1


==========
==========
```

```
 (configuration difference between cl061 and
cl064)
==========================================
======================

[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="429401"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl064 LMID=SITE1


==========================================
======================
 (configuration difference between cl061 and
cl065)
==========================================
======================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="565001"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl065 LMID=SITE1


==========================================
======================
 (configuration difference between cl061 and
cl066)
==========================================
======================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="700601"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl066 LMID=SITE1


==========================================
======================
 (configuration difference between cl061 and
cl067)
==========================================
======================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="836201"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl067 LMID=SITE1


==========================================
======================
 (configuration difference between cl061 and
cl105)
==========================================
======================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="45201"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::



[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl105 LMID=SITE1


==========================================
======================
```

```
 (configuration difference between cl061 and
cl106)
==========================================
======================

[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="180801"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl106 LMID=SITE1


==========================================
======================
 (configuration difference between cl061 and
cl108)
==========================================
======================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="316401"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl108 LMID=SITE1


==========================================
======================
 (configuration difference between cl061 and
cl109)
==========================================
======================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="452001"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl109 LMID=SITE1


==========================================
======================
 (configuration difference between cl061 and
cl110)
==========================================
======================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="587601"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl110 LMID=SITE1


==========================================
======================
 (configuration difference between cl061 and
cl111)
==========================================
======================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="723201"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
```

```
> cl111 LMID=SITE1


==========
 (configuration difference between cl061 and cl112)
==========


[Front-end application tunables]
----------

::::::::::
 tpapl.conf
::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="858801"


::::::::::
 tnsnames.ora
::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_g)(Port= 1527))


[TP monitor tunables]
----------

::::::::::
 ubbconfig
::::::::::

20c20
< cl061 LMID=SITE1
---
> cl112 LMID=SITE1


==========
 (configuration difference between cl061 and cl113)
==========


[Front-end application tunables]
----------

::::::::::
 tpapl.conf
::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="994401"


::::::::::
 tnsnames.ora
::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_h)(Port= 1528))


[TP monitor tunables]
----------

::::::::::
 ubbconfig
::::::::::

20c20
< cl061 LMID=SITE1
---
> cl113 LMID=SITE1


==========
 (configuration difference between cl061 and cl114)
==========


[Front-end application tunables]
----------

::::::::::
 tpapl.conf
::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="67801"


::::::::::
 tnsnames.ora
::::::::::



[TP monitor tunables]
----------

::::::::::
 ubbconfig
::::::::::

20c20
< cl061 LMID=SITE1
---
> cl114 LMID=SITE1


==========
 (configuration difference between cl061 and cl115)
==========


[Front-end application tunables]
----------

::::::::::
 tpapl.conf
::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="203401"


::::::::::
 tnsnames.ora
::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_b)(Port= 1522))


[TP monitor tunables]
----------

::::::::::
 ubbconfig
::::::::::

20c20
< cl061 LMID=SITE1
---
> cl115 LMID=SITE1


==========
 (configuration difference between cl061 and cl116)
==========


[Front-end application tunables]
----------

::::::::::
 tpapl.conf
::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="339001"


::::::::::
 tnsnames.ora
::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_c)(Port= 1523))


[TP monitor tunables]
----------

::::::::::
 ubbconfig
::::::::::

20c20
< cl061 LMID=SITE1
---
> cl116 LMID=SITE1
```

```
==========
(configuration difference between cl061 and cl117)
==========

[Front-end application tunables]
----------

::::::::::
tpapl.conf
::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="474601"

::::::::::
tnsnames.ora
::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_d)(Port= 1524))

[TP monitor tunables]
----------

::::::::::
ubbconfig
::::::::::

20c20
< cl061 LMID=SITE1
---
> cl117 LMID=SITE1

==========
(configuration difference between cl061 and cl118)
==========

[Front-end application tunables]
----------

::::::::::
tpapl.conf
::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="610201"

::::::::::
tnsnames.ora
::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_e)(Port= 1525))

[TP monitor tunables]
----------

::::::::::
ubbconfig
::::::::::

20c20
< cl061 LMID=SITE1
---
> cl118 LMID=SITE1

==========
(configuration difference between cl061 and cl119)
==========

[Front-end application tunables]
----------

::::::::::
tpapl.conf
::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="745801"

::::::::::
tnsnames.ora
::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_f)(Port= 1526))

[TP monitor tunables]
----------

::::::::::
ubbconfig
::::::::::

20c20
< cl061 LMID=SITE1
---
> cl119 LMID=SITE1

==========
(configuration difference between cl061 and cl121)
==========

[Front-end application tunables]
----------

::::::::::
tpapl.conf
::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="881401"

::::::::::
tnsnames.ora
::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_g)(Port= 1527))

[TP monitor tunables]
----------

::::::::::
ubbconfig
::::::::::

20c20
< cl061 LMID=SITE1
---
> cl121 LMID=SITE1

==========
(configuration difference between cl061 and cl122)
==========

[Front-end application tunables]
----------

::::::::::
tpapl.conf
::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="1017001"

::::::::::
tnsnames.ora
::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_h)(Port= 1528))

[TP monitor tunables]
----------

::::::::::
ubbconfig
```

```
::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl122 LMID=SITE1


==========================================
======================
 (configuration difference between cl061 and
cl124)
==========================================
======================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="90401"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::



[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl124 LMID=SITE1


==========================================
======================
 (configuration difference between cl061 and
cl125)
==========================================
======================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="226001"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl125 LMID=SITE1


==========================================
======================
 (configuration difference between cl061 and
cl126)
==========================================
======================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="361601"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl126 LMID=SITE1


==========================================
======================
 (configuration difference between cl061 and
cl127)
==========================================
======================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="497201"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl127 LMID=SITE1


==========================================
======================
 (configuration difference between cl061 and
cl128)
==========================================
======================


[Front-end application tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="632801"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))


[TP monitor tunables]
------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
```

```
::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl128 LMID=SITE1


====================================
======================
 (configuration difference between cl061 and
cl129)
====================================
======================


[Front-end application tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="768401"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))


[TP monitor tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl129 LMID=SITE1


====================================
======================
 (configuration difference between cl061 and
cl130)
====================================
======================


[Front-end application tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="904001"
```

```
::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))


[TP monitor tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl130 LMID=SITE1


====================================
======================
 (configuration difference between cl061 and
cl131)
====================================
======================


[Front-end application tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="1039601"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))


[TP monitor tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl131 LMID=SITE1


====================================
======================
 (configuration difference between cl061 and
cl132)
```

```
====================================
======================


[Front-end application tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="113001"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::



[TP monitor tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl132 LMID=SITE1


====================================
======================
 (configuration difference between cl061 and
cl133)
====================================
======================


[Front-end application tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="248601"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))


[TP monitor tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::
```

```
20c20
< cl061 LMID=SITE1
---
> cl133 LMID=SITE1


================================================================
 (configuration difference between cl061 and cl134)
================================================================

[Front-end application tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="384201"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_c)(Port= 1523))


[TP monitor tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl134 LMID=SITE1


================================================================
 (configuration difference between cl061 and cl135)
================================================================

[Front-end application tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="519801"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_d)(Port= 1524))


[TP monitor tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl135 LMID=SITE1


================================================================
 (configuration difference between cl061 and cl136)
================================================================

[Front-end application tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="655401"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_e)(Port= 1525))


[TP monitor tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl136 LMID=SITE1


================================================================
 (configuration difference between cl061 and cl137)
================================================================

[Front-end application tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="791001"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_f)(Port= 1526))


[TP monitor tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl137 LMID=SITE1


================================================================
 (configuration difference between cl061 and cl138)
================================================================

[Front-end application tunables]
----------------------------------------------------------------

::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="926601"


::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::

13c13
<     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host= pqtpc_g)(Port= 1527))
```

```
[TP monitor tunables]
------------------------------------------------------------


::::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl138 LMID=SITE1


==========================================
=======================
 (configuration difference between cl061 and
cl139)
==========================================
=======================


[Front-end application tunables]
------------------------------------------------------------


::::::::::::::::::::::::::::::::::
 tpapl.conf
::::::::::::::::::::::::::::::::::

2c2
< Term_Base="22601"
---
> Term_Base="1062201"


::::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::::

13c13
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))


[TP monitor tunables]
------------------------------------------------------------


::::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::::

20c20
< cl061 LMID=SITE1
---
> cl139 LMID=SITE1


==========================================
=======================
 (cl069 configuration)
==========================================
=======================


[OS tunables]
------------------------------------------------------------


::::::::::::::::::::::::::::::::::
 chkconfig
::::::::::::::::::::::::::::::::::

ypxfrd        0:off  1:off  2:off  3:off  4:off  5:off
6:off
saslauthd     0:off  1:off  2:off  3:off  4:off
5:off  6:off
snmptrapd     0:off  1:off  2:off  3:off  4:off
5:off  6:off
squid         0:off  1:off  2:off  3:off  4:off  5:off
6:off
zebra         0:off  1:off  2:off  3:off  4:off  5:off
6:off
dovecot       0:off  1:off  2:off  3:off  4:off
5:off  6:off
bgpd          0:off  1:off  2:off  3:off  4:off  5:off
6:off
ibmasm        0:off  1:off  2:off  3:off  4:off
5:off  6:off
ldap          0:off  1:off  2:off  3:off  4:off  5:off
6:off
spamassassin  0:off  1:off  2:off  3:off  4:off
5:off  6:off
readahead_early 0:off  1:off  2:off  3:off  4:off
5:on   6:off
irqbalance    0:off  1:off  2:off  3:on   4:on
5:on   6:off
winbind       0:off  1:off  2:off  3:off  4:off  5:off
6:off
messagebus    0:off  1:off  2:off  3:off  4:off
5:off  6:off
mdmonitor     0:off  1:off  2:off  3:off  4:off
5:off  6:off
crond         0:off  1:off  2:on   3:on   4:on
5:on   6:off
nfslock       0:off  1:off  2:off  3:off  4:off  5:off
6:off
pcmcia        0:off  1:off  2:off  3:off  4:off
5:off  6:off
arpwatch      0:off  1:off  2:off  3:off  4:off
5:off  6:off
rpcidmapd     0:off  1:off  2:off  3:off  4:off
5:off  6:off
xinetd        0:off  1:off  2:off  3:on   4:on
5:on   6:off
multipathd    0:off  1:off  2:off  3:off  4:off
5:off  6:off
rarpd         0:off  1:off  2:off  3:off  4:off  5:off
6:off
xfs           0:off  1:off  2:off  3:off  4:off  5:off
6:off
ospf6d        0:off  1:off  2:off  3:off  4:off  5:off
6:off
iptables      0:off  1:off  2:on   3:on   4:on
5:on   6:off
ypbind        0:off  1:off  2:off  3:off  4:off  5:off
6:off
ntpd          0:off  1:off  2:off  3:off  4:off  5:off
6:off
haldaemon     0:off  1:off  2:off  3:off  4:off
5:off  6:off
snmpd         0:off  1:off  2:off  3:off  4:off
5:off  6:off
atd           0:off  1:off  2:off  3:off  4:off  5:off
6:off
NetworkManager 0:off  1:off  2:off  3:off  4:off
5:off  6:off
irda          0:off  1:off  2:off  3:off  4:off  5:off
6:off
krb5kdc       0:off  1:off  2:off  3:off  4:off
5:off  6:off
yppasswdd     0:off  1:off  2:off  3:off  4:off
5:off  6:off
gpm           0:off  1:off  2:off  3:off  4:off  5:off
6:off
rusersd       0:off  1:off  2:off  3:off  4:off  5:off
6:off
canna         0:off  1:off  2:off  3:off  4:off  5:off
6:off
mysqld        0:off  1:off  2:off  3:off  4:off
5:off  6:off
sshd          0:off  1:off  2:on   3:on   4:on
5:on   6:off
kadmin        0:off  1:off  2:off  3:off  4:off
5:off  6:off
postgresql    0:off  1:off  2:off  3:off  4:off
5:off  6:off
acpid         0:off  1:off  2:off  3:off  4:off  5:off
6:off
hpoj          0:off  1:off  2:off  3:off  4:off  5:off
6:off
apmd          0:off  1:off  2:off  3:off  4:off  5:off
6:off
dhcrelay      0:off  1:off  2:off  3:off  4:off
5:off  6:off
isdn          0:off  1:off  2:off  3:off  4:off  5:off
6:off
diskdump      0:off  1:off  2:off  3:off  4:off
5:off  6:off
rwhod         0:off  1:off  2:off  3:off  4:off  5:off
6:off
kprop         0:off  1:off  2:off  3:off  4:off  5:off
6:off
cyrus-imapd   0:off  1:off  2:off  3:off  4:off
5:off  6:off
amd           0:off  1:off  2:off  3:off  4:off  5:off
6:off
krb524        0:off  1:off  2:off  3:off  4:off  5:off
6:off
arptables_jf  0:off  1:off  2:off  3:off  4:off
5:off  6:off
microcode_ctl 0:off  1:off  2:off  3:off  4:off
5:off  6:off
psacct        0:off  1:off  2:off  3:off  4:off  5:off
6:off
ip6tables     0:off  1:off  2:off  3:off  4:off
5:off  6:off
cpuspeed      0:off  1:on   2:on   3:on   4:on
5:on   6:off
autofs        0:off  1:off  2:off  3:off  4:off  5:off
6:off
portmap       0:off  1:off  2:off  3:off  4:off
5:off  6:off
readahead     0:off  1:off  2:off  3:off  4:off
5:on   6:off
dc_client     0:off  1:off  2:off  3:off  4:off  5:off
6:off
cups          0:off  1:off  2:off  3:off  4:off  5:off
6:off
tog-pegasus   0:off  1:off  2:off  3:off  4:off
5:off  6:off
netfs         0:off  1:off  2:off  3:off  4:off  5:off
6:off
ripd          0:off  1:off  2:off  3:off  4:off  5:off
6:off
vsftpd        0:off  1:off  2:off  3:off  4:off  5:off
6:off
cups-config-daemon 0:off  1:off  2:off  3:off
4:off  5:off  6:off
dc_server     0:off  1:off  2:off  3:off  4:off
5:off  6:off
netplugd      0:off  1:off  2:off  3:off  4:off
5:off  6:off
anacron       0:off  1:off  2:off  3:off  4:off
5:off  6:off
bootparamd    0:off  1:off  2:off  3:off  4:off
5:off  6:off
mailman       0:off  1:off  2:off  3:off  4:off
5:off  6:off
sendmail      0:off  1:off  2:off  3:off  4:off
5:off  6:off
iscsi         0:off  1:off  2:off  3:off  4:off  5:off
6:off
```

```
rawdevices      0:off  1:off  2:off  3:on   4:on
5:on   6:off
network         0:off  1:off  2:on   3:on   4:on
5:on   6:off
smartd          0:off  1:off  2:off  3:off  4:off  5:off
6:off
sysstat         0:off  1:on   2:off  3:off  4:off  5:off
6:off
dhcp6s          0:off  1:off  2:off  3:off  4:off
5:off  6:off
auditd          0:off  1:off  2:off  3:off  4:off  5:off
6:off
rstatd          0:off  1:off  2:off  3:off  4:off  5:off
6:off
bluetooth       0:off  1:off  2:off  3:off  4:off
5:off  6:off
iiim            0:off  1:off  2:off  3:off  4:off  5:off
6:off
radiusd         0:off  1:off  2:off  3:off  4:off  5:off
6:off
oddjobd         0:off  1:off  2:off  3:off  4:off
5:off  6:off
ypserv          0:off  1:off  2:off  3:off  4:off  5:off
6:off
nfs             0:off  1:off  2:off  3:off  4:off  5:off
6:off
vncserver       0:off  1:off  2:off  3:off  4:off
5:off  6:off
mdmpd           0:off  1:off  2:off  3:off  4:off
5:off  6:off
rhnsd           0:off  1:off  2:off  3:off  4:off  5:off
6:off
named           0:off  1:off  2:off  3:off  4:off
5:off  6:off
FreeWnn         0:off  1:off  2:off  3:off  4:off
5:off  6:off
netdump-server  0:off  1:off  2:off  3:off  4:off
5:off  6:off
kudzu           0:off  1:off  2:off  3:off  4:off  5:off
6:off
ipmi            0:off  1:off  2:off  3:off  4:off  5:off
6:off
lisa            0:off  1:off  2:off  3:off  4:off  5:off
6:off
smb             0:off  1:off  2:off  3:off  4:off  5:off
6:off
tux             0:off  1:off  2:off  3:off  4:off  5:off
6:off
netdump         0:off  1:off  2:off  3:off  4:off
5:off  6:off
httpd           0:off  1:off  2:off  3:off  4:off  5:off
6:off
ospfd           0:off  1:off  2:off  3:off  4:off  5:off
6:off
dhcpd           0:off  1:off  2:off  3:off  4:off  5:off
6:off
radvd           0:off  1:off  2:off  3:off  4:off  5:off
6:off
innd            0:off  1:off  2:off  3:off  4:off  5:off
6:off
openibd         0:off  1:off  2:off  3:off  4:off
5:off  6:off
lm_sensors      0:off  1:off  2:off  3:off  4:off
5:off  6:off
rpcgssd         0:off  1:off  2:off  3:off  4:off
5:off  6:off
ripngd          0:off  1:off  2:off  3:off  4:off  5:off
6:off
nscd            0:off  1:off  2:off  3:off  4:off  5:off
6:off
syslog          0:off  1:off  2:on   3:on   4:on
5:on   6:off
xinetd based services:
    ktalk:  off
    chargen-udp:    off
    krb5-telnet:    off
    amandaidx:  off
    time:   off
    klogin: off
    rlogin: on
    cups-lpd:   off
    amidxtape:  off
    dbskkd-cdb: off
    talk:   off
    auth:   on
    time-udp:   off
    kshell: off
    chargen:    off
    daytime:    off
    amanda: off
    eklogin:    off
    echo:   off
    rsync:  off
    daytime-udp:    off
    gssftp: off
    swat:   off
    finger: off
    tftp:   off
    ntalk:  off
    telnet: on
    rsh:    on
    echo-udp:   off
    rexec:  off


::::::::::::::::::::::::::::::::::
 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)
#       - 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
#       - maxsyslogins - max number of logins on
the system
#       - priority - the priority to run user process
with
#       - locks - max number of file locks the user
can hold
#       - sigpending - max number of pending
signals
#       - msgqueue - max memory used by
POSIX message queues (bytes)
#
#<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 filediscriptor
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:$TUXDIR/lib

ulimit -u 30000
ulimit -s 1536

#/sbin/swapoff -a

# For 3tier tune
SVRAPL=`ps -e | grep tpccfmlw | 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 logfile 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         46
ServerLimit          46
ThreadLimit          500
MaxClients           23000
MinSpareThreads      1
MaxSpareThreads      23000
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 tpapl_module modules/mod_tpapl.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) - Luxembourgeois* (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
char set.
#
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
(SSI):
# (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="971801"
NumWarehouses="108480"
MaxUsers="1084800"
MaxTerm of Client="22600"
CONTROL_Flag="0"
LogPath="/home/tpc/log/userlog.log"

[SVRAPL_INFO]
LogPath="/home/tpc/log/DBDepend_Userlog.log
"




::::::::::::::::::::::::::::::::::
 tnsnames.ora
::::::::::::::::::::::::::::::::::

#
# 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_h)(Port= 1528))
    (SDU=14600)
    (CONNECT_DATA = (SERVICE_NAME =
tpcc))
  )


[TP monitor tunables]
----------------------------------------------------------


::::::::::::::::::::::::::::::::::
 ubbconfig
::::::::::::::::::::::::::::::::::

#
#   ubbconfig : 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
cl069   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

tpccfmlw    SRVGRP=group1 RQADDR=ware01
SRVID=1 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware02
SRVID=2 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware03
SRVID=3 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware04
SRVID=4 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware05
SRVID=5 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware06
SRVID=6 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware07
SRVID=7 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware08
SRVID=8 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
tpccfmlw    SRVGRP=group1 RQADDR=ware09
SRVID=9 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"
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"
tpccfmlw    SRVGRP=group1 RQADDR=ware19
SRVID=19 REPLYQ=N CLOPT="-s
OPSTUXSERVER:OPSTUXSERVER"

*SERVICES
"OPSTUXSERVER"  TRANTIME=0
SRVGRP=group1

*ROUTING
```

Appendix E: Database Creation Code

```
::::::::::::::::::::::::::::::::
createtable_cust.sql
::::::::::::::::::::::::::::::::

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 3360000000
  hash is ( (c_id * ( 112000 * 10 ) + c_w_id * 10 +
c_d_id) )
  size 200
  pctfree 0  initrans 3
  storage ( buffer_pool recycle ) parallel ( degree
32 )
  tablespace cust_0;

create table cust (
  c_id number
, c_d_id number
, c_w_id number
, c_discount number
, c_credit char(2)
, c_last varchar2(16)
, c_first varchar2(16)
, c_credit_lim number
, c_balance 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;


::::::::::::::::::::::::::::::::
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 205078M reuse,
          '/ora_dev/log_1_2' size 205078M 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


::::::::::::::::::::::::::::::::
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  initrans 3
  storage ( buffer_pool default )
  parallel 64
  compute statistics
  tablespace icust1_0 ;
    set echo off
    spool off
    exit sql.sqlcode;


::::::::::::::::::::::::::::::::
Makefile.linux
::::::::::::::::::::::::::::::::

#=================================
==============================+
#       Copyright (c) 1996  Oracle Corp,
Redwood Shores, CA       |
#                OPEN SYSTEMS PERFORMANCE
GROUP                |
#                   All Rights Reserved
|
#=================================
==============================+
# FILENAME
#      Makefile
# DESCRIPTION
#      Makefile for lib for batch driver, load
program and tx testing.
#=================================
==============================
#
#  Programs:
#
#      dpblibunix.o
#

all: compile dpblibunix.o

#include
$(ORACLE_HOME)/bench/buildtools/prefix.mk
I_SYM=-I
#include
$(ORACLE_HOME)/rdbms/lib/env_rdbms.mk
REMOVE=rm
#CC=/opt/SunProd/SUNWspro6.1/bin/../WS6U1/
bin/cc
CC=/usr/bin/gcc

TARGS=compile cleanup

TPCBIN=.
INCLUDE=$(I_SYM).
$(I_SYM)$(ORACLE_HOME)/rdbms/demo \
 $(I_SYM)$(ORACLE_HOME)/rdbms/public \
 $(I_SYM)$(ORACLE_HOME)/rdbms/include \
 $(I_SYM)$(ORACLE_HOME)/plsql/public \
 $(I_SYM)$(ORACLE_HOME)/network/public
ITUX=$(I_SYM)$(ROOTDIR)/include

MEMBS=
OBJS=gettime.o dpbproc.o dpbwait.o dpbpchk.o
dpbtimef.o

CFLAGS=

files:

compile: $(OBJS)
    @-$(DOTARGS)

cleanup:
    $(REMOVE) $(OBJS) dpblibunix.o

dpbtimef.o: dpbtimef.c
    $(CC) $(CFLAGS) -DORA_PC $(INCLUDE) -
c dpbtimef.c

dpbproc.o: dpbproc.c
    $(CC) $(CFLAGS) -DORA_AUX $(INCLUDE)
-c dpbproc.c

dpbwait.o: dpbwait.c
    $(CC) $(CFLAGS) -DORA_AUX $(INCLUDE)
-c dpbwait.c

dpbpchk.o: dpbpchk.c
    $(CC) $(CFLAGS) -DORA_AUX $(INCLUDE)
-c dpbpchk.c

gettime.o: gettime.c
    $(CC) $(CFLAGS) $(INCLUDE) -c gettime.c

trigger.o: trigger.c

dpblibunix.o: $(OBJS)
    $(LD) -r -o $@ $(OBJS)

c_trans_tux.o: $(CTRANTUX_OBJS)
    $(LD) -r -o $@ $(CTRANTUX_OBJS)


::::::::::::::::::::::::::::::::
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:
#
#     dpblibunix.o
#

all: compile dpblibunix.o

#include
$(ORACLE_HOME)/bench/buildtools/prefix.mk
I_SYM=-I
#include
$(ORACLE_HOME)/rdbms/lib/env_rdbms.mk
REMOVE=rm
#CC=/opt/SunProd/SUNWspro6.1/bin/../WS6U1/
bin/cc
CC=/usr/bin/gcc

TARGS=compile cleanup

TPCBIN=.
INCLUDE=$(I_SYM).
$(I_SYM)$(ORACLE_HOME)/rdbms/demo \
 $(I_SYM)$(ORACLE_HOME)/rdbms/public \
 $(I_SYM)$(ORACLE_HOME)/rdbms/include \
 $(I_SYM)$(ORACLE_HOME)/plsql/public \
 $(I_SYM)$(ORACLE_HOME)/network/public
ITUX=$(I_SYM)$(ROOTDIR)/include

MEMBS=
OBJS=gettime.o dpbproc.o dpbwait.o dpbpchk.o
dpbtimef.o

CFLAGS=

files:

compile: $(OBJS)
   @-$(DOTARGS)

cleanup:
   $(REMOVE) $(OBJS) dpblibunix.o

dpbtimef.o: dpbtimef.c
   $(CC) $(CFLAGS) -DORA_PC $(INCLUDE) -
c dpbtimef.c

dpbproc.o: dpbproc.c
   $(CC) $(CFLAGS) -DORA_AUX $(INCLUDE)
-c dpbproc.c

dpbwait.o: dpbwait.c
   $(CC) $(CFLAGS) -DORA_AUX $(INCLUDE)
-c dpbwait.c

dpbpchk.o: dpbpchk.c
   $(CC) $(CFLAGS) -DORA_AUX $(INCLUDE)
-c dpbpchk.c

gettime.o: gettime.c
   $(CC) $(CFLAGS) $(INCLUDE) -c gettime.c

trigger.o: trigger.c

dpblibunix.o: $(OBJS)
   $(LD) -r -o $@ $(OBJS)

c_trans_tux.o: $(CTRANTUX_OBJS)
   $(LD) -r -o $@ $(CTRANTUX_OBJS)


::::::::::::::::::::::::::::::::::
 lib/dpbcore.h
::::::::::::::::::::::::::::::::::

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

/*
NAME    DPBCORE.H

DESCRIPTION
  Header for CORE function

NOTES
  Desktop Performance Group

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

*/

#ifndef __dpbcore__
# define __dpbcore__


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

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


#endif /* __dpbcore__ */



::::::::::::::::::::::::::::::::::
 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 millaseconds as a
clock_t.

NOTES
  Desktop Performance Group

MODIFIED      (MM/DD/YY)
  B Moriarty   06/02/95 -  V4.8 Initial Version

*/

#ifdef ORA_OS2
#endif /* ORA_OS2 */

#ifdef ORA_NT
#include <windows.h>
#include <sys\types.h>
#include <time.h>
#include <stdio.h>

BOOL First = TRUE;
LARGE_INTEGER ICount;          /* Initial Time */
LARGE_INTEGER Tptms;       /* Ticks per tenth
of millisecond */
#endif /* ORA_NT */

# ifdef __STDC__
clock_t dpbetime(void)
# else
clock_t dpbetime()
# endif /* __STDC__ */
{

#ifdef ORA_NT

  LARGE_INTEGER PFreq;     /* Ticks per
Second */
  LARGE_INTEGER PCount;    /* Ticks Since
1970 */
  clock_t etime;      /* Elapsed time in tenths of
millaseconds */

  if (First)  {
    if (!QueryPerformanceFrequency(&PFreq))
      return((clock_t)-1);
    if (!QueryPerformanceCounter(&ICount))
      return((clock_t)-1);
    Tptms.QuadPart = PFreq.QuadPart / 10000;
    First = FALSE;
    return((clock_t)0);
  }
  if (!QueryPerformanceCounter(&PCount))
    return((clock_t)-1);
  etime = (clock_t) ((PCount.QuadPart -
ICount.QuadPart) / Tptms.QuadPart);
  return(etime);

#endif /* ORA_NT */

}
```

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.

  Calling fclose() or fflush() is not enough.
These calls will only flush
  the buffer in the FILE struture by making a
write() call to the o/s, and
  the o/s will probably place these data in its own
disk buffers.
  dpbfsync() will cause the o/s disk buffers for a
file to be written to disk.

  This function should normally be called *after*
an fflush() is done, or you
  will miss the data that is buffered in the FILE
structure.

NOTES
  Desktop Performance Group

MODIFIED      (MM/DD/YY)
  C Kelly     02/24/94 -  V4.4 initial version
*/


#include <stdio.h>



#ifdef ORA_OS2
int dpbfsync(FILE *fp)
{
 return 0;
}
#endif /* ORA_OS2 */



#ifdef ORA_NT
# include <windows.h>

int dpbfsync(FILE *fp)
{
 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 samtgid;

#else /* ORA_NW */
#endif /* ORA_NW */



#ifdef __STDC__
void dpbinpgm(void)
#else
void dpbinpgm()
#endif /* __STDC__ */
{
# ifdef ORA_NW

  samtid  = GetThreadID();     /* get this
program's thread id        */
  samtgid = GetThreadGroupID(); /* get this
program's thread group id   */
```

```
# else /* ORA_NW */

  return;               /* do nothing for everything else
*/

# endif /* ORA_NW */
}



::::::::::::::::::::::::::::::::
 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;
   }
   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    DPBPCNTL.H

DESCRIPTION
   OSD structures for process control

NOTES
   Desktop Performance Group
```

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

*/

#ifndef __dpbpcntl__
# define __dpbpcntl__



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



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



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



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



#endif /* __dpbpcntl__ */



::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::
 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 -  Iniital version

*/


#ifdef ORA_OS2
#include <string.h>
#include <sys\types.h>
#endif /* ORA_OS2 */


#ifdef ORA_NW
#endif /* ORA_NW */


#ifdef ORA_NT
# include <windows.h>
# include <string.h>
# define REALCLASS 'R'
# define HIGHCLASS 'H'
# define NORMALCLASS 'N'
# define IDLECLASS 'I'
#endif /* ORA_NT */


#ifdef ORA_AUX
#endif /* ORA_AUX */


#ifdef __STDC__
int dpbprty(char *prty_str)
#else
int dpbprty(prty_str)
char *prty_str;
#endif
{
#ifdef ORA_OS2
  return 0;
#endif /* ORA_OS2 */

#ifdef ORA_AUX
  return 0;
#endif /* ORA_AUX */

#ifdef ORA_NW
  return 0;
#endif /* ORA_NW */

#ifdef ORA_NT

  HANDLE this_process, this_thread;

  DWORD class;

 int prios;

  if ( (strlen(prty_str) > 2) || prty_str[0] == '0')
  {
    return(0);                  /* return if invalid length
or 0 */
  };

  this_process = GetCurrentProcess();

  switch (prty_str[0])
  {
  case IDLECLASS:
  case 'i':
    class = IDLE_PRIORITY_CLASS;
    break;

  case NORMALCLASS:
  case 'n':
    class = NORMAL_PRIORITY_CLASS;
    break;

  case HIGHCLASS:
  case 'h':
    class = HIGH_PRIORITY_CLASS;
    break;

  case REALCLASS:
  case 'r':
    class = REALTIME_PRIORITY_CLASS;
    break;
  };

  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:
  strcat(os_pri,"1");
  break;

case THREAD_PRIORITY_LOWEST:
  strcat(os_pri,"2");
  break;

case THREAD_PRIORITY_BELOW_NORMAL:
  strcat(os_pri,"3");
  break;

case THREAD_PRIORITY_NORMAL:
  strcat(os_pri,"4");
  break;

case THREAD_PRIORITY_ABOVE_NORMAL:
  strcat(os_pri,"5");
  break;

case THREAD_PRIORITY_HIGHEST:
  strcat(os_pri,"6");
  break;

case THREAD_PRIORITY_TIME_CRITICAL:
  strcat(os_pri,"7");
  break;

default:
  strcat(os_pri,"?");
  break;
};

 return 0;

#endif /* ORA_NT */
}



::::::::::::::::::::::::::::::::
 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 samtid;
extern int samtgid;

#else /* ORA_NW */
#endif /* ORA_NW */



#ifdef __STDC__
void dpbxtpgm(void)
#else
void dpbxtpgm()
#endif /* __STDC__ */
{
#ifdef ORA_NW

/*
** Cleanup code for NetWare.
**   This routine will cleanup any Oracle
connection should the module
**   be unexpectedly unloaded.
*/

  int oldtgid;

  oldtgid = SetThreadGroupID(samtgid);  /*
switch to this NLM's thread group */
  OraClientExit(samtid);                /* cleanup
Oracle connection    */
  SetThreadGroupID(oldtgid);            /* reset the
thread group      */

#else /* ORA_NW */

  return;               /* do nothing for everything else
*/

#endif /* ORA_NW */
}



::::::::::::::::::::::::::::::::
 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_PSX)
# 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_involcsw);
   fprintf (fp, "%lu ", ps->ps_signal);
   fprintf (fp, "%lu ", ps->ps_lread);
   fprintf (fp, "%lu ", ps->ps_lwrite);
   fprintf (fp, "%lu ", ps->ps_bread);
   fprintf (fp, "%lu ", ps->ps_bwrite);
   fprintf (fp, "%lu ", ps->ps_phread);
   fprintf (fp, "%lu",  ps->ps_phwrite);
}



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 <sys\types.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);
}



::::::::::::::::::::::::::::::::::
 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  initrans 3
  storage ( buffer_pool default )
  parallel 64
  compute statistics
  tablespace icust2_0 ;
    set echo off
    spool off
    exit sql.sqlcode;



::::::::::::::::::::::::::::::::::
 dpbcore.h
::::::::::::::::::::::::::::::::::

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

/*
NAME    DPBCORE.H

DESCRIPTION
   Header for CORE function

NOTES
   Desktop Performance Group

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

*/

#ifndef __dpbcore__
# define __dpbcore__


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

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

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


#endif /* __dpbcore__ */



::::::::::::::::::::::::::::::::::
 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  initrans 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  initrans 4
  storage ( buffer_pool default )

  compute statistics
  tablespace iitem_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_c_id
, o_d_id
, o_w_id
, o_id )
  pctfree 25  initrans 4
  storage ( buffer_pool default )
  parallel 64
  compute statistics
  tablespace iordr2_0 ;
    set echo off
    spool off
    exit sql.sqlcode;


::::::::::::::::::::::::::::::::::
 createindex_istok.sql
::::::::::::::::::::::::::::::::::

set timing on
    set sqlblanklines on
    spool createindex_istok.log ;
    set echo on ;
    drop index istok ;
      create unique index istok on stok ( s_i_id
, s_w_id )
  pctfree 1  initrans 3
  storage ( buffer_pool default )
  parallel 64
  compute statistics
  tablespace istok_0 ;
    set echo off
    spool off
    exit sql.sqlcode;


::::::::::::::::::::::::::::::::::
 createindex_iware.sql
::::::::::::::::::::::::::::::::::

/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreateindex.sh Sat May 31
04:42:14 JST 2008 */
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;


::::::::::::::::::::::::::::::::::
 createmisc.sh
::::::::::::::::::::::::::::::::::

#!/bin/sh

$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect

spool createmisc.log
set echo on;
alter user tpcc temporary tablespace system;
grant execute on dbms_lock to public;
grant execute on dbms_pipe to public;
grant select on v_\$parameter to public;

REM
REM begin plsql_mon.sql
REM

connect tpcc/tpcc;
set echo on;
CREATE OR REPLACE PACKAGE
plsql_mon_pack
IS
  PROCEDURE print
  (
    info        VARCHAR2
  );
END;
/
show errors;

CREATE OR REPLACE PACKAGE BODY
plsql_mon_pack
IS
  PROCEDURE print
  (
    info        VARCHAR2
  )
  IS
    s           NUMBER;
  BEGIN
    dbms_pipe.pack_message (info);
    s := dbms_pipe.send_message
('plsql_mon');
    IF (s <> 0) THEN
      raise_application_error (-20000, 'Error:' ||
to_char(s) ||
                        ' sending on pipe');
    END IF;
  END;
END;
/
show errors;
```

```
set echo off;

REM
REM end plsql_mon.sql
REM

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;

!









::::::::::::::::::::::::::::::::::
 createstoredprocs.sql
::::::::::::::::::::::::::::::::::

spool createstoreprocs.log
@tkvcinin.sql
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 1120000
  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;


::::::::::::::::::::::::::::::::::
 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 ;
    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 1120000
    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_ordl.sql
::::::::::::::::::::::::::::::::

set timing on
    set sqlblanklines on
    spool createtable_ordl.log
    set echo on
       create table ordl (
    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 ordl_uk primary key (ol_w_id,
ol_d_id, ol_o_id, ol_number  )) CLUSTER
ordrcluster_queue(ol_w_id, ol_d_id, ol_o_id,
ol_number) ;
    set echo off
    spool off
    exit sql.sqlcode;


::::::::::::::::::::::::::::::::
 createtable_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 1120000
    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
::::::::::::::::::::::::::::::::

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 11200000000
  hash is ( (s_i_id * 112000 + s_w_id) )
  size 256
  pctfree 0  initrans 2 maxtrans 2
  storage ( 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 112000
  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;


::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::
 createts.sh
::::::::::::::::::::::::::::::::::

#created automatically by /home/oracle/tpcc-
kit/scripts/buildcreatets.sh Sat May 31 04:41:46
JST 2008

# Tablespace ware, ts size 240M (245760K)
# each file 240M (245760K)
# extents 240704K (240704K)
# 1 files

$tpcc_createts ware 1 1       240M 240704K unix
0       0 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for ware failed.
Exiting.
      exit 0
    fi

# Tablespace cust, ts size 2989440M
(3061186560K)
# each file 5190M (5314560K)
# extents 663018K (663018K)
# 576 files

$tpcc_createts cust 576 1       5190M 663018K
unix 0       1 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for cust failed.
Exiting.
      exit 0
    fi

# Tablespace dist, ts size 2350M (2406400K)
# each file 2350M (2406400K)
# extents 1199424K (1199424K)
# 1 files

$tpcc_createts dist 1 1       2350M 1199424K
unix 0       577 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for dist failed.
Exiting.
      exit 0
    fi

# Tablespace hist, ts size 336000M
(344064000K)
# each file 3500M (3584000K)
# extents 102172K (102172K)
# 96 files

$tpcc_createts hist 96 1       3500M 102172K
unix 0       578 32 4K t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for hist failed.
Exiting.
      exit 0
    fi

# Tablespace stok, ts size 3350400M
(3430809600K)
# each file 6980M (7147520K)
# extents 714358K (714358K)
# 480 files

$tpcc_createts stok 480 1       6980M 714358K
unix 0       674 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for stok 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
1154 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for item failed.
Exiting.
      exit 0
    fi

# Tablespace ordr, ts size 4785600M
(4900454400K)
# each file 49850M (51046400K)
# extents 103328K (103328K)
# 96 files

$tpcc_createts ordr 96 1       49850M 103328K
unix 0       1155 32 16K t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for ordr failed.
Exiting.
      exit 0
    fi

# Tablespace nord, ts size 40320M (41287680K)
# each file 420M (430080K)
# extents 42636K (42636K)
# 96 files

$tpcc_createts nord 96 1       420M 42636K unix
0       1251 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for nord failed.
Exiting.
      exit 0
    fi

# Tablespace iware, ts size 140M (143360K)
# each file 140M (143360K)
# extents 141024K (141024K)
# 1 files

$tpcc_createts iware 1 1       140M 141024K
unix 0       1347 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for iware failed.
Exiting.
      exit 0
    fi

# Tablespace icust1, ts size 91680M
(93880320K)
# each file 1910M (1955840K)
# extents 7600K (7600K)
# 48 files

$tpcc_createts icust1 48 1       1910M 7600K
unix 0       1348 32 16K t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for icust1 failed.
Exiting.
      exit 0
    fi

# Tablespace icust2, ts size 196800M
(201523200K)
# each file 2050M (2099200K)
# extents 8192K (8192K)
# 96 files

$tpcc_createts icust2 96 1       2050M 8192K
unix 0       1396 32 16K t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for icust2 failed.
Exiting.
      exit 0
    fi

# Tablespace idist, ts size 550M (563200K)
# each file 550M (563200K)
# extents 561024K (561024K)
# 1 files

$tpcc_createts idist 1 1       550M 561024K unix
0       1492 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for idist failed.
Exiting.
      exit 0
    fi

# Tablespace istok, ts size 244000M
(249856000K)
# each file 6100M (6246400K)
# extents 24368K (24368K)
# 40 files

$tpcc_createts istok 40 1       6100M 24368K
unix 0       1493 32 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
1533 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for iitem failed.
Exiting.
      exit 0
    fi

# Tablespace iordr2, ts size 214080M
(219217920K)
# each file 2230M (2283520K)
# extents 8912K (8912K)
# 96 files

$tpcc_createts iordr2 96 1       2230M 8912K
unix 0       1534 32 16K t
    if expr $? != 0 > /dev/null; then
```

```
    echo Creating tablespace for iordr2 failed.
Exiting.
    exit 0
  fi

# Tablespace temp, ts size 586560M
(600637440K)
# each file 6110M (6256640K)
# extents 201740K (201740K)
# 96 files

$tpcc_createts temp 96 1      6110M 201740K
unix 1      1630 32 auto t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for temp failed.
Exiting.
    exit 0
  fi


::::::::::::::::::::::::::::::::
 dpbpcntl.h
::::::::::::::::::::::::::::::::

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

/*
NAME    DPBPCNTL.H

DESCRIPTION
  OSD structures for process control

NOTES
  Desktop Performance Group

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

*/

#ifndef __dpbpcntl__
# define __dpbpcntl__



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



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



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



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



#endif /* __dpbpcntl__ */



::::::::::::::::::::::::::::::::
 loadcust.sh
::::::::::::::::::::::::::::::::

#created automatically by /home/oracle/tpcc-
kit/scripts/evenload.sh Sat May 31 04:42:12 JST
2008
rm -f loadcust*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 112000 -C  -l 1 -m 46 >>
loadcust0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 47 -m 92 >>
loadcust1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 93 -m 138 >>
loadcust2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 139 -m 184 >>
loadcust3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 185 -m 230 >>
loadcust4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 231 -m 276 >>
loadcust5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 277 -m 322 >>
loadcust6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 323 -m 368 >>
loadcust7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 369 -m 415 >>
loadcust8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 416 -m 462 >>
loadcust9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 463 -m 509 >>
loadcust10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 510 -m 556 >>
loadcust11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 557 -m 603 >>
loadcust12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 604 -m 650 >>
loadcust13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 651 -m 697 >>
loadcust14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 698 -m 744 >>
loadcust15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 745 -m 791 >>
loadcust16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 792 -m 838 >>
loadcust17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 839 -m 885 >>
loadcust18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 886 -m 932 >>
loadcust19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 933 -m 979 >>
loadcust20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 980 -m 1026 >>
loadcust21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1027 -m 1073 >>
loadcust22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1074 -m 1120 >>
loadcust23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1121 -m 1167 >>
loadcust24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1168 -m 1214 >>
loadcust25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1215 -m 1261 >>
loadcust26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1262 -m 1308 >>
loadcust27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1309 -m 1355 >>
loadcust28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1356 -m 1402 >>
loadcust29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1403 -m 1449 >>
loadcust30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1450 -m 1496 >>
loadcust31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1497 -m 1543 >>
loadcust32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1544 -m 1590 >>
loadcust33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1591 -m 1637 >>
loadcust34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1638 -m 1684 >>
loadcust35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1685 -m 1731 >>
loadcust36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1732 -m 1778 >>
loadcust37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1779 -m 1825 >>
loadcust38.log 2>&1 &
allprocs="$allprocs ${!}"
```

```
$tpcc_load -M 112000 -C  -l 1826 -m 1872 >>
loadcust39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1873 -m 1919 >>
loadcust40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1920 -m 1966 >>
loadcust41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 1967 -m 2013 >>
loadcust42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2014 -m 2060 >>
loadcust43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2061 -m 2107 >>
loadcust44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2108 -m 2154 >>
loadcust45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2155 -m 2201 >>
loadcust46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2202 -m 2248 >>
loadcust47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2249 -m 2295 >>
loadcust48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2296 -m 2342 >>
loadcust49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2343 -m 2389 >>
loadcust50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2390 -m 2436 >>
loadcust51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2437 -m 2483 >>
loadcust52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2484 -m 2530 >>
loadcust53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2531 -m 2577 >>
loadcust54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2578 -m 2624 >>
loadcust55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2625 -m 2671 >>
loadcust56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2672 -m 2718 >>
loadcust57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2719 -m 2765 >>
loadcust58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2766 -m 2812 >>
loadcust59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2813 -m 2859 >>
loadcust60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2860 -m 2906 >>
loadcust61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2907 -m 2953 >>
loadcust62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -C  -l 2954 -m 3000 >>
loadcust63.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
  wait $curproc
  error=`expr $? + $error`
done
exit `expr $error != 0`


::::::::::::::::::::::::::::::::
 loaddist.sh
::::::::::::::::::::::::::::::::

cd $tpcc_bench
$tpcc_load -M $tpcc_scale -d > loaddist.log
2>&1


::::::::::::::::::::::::::::::::
 loadhist.sh
::::::::::::::::::::::::::::::::

#created automatically by /home/oracle/tpcc-
kit/scripts/evenload.sh Sat May 31 04:42:10 JST
2008
exit
rm -f loadhist*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 112000 -h  -b 1 -e 1750 >>
loadhist0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 1751 -e 3500 >>
loadhist1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 3501 -e 5250 >>
loadhist2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 5251 -e 7000 >>
loadhist3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 7001 -e 8750 >>
loadhist4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 8751 -e 10500 >>
loadhist5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 10501 -e 12250 >>
loadhist6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 12251 -e 14000 >>
loadhist7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 14001 -e 15750 >>
loadhist8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 15751 -e 17500 >>
loadhist9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 17501 -e 19250 >>
loadhist10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 19251 -e 21000 >>
loadhist11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 21001 -e 22750 >>
loadhist12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 22751 -e 24500 >>
loadhist13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 24501 -e 26250 >>
loadhist14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 26251 -e 28000 >>
loadhist15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 28001 -e 29750 >>
loadhist16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 29751 -e 31500 >>
loadhist17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 31501 -e 33250 >>
loadhist18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 33251 -e 35000 >>
loadhist19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 35001 -e 36750 >>
loadhist20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 36751 -e 38500 >>
loadhist21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 38501 -e 40250 >>
loadhist22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 40251 -e 42000 >>
loadhist23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 42001 -e 43750 >>
loadhist24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 43751 -e 45500 >>
loadhist25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 45501 -e 47250 >>
loadhist26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 47251 -e 49000 >>
loadhist27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 49001 -e 50750 >>
loadhist28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 50751 -e 52500 >>
loadhist29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 52501 -e 54250 >>
loadhist30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 54251 -e 56000 >>
loadhist31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 56001 -e 57750 >>
loadhist32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 57751 -e 59500 >>
loadhist33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 59501 -e 61250 >>
loadhist34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 61251 -e 63000 >>
loadhist35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 63001 -e 64750 >>
loadhist36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 64751 -e 66500 >>
loadhist37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 66501 -e 68250 >>
loadhist38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 68251 -e 70000 >>
loadhist39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 70001 -e 71750 >>
loadhist40.log 2>&1 &
```

```
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 71751 -e 73500 >>
loadhist41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 73501 -e 75250 >>
loadhist42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 75251 -e 77000 >>
loadhist43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 77001 -e 78750 >>
loadhist44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 78751 -e 80500 >>
loadhist45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 80501 -e 82250 >>
loadhist46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 82251 -e 84000 >>
loadhist47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 84001 -e 85750 >>
loadhist48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 85751 -e 87500 >>
loadhist49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 87501 -e 89250 >>
loadhist50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 89251 -e 91000 >>
loadhist51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 91001 -e 92750 >>
loadhist52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 92751 -e 94500 >>
loadhist53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 94501 -e 96250 >>
loadhist54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 96251 -e 98000 >>
loadhist55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 98001 -e 99750 >>
loadhist56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 99751 -e 101500
>> loadhist57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 101501 -e 103250
>> loadhist58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 103251 -e 105000
>> loadhist59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 105001 -e 106750
>> loadhist60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 106751 -e 108500
>> loadhist61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 108501 -e 110250
>> loadhist62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -h  -b 110251 -e 112000
>> loadhist63.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
  wait $curproc
  error=`expr $? + $error`
done
exit `expr $error != 0`


::::::::::::::::::::::::::::::::::
 loaditem.sh
::::::::::::::::::::::::::::::::::

cd $tpcc_bench
$tpcc_load -M $tpcc_scale -i > loaditem.log
2>&1


::::::::::::::::::::::::::::::::::
 loadnord.sh
::::::::::::::::::::::::::::::::::

#created automatically by /home/oracle/tpcc-
kit/scripts/evenload.sh Sat May 31 04:42:10 JST
2008
rm -f loadnord*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 112000 -n  -b 1 -e 1750 >>
loadnord0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 1751 -e 3500 >>
loadnord1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 3501 -e 5250 >>
loadnord2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 5251 -e 7000 >>
loadnord3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 7001 -e 8750 >>
loadnord4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 8751 -e 10500 >>
loadnord5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 10501 -e 12250 >>
loadnord6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 12251 -e 14000 >>
loadnord7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 14001 -e 15750 >>
loadnord8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 15751 -e 17500 >>
loadnord9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 17501 -e 19250 >>
loadnord10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 19251 -e 21000 >>
loadnord11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 21001 -e 22750 >>
loadnord12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 22751 -e 24500 >>
loadnord13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 24501 -e 26250 >>
loadnord14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 26251 -e 28000 >>
loadnord15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 28001 -e 29750 >>
loadnord16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 29751 -e 31500 >>
loadnord17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 31501 -e 33250 >>
loadnord18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 33251 -e 35000 >>
loadnord19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 35001 -e 36750 >>
loadnord20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 36751 -e 38500 >>
loadnord21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 38501 -e 40250 >>
loadnord22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 40251 -e 42000 >>
loadnord23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 42001 -e 43750 >>
loadnord24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 43751 -e 45500 >>
loadnord25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 45501 -e 47250 >>
loadnord26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 47251 -e 49000 >>
loadnord27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 49001 -e 50750 >>
loadnord28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 50751 -e 52500 >>
loadnord29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 52501 -e 54250 >>
loadnord30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 54251 -e 56000 >>
loadnord31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 56001 -e 57750 >>
loadnord32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 57751 -e 59500 >>
loadnord33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 59501 -e 61250 >>
loadnord34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 61251 -e 63000 >>
loadnord35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 63001 -e 64750 >>
loadnord36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 64751 -e 66500 >>
loadnord37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 66501 -e 68250 >>
loadnord38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 68251 -e 70000 >>
loadnord39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 70001 -e 71750 >>
loadnord40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 71751 -e 73500 >>
loadnord41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 73501 -e 75250 >>
loadnord42.log 2>&1 &
```

```
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 75251 -e 77000 >>
loadnord43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 77001 -e 78750 >>
loadnord44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 78751 -e 80500 >>
loadnord45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 80501 -e 82250 >>
loadnord46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 82251 -e 84000 >>
loadnord47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 84001 -e 85750 >>
loadnord48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 85751 -e 87500 >>
loadnord49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 87501 -e 89250 >>
loadnord50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 89251 -e 91000 >>
loadnord51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 91001 -e 92750 >>
loadnord52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 92751 -e 94500 >>
loadnord53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 94501 -e 96250 >>
loadnord54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 96251 -e 98000 >>
loadnord55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 98001 -e 99750 >>
loadnord56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 99751 -e 101500
>> loadnord57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 101501 -e 103250
>> loadnord58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 103251 -e 105000
>> loadnord59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 105001 -e 106750
>> loadnord60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 106751 -e 108500
>> loadnord61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 108501 -e 110250
>> loadnord62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -n  -b 110251 -e 112000
>> loadnord63.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
  wait $curproc
  error=`expr $? + $error`
done
exit `expr $error != 0`
```

```
::::::::::::::::::::::::::::::::::
 loadordrordl.sh
::::::::::::::::::::::::::::::::::
#created automatically by /home/oracle/tpcc-
kit/scripts/evenload.sh Sat May 31 04:42:11 JST
2008
rm -f loadordrordl*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy0.dat -b 1 -e 1750
>> loadordrordl0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy1.dat -b 1751 -e
3500 >> loadordrordl1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy2.dat -b 3501 -e
5250 >> loadordrordl2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy3.dat -b 5251 -e
7000 >> loadordrordl3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy4.dat -b 7001 -e
8750 >> loadordrordl4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy5.dat -b 8751 -e
10500 >> loadordrordl5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy6.dat -b 10501 -e
12250 >> loadordrordl6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy7.dat -b 12251 -e
14000 >> loadordrordl7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy8.dat -b 14001 -e
15750 >> loadordrordl8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy9.dat -b 15751 -e
17500 >> loadordrordl9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy10.dat -b 17501 -e
19250 >> loadordrordl10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy11.dat -b 19251 -e
21000 >> loadordrordl11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy12.dat -b 21001 -e
22750 >> loadordrordl12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy13.dat -b 22751 -e
24500 >> loadordrordl13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy14.dat -b 24501 -e
26250 >> loadordrordl14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy15.dat -b 26251 -e
28000 >> loadordrordl15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy16.dat -b 28001 -e
29750 >> loadordrordl16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy17.dat -b 29751 -e
31500 >> loadordrordl17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy18.dat -b 31501 -e
33250 >> loadordrordl18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy19.dat -b 33251 -e
35000 >> loadordrordl19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy20.dat -b 35001 -e
36750 >> loadordrordl20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy21.dat -b 36751 -e
38500 >> loadordrordl21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy22.dat -b 38501 -e
40250 >> loadordrordl22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy23.dat -b 40251 -e
42000 >> loadordrordl23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy24.dat -b 42001 -e
43750 >> loadordrordl24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy25.dat -b 43751 -e
45500 >> loadordrordl25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy26.dat -b 45501 -e
47250 >> loadordrordl26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy27.dat -b 47251 -e
49000 >> loadordrordl27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy28.dat -b 49001 -e
50750 >> loadordrordl28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy29.dat -b 50751 -e
52500 >> loadordrordl29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy30.dat -b 52501 -e
54250 >> loadordrordl30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy31.dat -b 54251 -e
56000 >> loadordrordl31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy32.dat -b 56001 -e
57750 >> loadordrordl32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy33.dat -b 57751 -e
59500 >> loadordrordl33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy34.dat -b 59501 -e
61250 >> loadordrordl34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy35.dat -b 61251 -e
63000 >> loadordrordl35.log 2>&1 &
```

```
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy36.dat -b 63001 -e
64750 >> loadordrordl36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy37.dat -b 64751 -e
66500 >> loadordrordl37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy38.dat -b 66501 -e
68250 >> loadordrordl38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy39.dat -b 68251 -e
70000 >> loadordrordl39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy40.dat -b 70001 -e
71750 >> loadordrordl40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy41.dat -b 71751 -e
73500 >> loadordrordl41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy42.dat -b 73501 -e
75250 >> loadordrordl42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy43.dat -b 75251 -e
77000 >> loadordrordl43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy44.dat -b 77001 -e
78750 >> loadordrordl44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy45.dat -b 78751 -e
80500 >> loadordrordl45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy46.dat -b 80501 -e
82250 >> loadordrordl46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy47.dat -b 82251 -e
84000 >> loadordrordl47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy48.dat -b 84001 -e
85750 >> loadordrordl48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy49.dat -b 85751 -e
87500 >> loadordrordl49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy50.dat -b 87501 -e
89250 >> loadordrordl50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy51.dat -b 89251 -e
91000 >> loadordrordl51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy52.dat -b 91001 -e
92750 >> loadordrordl52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy53.dat -b 92751 -e
94500 >> loadordrordl53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy54.dat -b 94501 -e
96250 >> loadordrordl54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy55.dat -b 96251 -e
98000 >> loadordrordl55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy56.dat -b 98001 -e
99750 >> loadordrordl56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy57.dat -b 99751 -e
101500 >> loadordrordl57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy58.dat -b 101501 -
e 103250 >> loadordrordl58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy59.dat -b 103251 -
e 105000 >> loadordrordl59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy60.dat -b 105001 -
e 106750 >> loadordrordl60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy61.dat -b 106751 -
e 108500 >> loadordrordl61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy62.dat -b 108501 -
e 110250 >> loadordrordl62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -o
${tpcc_disks_location}dummy63.dat -b 110251 -
e 112000 >> loadordrordl63.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
  wait $curproc
  error=`expr $? + $error`
done
exit `expr $error != 0`


::::::::::::::::::::::::::::::::::
 loadstok.sh
::::::::::::::::::::::::::::::::::

#created automatically by /home/oracle/tpcc-
kit/scripts/evenload.sh Sat May 31 04:42:13 JST
2008
rm -f loadstok*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 112000 -S  -j 1 -k 1562 >>
loadstok0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 1563 -k 3124 >>
loadstok1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 3125 -k 4686 >>
loadstok2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 4687 -k 6248 >>
loadstok3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 6249 -k 7810 >>
loadstok4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 7811 -k 9372 >>
loadstok5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 9373 -k 10934 >>
loadstok6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 10935 -k 12496 >>
loadstok7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 12497 -k 14058 >>
loadstok8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 14059 -k 15620 >>
loadstok9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 15621 -k 17182 >>
loadstok10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 17183 -k 18744 >>
loadstok11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 18745 -k 20306 >>
loadstok12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 20307 -k 21868 >>
loadstok13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 21869 -k 23430 >>
loadstok14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 23431 -k 24992 >>
loadstok15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 24993 -k 26554 >>
loadstok16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 26555 -k 28116 >>
loadstok17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 28117 -k 29678 >>
loadstok18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 29679 -k 31240 >>
loadstok19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 31241 -k 32802 >>
loadstok20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 32803 -k 34364 >>
loadstok21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 34365 -k 35926 >>
loadstok22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 35927 -k 37488 >>
loadstok23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 37489 -k 39050 >>
loadstok24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 39051 -k 40612 >>
loadstok25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 40613 -k 42174 >>
loadstok26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 42175 -k 43736 >>
loadstok27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 43737 -k 45298 >>
loadstok28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 45299 -k 46860 >>
loadstok29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 46861 -k 48422 >>
loadstok30.log 2>&1 &
```

```
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 48423 -k 49984 >>
loadstok31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 49985 -k 51547 >>
loadstok32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 51548 -k 53110 >>
loadstok33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 53111 -k 54673 >>
loadstok34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 54674 -k 56236 >>
loadstok35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 56237 -k 57799 >>
loadstok36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 57800 -k 59362 >>
loadstok37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 59363 -k 60925 >>
loadstok38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 60926 -k 62488 >>
loadstok39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 62489 -k 64051 >>
loadstok40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 64052 -k 65614 >>
loadstok41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 65615 -k 67177 >>
loadstok42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 67178 -k 68740 >>
loadstok43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 68741 -k 70303 >>
loadstok44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 70304 -k 71866 >>
loadstok45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 71867 -k 73429 >>
loadstok46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 73430 -k 74992 >>
loadstok47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 74993 -k 76555 >>
loadstok48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 76556 -k 78118 >>
loadstok49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 78119 -k 79681 >>
loadstok50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 79682 -k 81244 >>
loadstok51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 81245 -k 82807 >>
loadstok52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 82808 -k 84370 >>
loadstok53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 84371 -k 85933 >>
loadstok54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 85934 -k 87496 >>
loadstok55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 87497 -k 89059 >>
loadstok56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 89060 -k 90622 >>
loadstok57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 90623 -k 92185 >>
loadstok58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 92186 -k 93748 >>
loadstok59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 93749 -k 95311 >>
loadstok60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 95312 -k 96874 >>
loadstok61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 96875 -k 98437 >>
loadstok62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 112000 -S  -j 98438 -k 100000 >>
loadstok63.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
  wait $curproc
  error=`expr $? + $error`
done
exit `expr $error != 0`


::::::::::::::::::::::::::::::::::
 loadware.sh
::::::::::::::::::::::::::::::::::

cd $tpcc_bench
$tpcc_load -M $tpcc_scale -w > loadware.log
2>&1


::::::::::::::::::::::::::::::::::
 space_get.sql
::::::::::::::::::::::::::::::::::

REM================================
================================+
REM        Copyright (c) 1995  Oracle Corp,
Redwood Shores, CA      |
REM                 OPEN SYSTEMS
PERFORMANCE GROUP               |
REM                      All Rights Reserved
|
REM================================
================================+
REM FILENAME
REM       space_get.sql
REM DESCRIPTION
REM       Get sizes of tables, indexes and
tablespaces.
REM  Usage: sqlplus 'sys/change_on_install as
sysdba' @space_get [<tpm> <# of
warehouses>]
REM================================
================================*/

   set echo on;
   delete from tpcc_data;
   delete from tpcc_space;
   delete from tpcc_totspace;

   insert into tpcc_data
   select substr(segment_name,1,18),
substr(segment_type,1,15),
        sum(blocks), t.block_size,
        round(sum(blocks) * 0.05), 0,
        sum(blocks) + round(sum(blocks) * 0.05)
   from   dba_extents e, dba_tablespaces t
   where  owner = 'TPCC' AND ( segment_type =
'INDEX' OR
        segment_type = 'INDEX PARTITION' OR
segment_type = 'CLUSTER'
        OR segment_type = 'TABLE' OR
segment_type = 'TABLE PARTITION')
        AND e.tablespace_name <> 'SYSTEM'
AND e.tablespace_name <> 'SP_0'
        AND e.tablespace_name =
t.tablespace_name
   group by segment_name, segment_type,
t.block_size;

   insert into tpcc_data
      select 'SYSTEM', 'SYS', sum(blocks),
t.block_size, 0, 0, sum(blocks)
      from   dba_data_files f, dba_tablespaces t
      where  f.tablespace_name = 'SYSTEM' and
t.tablespace_name = f.tablespace_name
      group by t.block_size;

   insert into tpcc_data
      select 'SYSAUX', 'SYS', sum(blocks),
t.block_size, 0, 0, sum(blocks)
      from   dba_data_files f, dba_tablespaces t
      where  f.tablespace_name = 'SYSAUX' and
t.tablespace_name = f.tablespace_name
      group by t.block_size;

   insert into tpcc_data
      select 'ROLL_SEG', 'SYS', sum(blocks),
t.block_size, 0, 0, sum(blocks)
      from   dba_data_files f, dba_tablespaces t
      where  f.tablespace_name like
'%UNDO_TS%' and f.tablespace_name =
t.tablespace_name
      group by f.tablespace_name, t.block_size;

   insert into tpcc_data
      select 'DB_STAT', 'SYS', sum(blocks),
t.block_size, 0, 0, sum(blocks)
      from   dba_data_files f, dba_tablespaces t
      where  f.tablespace_name like '%SP_0%'
and f.tablespace_name = t.tablespace_name
      group by f.tablespace_name, t.block_size;

   update tpcc_data
      set five_pct = 0,
         daily_grow = round(blocks * &&1 / 62.5 /
&&2),
         total = blocks + round(blocks * &&1 / 62.5 /
&&2)
      where segment = 'HIST' OR segment =
'ORDRCLUSTER_QUEUE' OR
           segment = 'IORDL';

   insert into tpcc_space
   select substr(ex$.name,1,18),
sum(sp$.sz_blocks), sp$.block_size, 0, 0, 0, 0
   from
     (select f.tablespace_name , sum(blocks)
sz_blocks, t.block_size block_size
      from dba_data_files f, dba_tablespaces t
      where f.tablespace_name <> 'SYSTEM' and
f.tablespace_name = t.tablespace_name
      group by f.tablespace_name, t.block_size
     ) sp$,
     (select distinct tablespace_name,
segment_name name
      from dba_extents
      where owner = 'TPCC'
```

```
      and (segment_type = 'CLUSTER' or
segment_type = 'TABLE'
       or segment_type = 'TABLE PARTITION' or
segment_type = 'INDEX'
       or segment_type = 'INDEX PARTITION')
      and tablespace_name <> 'SYSTEM'
    ) ex$
  where sp$.tablespace_name =
ex$.tablespace_name
  group by ex$.name, sp$.block_size;

  insert into tpcc_space
  select substr(f.tablespace_name,1,18),
sum(blocks), t.block_size, 0, 0, 0, 0
  from dba_data_files f, dba_tablespaces t
  where (f.tablespace_name = 'SYSTEM' or
f.tablespace_name = 'SYSAUX')
      and f.tablespace_name =
t.tablespace_name
  group by f.tablespace_name, t.block_size;

  insert into tpcc_space
  select 'ROLL_SEG', sum(blocks), t.block_size,
0, 0, 0, 0
  from dba_data_files f, dba_tablespaces t
  where f.tablespace_name = 'UNDO_TS' and
f.tablespace_name = t.tablespace_name
  group by f.tablespace_name, t.block_size;

  insert into tpcc_space
  select 'DB_STAT', sum(blocks), t.block_size, 0,
0, 0, 0
  from dba_data_files f, dba_tablespaces t
  where f.tablespace_name = 'SP_0' and
f.tablespace_name = t.tablespace_name
  group by f.tablespace_name, t.block_size;

  update tpcc_space
    set required =
    (
      select sum(total)
      from   tpcc_data
      where  tpcc_data.segment =
tpcc_space.segment
    )
    where segment in
    (
      select segment from tpcc_data
    );

  update tpcc_space
    set static =
    (
      select sum(total)
      from   tpcc_data
      where  tpcc_data.segment =
tpcc_space.segment
    )
    where segment in
    (
      select segment from tpcc_data
    );

  update tpcc_space
    set static = 0,
       dynamic =
    (
      select sum(blocks)
      from   tpcc_data
      where  tpcc_data.segment =
tpcc_space.segment
    )
    where segment in ('HIST',
'ORDRCLUSTER_QUEUE', 'IORDL');

  update tpcc_space
    set oversize = blocks  - required;


  insert into tpcc_totspace
    select &&1, &&2, sum(static *
block_size)/1024, sum(dynamic *
block_size)/1024, sum(oversize *
block_size)/1024, 0, 0, 0
    from   tpcc_space;

  update tpcc_totspace
    set daily_grow =
    (
      select sum(daily_grow * block_size)/1024
      from   tpcc_data
    );
  update tpcc_totspace
    set space60 = static + 60 * daily_grow;
  set echo off;



::::::::::::::::::::::::::::::::::
 space_init.sql
::::::::::::::::::::::::::::::::::

REM================================
================================+
REM FILENAME
REM       space_init.sql
REM DESCRIPTION
REM       Create tables for space calculations.
REM  Usage: sqlplus 'sys/change_on_install as
sysdba' @space_init.sql
REM================================
================================*/
  set echo on;
  drop table tpcc_data;
  drop table tpcc_space;
  drop table tpcc_totspace;
  create table tpcc_data (
    segment      varchar2(18),
    type         varchar2(15),
    blocks       number,
    block_size   number,
    five_pct     number,
    daily_grow   number,
    total        number
  );
  create table tpcc_space (
    segment      varchar2(18),
    blocks       number,
    block_size   number,
    required     number,
    static       number,
    dynamic      number,
    oversize     number
  );
  create table tpcc_totspace (
    tpm          number,
    nware        number,
    static       number,
    dynamic      number,
    oversize     number,
    daily_grow   number,
    daily_spre   number,
    space60      number
  );
  create unique index itpcc_data on tpcc_data
(segment);
  create unique index itpcc_space on
tpcc_space (segment);
  set echo off;



::::::::::::::::::::::::::::::::::
 space_rpt.sql
::::::::::::::::::::::::::::::::::

REM================================
================================+
REM        Copyright (c) 1995  Oracle Corp,
Redwood Shores, CA      |
REM                  OPEN SYSTEMS
PERFORMANCE GROUP               |
REM                       All Rights Reserved
|
REM================================
================================+
REM FILENAME
REM       space_rpt.sql
REM DESCRIPTION
REM       Generate space report and save it in
space.rpt
REM  Usage: sqlplus 'sys/change_on_install as
sysdba' @space_rpt.sql
REM================================
================================*/
  set space 2
  set pagesize 2000
  set echo off
  set termout off
  set verify off
  set feedback off
  set pagesize 60 linesize 120
  spool space.rpt
  select tpm, nware from tpcc_totspace;
  select * from tpcc_data order by segment;
  select * from tpcc_space order by segment;
  select static, dynamic, oversize, daily_grow,
daily_spre, space60
    from tpcc_totspace;
  spool off;



::::::::::::::::::::::::::::::::::
 tkvcinin.sql
::::::::::::::::::::::::::::::::::

-- The initnew package for storing variables used
in the
-- New Order anonymous block

CREATE OR REPLACE PACKAGE inittpcc
AS
 TYPE intarray IS TABLE OF INTEGER INDEX
BY BINARY_INTEGER;
 TYPE distarray IS TABLE OF VARCHAR(24)
INDEX BY BINARY_INTEGER;
 nulldate      DATE;
 TYPE rowidarray IS TABLE OF ROWID INDEX
BY PLS_INTEGER;
 s_dist        distarray;
 idx1arr       intarray;
 s_remote      intarray;
 dist          intarray;
 row_id        rowidarray;
 cust_rowid    rowid;
 dist_name     VARCHAR2(11);
 ware_name     VARCHAR2(11);
 c_num         PLS_INTEGER;

 PROCEDURE init_no(idxarr intarray);
 PROCEDURE init_del;
 PROCEDURE init_pay;
END inittpcc;
/
```

```
show errors;

CREATE OR REPLACE PACKAGE BODY
inittpcc AS
  PROCEDURE init_no (idxarr  intarray)
  IS
  BEGIN
       -- initialize null date
   nulldate := TO_DATE('01-01-1811', 'MM-DD-
YYYY');
    idx1arr := idxarr;
  END init_no;

  PROCEDURE init_del
  IS
  BEGIN
    FOR i IN 1 .. 10 LOOP
      dist(i) := i;
    END LOOP;
  END init_del;

  PROCEDURE init_pay IS
  BEGIN
    NULL;
  END init_pay;

END inittpcc;
/
show errors
exit


::::::::::::::::::::::::::::::::
 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 ;
!


::::::::::::::::::::::::::::::::
 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



::::::::::::::::::::::::::::::::
 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(OWNNA
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(OWNNA
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(OWNNA
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(OWNNA
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;


::::::::::::::::::::::::::::::::
 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


::::::::::::::::::::::::::::::::
 assigntemp.sql
::::::::::::::::::::::::::::::::

spool assigntemp.log;

set echo on;

alter user tpcc temporary tablespace temp_0;

set echo off;
spool off;

exit ;


::::::::::::::::::::::::::::::::
 bcexpr.sh
::::::::::::::::::::::::::::::::

#!/bin/sh
# send command line to bc
echo "$*" | bc


::::::::::::::::::::::::::::::::
 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
perfmon
REM

@$tpcc_sql_dir/cs_tpcc
@$tpcc_sql_dir/cs_cpu
@$tpcc_sql_dir/cs_os
@$tpcc_sql_dir/cs_proc
@$tpcc_sql_dir/cs_thread

REM
REM tpcc result table for unix and NT
REM

@$tpcc_sql_dir/${cstat}
@$tpcc_sql_dir/pst_c

!


::::::::::::::::::::::::::::::::
 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


::::::::::::::::::::::::::::::::
 createuser.sh
::::::::::::::::::::::::::::::::

#!/bin/sh

echo Creating user tpcc...
$tpcc_sqlplus $tpcc_dba_user_pass
@$tpcc_sql_dir/createuser > junk 2>&1
if test $? -ne 0
then
  exit 1;
else
  exit 0;
fi


::::::::::::::::::::::::::::::::
 createuser.sql
::::::::::::::::::::::::::::::::

spool createusertpcc.log;
```

```
set echo on;

create user tpcc identified by tpcc;

grant dba to tpcc;

set echo off;
spool off;

exit ;


::::::::::::::::::::::::::::::::
 ddview.sh
::::::::::::::::::::::::::::::::

#!/bin/sh

$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect

spool ddview.log


REM
REM In an ade/nde view we might need to run
standard.sql and dbmsstdx manually
REM catalog and catproc suppose to take care
of it
REM

@$ORACLE_HOME/plsql/admin/standard
@$ORACLE_HOME/rdbms/admin/dbmsstdx

@$ORACLE_HOME/rdbms/admin/catalog
@$ORACLE_HOME/rdbms/admin/catproc

REM
REM In an ade/nde view we might need to run
pupbld manually
REM catalog and catproc suppose to take care
of it
REM

connect system/manager
REM @$ORACLE_HOME/sqlplus/admin/pupbld


REM
REM Oracle
REM

REM if test $NUMBER_ORACLE_NODE -qt 1
REM then

REM @$ORACLE_HOME/rdbms/admin/catparr

REM fi

spool off
!



::::::::::::::::::::::::::::::::
 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;



::::::::::::::::::::::::::::::::
 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 / $tpcc_np`
  procpernode=`$tpcc_bcexpr $tpcc_cpu \* 2`
  curstuff=`$tpcc_bcexpr $warepernode \*
\( $tpcc_rac_node - 1 \) + 1`
  stuffinc=`expr $warepernode / $procpernode`
  stuffextra=`expr $warepernode \%
$procpernode`
  curloader=`$tpcc_bcexpr $procpernode \*
\( $tpcc_rac_node - 1 \) + 1`
  endloader=`$tpcc_bcexpr $procpernode \*
$tpcc_rac_node + 1`

  while expr $curloader \< $endloader >
/dev/null ; do

    newstuff=`expr $curstuff + $stuffinc +
\( $stuffextra / $procpernode \)`
    if expr x$6 = xt > /dev/null; then
      if expr $tpcc_os = unix > /dev/null; then

adddummy=\${tpcc_disks_location}dummy${curl
oader}.dat
      else
        # is this what we actually want to do?
check nt stuff
        adddummy=\\\\\\.\\\\dummy${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${curl
oader}.dat
      else
        # is this what we actually want to do?
check nt stuff
        adddummy=\\\\\\.\\\\dummy${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
fi

cat >> $loadout <<!
```

```
error=0
for curproc in \$allprocs; do
  wait \$curproc
  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/tabledata.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


::::::::::::::::::::::::::::::::
 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


::::::::::::::::::::::::::::::::
 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


::::::::::::::::::::::::::::::::
 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
!


::::::::::::::::::::::::::::::::
 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 "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 TPCdumppay ();
extern void TPCdumpord ();
extern void TPCdumpdel ();
extern void TPCdumpsto ();
extern void TPCdumpexit ();
extern void userlog(char* fmtp, ...);

/* Error codes */

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

#define FULLDATE "dd-mon-yy.hh24:mi:ss"
#define SHORTDATE "dd-mm-yyyy"



#define DELRT 80.0

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

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

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

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


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

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

extern int execstatus;
extern int errcode;

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



/* for stock-level transaction */

extern int w_id;
extern int d_id;
extern int c_id;
#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_last[17];
extern char c_first[17];
extern char c_middle[3];
extern double c_balance;
extern int o_id;
extern text o_entry_d[20];
extern int o_carrier_id;
extern int o_ol_cnt;
extern int ol_supply_w_id[15];
extern int ol_i_id[15];
#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 nol_i_id[15];
extern int nol_supply_w_id[15];
#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 */
extern int nol_quanti10[15];
extern int nol_quanti91[15];
extern int nol_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 ol_d_base[15];

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#define VER7          2

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

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

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

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

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

#define OCIERROR(errp,function)\

ocierror(__FILE__,__LINE__,(errp),(function));

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

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



/* bind arrays for sql */
#define
OCIBNDRA(stmp,bndp,errp,sqlvar,progv,progvl,f
type,indp,alen,arcode) \
     DISCARD
ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_H
TYPE_BIND,0,(dvoid**)0)); \
     DISCARD
ocierror(__FILE__,__LINE__,(errp), \

OCIBindByName((stmp),&(bndp),(errp),(text
*)(sqlvar),strlen((sqlvar)),\

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


/* use with callback data */
#define
OCIBNDRAD(stmp,bndp,errp,sqlvar,progvl,ftype
,indp,ctxp,\
          cbf_nodata,cbf_data) \
     DISCARD
ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_H
TYPE_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_nodat
a),(ctxp),(cbf_data)));



/* bind in/out for plsql without indicator and rcode
*/
#define
OCIBNDPL(stmp,bndp,errp,sqlvar,progv,progvl,f
type,alen) \
   DISCARD ocierror(__FILE__,__LINE__,(errp),
\

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_H
TYPE_BIND,0,(dvoid**)0)); \
   DISCARD ocierror(__FILE__,__LINE__,(errp),
\

OCIBindByName((stmp),&(bndp),(errp),(CONST
text *)(sqlvar), \
     (sb4)strlen((CONST char *)(sqlvar)),
(dvoid*)(progv),(progvl),(ftype),\
          NULLP(dvoid),(alen), NULLP(ub2),
0,NULLP(ub4),OCI_DEFAULT));

/* bind  in values for plsql with indicator and
rcode */
#define
OCIBNDR(stmp,bndp,errp,sqlvar,progv,progvl,ft
ype,indp,alen,arcode) \
     DISCARD
ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_H
TYPE_BIND,0,(dvoid**)0)); \
     DISCARD
ocierror(__FILE__,__LINE__,(errp), \

OCIBindByName((stmp),&(bndp),(errp),(text
*)(sqlvar),strlen((sqlvar)),\

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

/* bind in/out for plsql arrays witout indicator and
rcode */
#define
OCIBNDPLA(stmp,bndp,errp,sqlvar,progv,progvl
,ftype,alen,ms,cu) \
     DISCARD ocierror(__FILE__,__LINE__,
(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_H
TYPE_BIND,0,(dvoid**)0));\
     DISCARD
ocierror(__FILE__,__LINE__,(errp),\

OCIBindByName((stmp),&(bndp),(errp),(CONST
text *)(sqlvar), \
          (sb4)strlen((CONST char *)
(sqlvar)),(void *)(progv), \

(progvl),(ftype),NULL,(alen),NULL,(ms),(cu),OCI
_DEFAULT));

/* bind in/out values for plsql with indicator and
rcode */
#define
OCIBNDRAA(stmp,bndp,errp,sqlvar,progv,progv
l,ftype,indp,alen,arcode,\
          ms,cu) \
     ocierror(__FILE__,__LINE__, (errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_H
TYPE_BIND,0,(dvoid**)0));\
     ocierror(__FILE__,__LINE__,(errp),\

OCIBindByName((stmp),&(bndp),(errp),(text
*)(sqlvar),strlen((sqlvar)),\

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

#define
OCIDEFINE(stmp,dfnp,errp,pos,progv,progvl,fty
pe)\

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


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

```
OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HT
YPE_DEFINE,0,\
                (dvoid**)0);\

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(pro
gv),(progvl),\

(ftype),NULL,NULL,NULL,OCI_DEFAULT); \


#define
OCIDFNRA(stmp,dfnp,errp,pos,progv,progvl,ftyp
e,indp,alen,arcode) \

OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HT
YPE_DEFINE,0,\
                (dvoid**)0);\

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(pro
gv),\
                (progvl),(ftype),(indp),(alen),\
                (arcode),OCI_DEFAULT);

#define
OCIDFNDYN(stmp,dfnp,errp,pos,progv,progvl,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), (progvl),(ftype),\
                        (indp),NULL,NULL,
OCI_DYNAMIC_FETCH));\
     ocierror(__FILE__,__LINE__,(errp), \

OCIDefineDynamic((dfnp),(errp),(ctxp),(cbf_data
)));




/* New order */

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

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

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


/* Payment */

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

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

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


/* Order status */

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

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

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


/* Delivery */

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

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


::::::::::::::::::::::::::::::::
 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
|                     -s  load stok table (cluster around
s_w_id)
|                     -S  load stok table (cluster
around s_i_id)
|                     -h  load hist table
|                     -n  load new-order table
|                     -o <oline file> load order and
order-line table
|                     -b <ware#>  beginning ware
number
|                     -e <ware#>  ending ware number
|                     -j <item#>  beginning item
number (with -S)
|                     -k <item#>  ending item number
(with -S)
|                     -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 "dpbcore.h"
# define gettime dpbtimef
# define getcpu  dpbcpu
#define lrand48() ((long)rand() <<15 | rand())
#ifdef __STDC__
#  define PROTO(args)    args
#else
#  define PROTO(args)    ()
#endif
#endif

#define DISTARR 10      /* dist insert array size
*/
#define CUSTARR 100     /* cust insert array
size   */
#define STOCARR 100     /* stok insert array
size   */
#define ITEMARR 100     /* item insert array size
*/
#define HISTARR 100             /* hist insert array
size    */
#define ORDEARR 100             /* order insert
array size      */
#define NEWOARR 100             /* new order
insert array size  */

#define DISTFAC 10      /* max. dist id */
#define CUSTFAC 3000        /* max. cust id */
#define STOCFAC 100000      /* max. stok id */
#define ITEMFAC 100000      /* max. item id     */
#define HISTFAC 30000           /* history /
warehouse  */
#define ORDEFAC 3000            /* order / district
*/
#define NEWOFAC 900             /* new order /
district */

#define C       0               /* constant in non-
uniform dist. eqt. */
#define CNUM1   1               /* first constant in
non-uniform dist. eqt. */
#define CNUM2   2               /* second constant
in non-uniform dist. eqt. */
#define CNUM3   3               /* third constant in
non-uniform dist. eqt. */

#define SEED    2               /* seed for random
functions */

#define NOT_SERIALIZABLE  8177  /* ORA-
08177: transaction not serializable */
#define SNAPSHOT_TOO_OLD  1555  /* ORA-
01555: snapshot too old */
#define RECOVERR -10
#define IRRECERR -20

#define SQLTXTW "INSERT INTO ware (w_id,
w_ytd, w_tax, w_name, w_street_1, w_street_2,
w_city, w_state, w_zip) VALUES (:w_id,
30000000, :w_tax, :w_name, :w_street_1, \
   :w_street_2, :w_city, :w_state, :w_zip)"

#define SQLTXTD "INSERT INTO dist (d_id,
d_w_id, d_ytd, d_tax, d_next_o_id, d_name,
d_street_1, d_street_2, d_city, d_state, d_zip)
VALUES (:d_id, :d_w_id,3000000, :d_tax, \

3001, :d_name, :d_street_1, :d_street_2, :d_city,
 :d_state, :d_zip)"

#define SQLTXTCQUERY "select /*+ HASH
( cust ) */ count(*) from cust where c_w_id
= :s_c_w_id and c_d_id = :s_c_d_id and c_id
= :s_c_id"

#define SQLTXTC "INSERT INTO cust (C_ID,
C_D_ID, C_W_ID, C_FIRST, C_MIDDLE,
C_LAST, C_STREET_1, C_STREET_2,
C_CITY, C_STATE, C_ZIP, C_PHONE,
C_SINCE, C_CREDIT, C_CREDIT_LIM,
C_DISCOUNT, C_BALANCE,
C_YTD_PAYMENT, C_PAYMENT_CNT,
C_DELIVERY_CNT, C_DATA) VALUES
(:c_id, :c_d_id, :c_w_id, \
   :c_first,
'OE', :c_last, :c_street_1, :c_street_2, :c_city, :c_
state, \
   :c_zip, :c_phone, SYSDATE, :c_credit,
5000000, :c_discount, -1000, 1000, 1, \
   0, :c_data)"

#define SQLTXTH "INSERT INTO hist (h_c_id,
h_c_d_id, h_c_w_id, h_d_id, h_w_id, h_date,
h_amount, h_data) VALUES
(:h_c_id, :h_c_d_id, :h_c_w_id, \
   :h_d_id, :h_w_id, SYSDATE, 1000, :h_data)"

#define SQLTXTSQUERY "select /*+ HASH
( stok ) */ count(*) from stok where s_w_id
= :s_s_w_id and s_i_id = :s_s_i_id "

#define SQLTXTS "INSERT INTO stok (s_i_id,
s_w_id, s_quantity,s_dist_01, s_dist_02,
s_dist_03, s_dist_04, s_dist_05 , s_dist_06,
s_dist_07, s_dist_08, s_dist_09, s_dist_10,
s_ytd, s_order_cnt, s_remote_cnt, s_data) \
VALUES (:s_i_id, :s_w_id, :s_quantity, \
   :s_dist_01, :s_dist_02, :s_dist_03, :s_dist_04, :
s_dist_05, :s_dist_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 ordr (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 ordr (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 SQLTXTOL1 "INSERT INTO ordl
(OL_O_ID, OL_D_ID, OL_W_ID, OL_NUMBER,
OL_DELIVERY_D, OL_I_ID,
OL_SUPPLY_W_ID, OL_QUANTITY,
OL_AMOUNT, OL_DIST_INFO) \
   VALUES (:ol_o_id, :ol_d_id, \
   :ol_w_id, :ol_number,
SYSDATE, :ol_i_id, :ol_supply_w_id, 5, 0, \
   :ol_dist_info)"

#define SQLTXTOL2 "INSERT INTO ordl
(OL_O_ID, OL_D_ID, OL_W_ID, OL_NUMBER,
OL_DELIVERY_D, OL_I_ID,
OL_SUPPLY_W_ID, OL_QUANTITY,
OL_AMOUNT, OL_DIST_INFO) \
   VALUES (:ol_o_id, :ol_d_id, \
   :ol_w_id, :ol_number, to_date('01-Jan-
1811'), :ol_i_id, :ol_supply_w_id, 5, :ol_amount, \
   :ol_dist_info)"

#define SQLTXTNO "INSERT INTO nord
(no_o_id, no_d_id, no_w_id) VALUES
(:no_o_id, :no_d_id, :no_w_id)"

#define SQLTXTENHA "alter session set
\"_enable_hash_overflow\"=true"
#define SQLTXTDIHA "alter session set
\"_enable_hash_overflow\"=false"

static char *lastname[] = {
```

```
  "BAR",
  "OUGHT",
  "ABLE",
  "PRI",
  "PRES",
  "ESE",
  "ANTI",
  "CALLY",
  "ATION",
  "EING"
};

char num9[10];
char num16[17];
char str2[3];
char str24[15][25];
int randperm3000[3000];

void initperm();
void randstr();
void randdatastr();
void randnum();
void randlastname (char*, int);
int NURand();
void sysdate();

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

OCIStmt *curw;
OCIStmt *curd;
OCIStmt *curc;
OCIStmt *curcs;
OCIStmt *curh;
OCIStmt *curs;
OCIStmt *curss;
OCIStmt *curi;
OCIStmt *curo1;
OCIStmt *curo2;
OCIStmt *curol1;
OCIStmt *curol2;
OCIStmt *curno;

OCIBind *w_id_bp = (OCIBind *) 0;
OCIBind *w_name_bp = (OCIBind *) 0;
OCIBind *w_street1_bp = (OCIBind *) 0;
OCIBind *w_street2_bp = (OCIBind *) 0;
OCIBind *w_city_bp = (OCIBind *) 0;
OCIBind *w_state_bp = (OCIBind *) 0;
OCIBind *w_zip_bp = (OCIBind *) 0;
OCIBind *w_tax_bp = (OCIBind *) 0;

OCIBind *d_id_bp = (OCIBind *) 0;
OCIBind *d_w_id_bp = (OCIBind *) 0;
OCIBind *d_name_bp = (OCIBind *) 0;
OCIBind *d_street1_bp = (OCIBind *) 0;
OCIBind *d_street2_bp = (OCIBind *) 0;
OCIBind *d_city_bp = (OCIBind *) 0;
OCIBind *d_state_bp = (OCIBind *) 0;
OCIBind *d_zip_bp = (OCIBind *) 0;
OCIBind *d_tax_bp = (OCIBind *) 0;

OCIDefine *s_c_ret_bp = (OCIDefine *) 0;
OCIBind *s_c_id_bp = (OCIBind *) 0;
OCIBind *s_c_d_id_bp = (OCIBind *) 0;
OCIBind *s_c_w_id_bp = (OCIBind *) 0;

OCIBind *c_id_bp = (OCIBind *) 0;
OCIBind *c_d_id_bp = (OCIBind *) 0;
OCIBind *c_w_id_bp = (OCIBind *) 0;
OCIBind *c_first_bp = (OCIBind *) 0;
OCIBind *c_last_bp = (OCIBind *) 0;
OCIBind *c_street1_bp = (OCIBind *) 0;
OCIBind *c_street2_bp = (OCIBind *) 0;
OCIBind *c_city_bp = (OCIBind *) 0;
OCIBind *c_state_bp = (OCIBind *) 0;
OCIBind *c_zip_bp = (OCIBind *) 0;
OCIBind *c_phone_bp = (OCIBind *) 0;
OCIBind *c_discount_bp = (OCIBind *) 0;
OCIBind *c_credit_bp = (OCIBind *) 0;
OCIBind *c_data_bp = (OCIBind *) 0;

OCIBind *i_id_bp = (OCIBind *) 0;
OCIBind *i_im_id_bp = (OCIBind *) 0;
OCIBind *i_name_bp = (OCIBind *) 0;
OCIBind *i_price_bp = (OCIBind *) 0;
OCIBind *i_data_bp = (OCIBind *) 0;

OCIDefine *s_s_ret_bp = (OCIDefine *) 0;
OCIBind *s_s_i_id_bp = (OCIBind *) 0;
OCIBind *s_s_w_id_bp = (OCIBind *) 0;

OCIBind *s_i_id_bp = (OCIBind *) 0;
OCIBind *s_w_id_bp = (OCIBind *) 0;
OCIBind *s_quantity_bp = (OCIBind *) 0;
OCIBind *s_dist_01_bp = (OCIBind *) 0;
OCIBind *s_dist_02_bp = (OCIBind *) 0;
OCIBind *s_dist_03_bp = (OCIBind *) 0;
OCIBind *s_dist_04_bp = (OCIBind *) 0;
OCIBind *s_dist_05_bp = (OCIBind *) 0;
OCIBind *s_dist_06_bp = (OCIBind *) 0;
OCIBind *s_dist_07_bp = (OCIBind *) 0;
OCIBind *s_dist_08_bp = (OCIBind *) 0;
OCIBind *s_dist_09_bp = (OCIBind *) 0;
OCIBind *s_dist_10_bp = (OCIBind *) 0;
OCIBind *s_data_bp = (OCIBind *) 0;

OCIBind *h_c_id_bp = (OCIBind *) 0;
OCIBind *h_c_d_id_bp = (OCIBind *) 0;
OCIBind *h_c_w_id_bp = (OCIBind *) 0;
OCIBind *h_d_id_bp = (OCIBind *) 0;
OCIBind *h_w_id_bp = (OCIBind *) 0;
OCIBind *h_data_bp = (OCIBind *) 0;

OCIBind *ol_o_id_bp = (OCIBind *) 0;
OCIBind *ol_d_id_bp = (OCIBind *) 0;
OCIBind *ol_w_id_bp = (OCIBind *) 0;
OCIBind *ol_i_id_bp = (OCIBind *) 0;
OCIBind *ol_number_bp = (OCIBind *) 0;
OCIBind *ol_supply_w_id_bp = (OCIBind *) 0;
OCIBind *ol_dist_info_bp = (OCIBind *) 0;
OCIBind *ol_amount_bp = (OCIBind *) 0;

OCIBind *o_id_bp = (OCIBind *) 0;
OCIBind *o_d_id_bp = (OCIBind *) 0;
OCIBind *o_w_id_bp = (OCIBind *) 0;
OCIBind *o_c_id_bp = (OCIBind *) 0;
OCIBind *o_carrier_id_bp = (OCIBind *) 0;
OCIBind *o_ol_cnt_bp  = (OCIBind *) 0;
OCIBind *o_ocnt_bp = (OCIBind *) 0;
OCIBind *o_olcnt_bp  = (OCIBind *) 0;


OCIBind *no_o_id_bp  = (OCIBind *) 0;
OCIBind *no_d_id_bp  = (OCIBind *) 0;
OCIBind *no_w_id_bp  = (OCIBind *) 0;

void myusage()
{
  fprintf (stderr, "\n");
  fprintf (stderr, "Usage:\ttpccload -M
<multiplier> [options]\n");
  fprintf (stderr, "options:\n");
  fprintf (stderr, "\t-A :\tload all tables\n");
  fprintf (stderr, "\t-w :\tload ware table\n");
  fprintf (stderr, "\t-d :\tload dist table\n");
  fprintf (stderr, "\t-c :\tload cust table (cluster
around c_w_id\n");
  fprintf (stderr, "\t-C :\tload cust table (cluster
around c_id\n");
  fprintf (stderr, "\t-i :\tload item table\n");
  fprintf (stderr, "\t-s :\tload stok table (cluster
around s_w_id)\n");
  fprintf (stderr, "\t-S :\tload stok table (cluster
around s_i_id)\n");
  fprintf (stderr, "\t-h :\tload hist table\n");
  fprintf (stderr, "\t-n :\tload new-order table\n");
  fprintf (stderr, "\t-o <oline file> :\tload order and
order-line table\n");
  fprintf (stderr, "\t-b <ware#> :\tbeginning ware
number\n");
  fprintf (stderr, "\t-e <ware#> :\tending ware
number\n");
  fprintf (stderr, "\t-j <item#> :\tbeginning item
number (with -S)\n");
  fprintf (stderr, "\t-k <item#> :\tending item
number (with -S)\n");
  fprintf (stderr, "\t-l <cid#> :\tbeginning 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_SVCCTX);
  OCIHandleFree((dvoid *)errhp,
OCI_HTYPE_ERROR);
  OCIHandleFree((dvoid *)tpcsrv,
OCI_HTYPE_SERVER);
  OCIHandleFree((dvoid *)tpcenv,
OCI_HTYPE_ENV);
}

void main (argc, argv)
int argc;
```

```
char *argv[];
{
  char *uid="tpcc";
  char *pwd="tpcc";
  int scale=0;
  int i, j;
  int loop;
  int loopcount;
  int cid;
  int dwid;
  int cdid;
  int cwid;
  int sid;
  int swid;
  int olcnt;
  int nrows;
  int row;

  int w_id;
  char w_name[11];
  char w_street_1[21];
  char w_street_2[21];
  char w_city[21];
  char w_state[2];
  char w_zip[9];
  float w_tax;

  int d_id[10];
  int d_w_id[10];
  char d_name[10][11];
  char d_street_1[10][21];
  char d_street_2[10][21];
  char d_city[10][21];
  char d_state[10][2];
  char d_zip[10][9];
  float d_tax[10];

  int s_c_id;
  int s_c_d_id;
  int s_c_w_id;
  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)
    eware = 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 ((eware  < bware) || (eware > scale)) {
    fprintf (stderr, "Invalid ending warehouse
number: '%d'\n", eware);
    myusage ();
  }

  if (gen && do_o) {
    if ((olfp = fopen (olfname, "w")) == NULL) {
      fprintf (stderr, "Can't open '%s' for writing
order lines\n", olfname);
      myusage ();
    }


  }

/*----------------------------------------------------------------+
 | Prepare to insert into database.                               |
 +----------------------------------------------------------------
*/

  sysdate (sdate);
  if (!gen) {

    /* log on to Oracle */


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,OCIStmtPrepare(curw,
errhp, (text *)SQLTXTW,
               strlen((char *)SQLTXTW), (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,OCIStmtPrepare(curd,
errhp, (text *)SQLTXTD,
               strlen((char *)SQLTXTD), (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,OCIStmtPrepare(curc,
errhp, (text *)SQLTXTC,
               strlen((char *)SQLTXTC), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoi
d **)(&curcs), OCI_HTYPE_STMT, 0,
(dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curcs,
errhp, (text *)SQLTXTCQUERY,
               strlen((char *)SQLTXTCQUERY),
(ub4) OCI_NTV_SYNTAX, (ub4)
OCI_DEFAULT));
    }

    if (do_A || do_h) {

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoi
d **)(&curh), OCI_HTYPE_STMT, 0,
(dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curh,
errhp, (text *)SQLTXTH,
               strlen((char *)SQLTXTH), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_A || do_s || do_S) {

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoi
d **)(&curs), OCI_HTYPE_STMT, 0, (dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curs,
errhp, (text *)SQLTXTS,
               strlen((char *)SQLTXTS), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoi
d **)(&curss), OCI_HTYPE_STMT, 0,
(dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curss,
errhp, (text *)SQLTXTSQUERY,
               strlen((char *)SQLTXTSQUERY),
(ub4) OCI_NTV_SYNTAX, (ub4)
OCI_DEFAULT));
    }

    if (do_A || do_i) {

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoi
d **)(&curi), OCI_HTYPE_STMT, 0, (dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curi,
errhp, (text *)SQLTXTI,
               strlen((char *)SQLTXTI), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_A || do_o) {
      int stat;
      char fname[160];

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(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 = sqlfile(fname, stmbuf);
      if (!stat)
      {
        fprintf (stderr, "unable to open %s
\n",fname);
        quit();
        exit(1);
      }
      OCIERROR(errhp,OCIStmtPrepare(curo1,
errhp, stmbuf,
               strlen((char *)stmbuf), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_A || do_n) {

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(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));


OCIDefineByPos(curcs,&s_c_ret_bp,errhp,1,&s_
c_count,sizeof(int),SQLT_INT,\
            0,0,0,OCI_DEFAULT);

    OCIERROR(errhp, OCIBindByName(curc,
&c_id_bp, errhp, (text *)":c_id",
            strlen(":c_id"), (ub1 *)c_id,
sizeof(int), SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,
&c_d_id_bp, errhp, (text *)":c_d_id",
            strlen(":c_d_id"), (ub1 *)c_d_id,
sizeof(int), SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,
&c_w_id_bp, errhp, (text *)":c_w_id",
            strlen(":c_w_id"), (ub1 *)c_w_id,
sizeof(int), SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,
&c_first_bp, errhp, (text *)":c_first",
        strlen(":c_first"), (ub1 *)c_first, 17,
SQLT_STR,
        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,
&c_last_bp, errhp, (text *)":c_last",
        strlen(":c_last"), (ub1 *)c_last, 17,
SQLT_STR,
        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,
&c_street1_bp, errhp, (text *)":c_street_1",
            strlen(":c_street_1"), (ub1
*)c_street_1, 21, SQLT_STR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,
&c_street2_bp, errhp, (text *)":c_street_2",
            strlen(":c_street_2"), (ub1
*)c_street_2, 21, SQLT_STR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,
&c_city_bp, errhp, (text *)":c_city",
            strlen(":c_city"), (ub1 *)c_city, 21,
SQLT_STR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,
&c_state_bp, errhp, (text *)":c_state",
            strlen(":c_state"), (ub1 *)c_state, 2,
SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,
&c_zip_bp, errhp, (text *)":c_zip",
            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_c_w_id_bp, errhp, (text *)":h_c_w_id",
                strlen(":h_c_w_id"), (ub1 *)h_w_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curh,
&h_d_id_bp, errhp, (text *)":h_d_id",
                strlen(":h_d_id"), (ub1 *)h_d_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curh,
&h_w_id_bp, errhp, (text *)":h_w_id",
                strlen(":h_w_id"), (ub1 *)h_w_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curh,
&h_data_bp, errhp, (text *)":h_data",
                strlen(":h_data"), (ub1 *)h_data, 25,
SQLT_STR,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
```

```
    }

    /* order and order_line (delivered) */

    if (do_A || do_o) {

      for (i = 0; i < ORDEARR; i++ ) {
        o_id_len[i] = sizeof(int);
        o_d_id_len[i] = sizeof(int);
        o_w_id_len[i] = sizeof(int);
        o_c_id_len[i] = sizeof(int);
        o_carrier_id_len[i] = sizeof(int);
        o_ol_cnt_len[i] = sizeof(int);
      }

      OCIERROR(errhp, OCIBindByName(curo1,
&ol_o_id_bp, errhp, (text *)":ol_o_id",
                strlen(":ol_o_id"), (ub1 *)ol_o_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)ol_o_id_len,
(ub2 *)0,
                (ub4) 15*ORDEARR, (ub4
*)&ol_o_id_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&ol_d_id_bp, errhp, (text *)":ol_d_id",
                strlen(":ol_d_id"), (ub1 *)ol_d_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)ol_d_id_len,
(ub2 *)0,
                (ub4) 15*ORDEARR, (ub4 *)
&ol_d_id_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&ol_w_id_bp, errhp, (text *)":ol_w_id",
                strlen(":ol_w_id"), (ub1 *)ol_w_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)ol_w_id_len,
(ub2 *)0,
                (ub4) 15*ORDEARR, (ub4 *)
&ol_w_id_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&ol_number_bp, errhp, (text *)":ol_number",
                strlen(":ol_number"), (ub1
*)ol_number, sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)ol_number_len,
(ub2 *)0,
                (ub4) 15*ORDEARR, (ub4 *)
&ol_number_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&ol_i_id_bp, errhp, (text *)":ol_i_id",
                strlen(":ol_i_id"), (ub1 *)ol_i_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)ol_i_id_len, (ub2
*)0,
                (ub4) 15*ORDEARR, (ub4 *)
&ol_i_id_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&ol_supply_w_id_bp, errhp, (text
*)":ol_supply_w_id",
                strlen(":ol_supply_w_id"), (ub1
*)ol_supply_w_id, sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2
*)ol_supply_w_id_len, (ub2 *)0,
                (ub4) 15*ORDEARR, (ub4 *)
&ol_supply_w_id_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&ol_dist_info_bp, errhp, (text *)":ol_dist_info",
                strlen(":ol_dist_info"), (ub1
*)ol_dist_info, 24, SQLT_CHR,
                (dvoid *) 0, (ub2 *)ol_dist_info_len,
(ub2 *)0,
                (ub4) 15*ORDEARR, (ub4 *)
&ol_dist_info_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&ol_amount_bp, errhp, (text *)":ol_amount",
                strlen(":ol_amount"), (ub1
*)ol_amount, sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)ol_amount_len,
(ub2 *)0,
                (ub4) 15*ORDEARR, (ub4 *)
&ol_amount_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&o_id_bp, errhp, (text *)":o_id",
                strlen(":o_id"), (ub1 *)o_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)o_id_len, (ub2
*)0,
                (ub4) ORDEARR, (ub4 *)
&o_id_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&o_d_id_bp, errhp, (text *)":o_d_id",
                strlen(":o_d_id"), (ub1 *)o_d_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)o_d_id_len, (ub2
*)0,
                (ub4) ORDEARR, (ub4 *)
&o_d_id_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&o_w_id_bp, errhp, (text *)":o_w_id",
                strlen(":o_w_id"), (ub1 *)o_w_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)o_w_id_len,
(ub2 *)0,
                (ub4) ORDEARR, (ub4 *)
&o_w_id_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&o_c_id_bp, errhp, (text *)":o_c_id",
                strlen(":o_c_id"), (ub1 *)o_c_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)o_c_id_len, (ub2
*)0,
                (ub4) ORDEARR, (ub4 *)
&o_c_id_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&o_carrier_id_bp, errhp, (text *)":o_carrier_id",
                strlen(":o_carrier_id"), (ub1
*)o_carrier_id, sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)o_carrier_id_len,
(ub2 *)0,
                (ub4) ORDEARR, (ub4 *)
&o_carrier_id_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&o_ol_cnt_bp, errhp, (text *)":o_ol_cnt",
                strlen(":o_ol_cnt"), (ub1 *)o_ol_cnt,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)o_ol_cnt_len,
(ub2 *)0,
                (ub4) ORDEARR, (ub4 *)
&o_ol_cnt_clen, (ub4) OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&o_ocnt_bp, errhp, (text *)":order_rows",
                strlen(":order_rows"), (ub1
*)&o_cnt, sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curo1,
&o_olcnt_bp, errhp, (text *)":ordl_rows",
                strlen(":ordl_rows"), (ub1 *)&ol_cnt,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
    }

    /* new order */

    if (do_A || do_n) {
      OCIERROR(errhp, OCIBindByName(curno,
&no_o_id_bp, errhp, (text *)":no_o_id",
                strlen(":no_o_id"), (ub1 *)no_o_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curno,
&no_d_id_bp, errhp, (text *)":no_d_id",
                strlen(":no_d_id"), (ub1 *)no_d_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curno,
&no_w_id_bp, errhp, (text *)":no_w_id",
                strlen(":no_w_id"), (ub1 *)no_w_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
    }
  }

/*---------------------------------------------------------------+
 | Initialize random number generator                            |
 +---------------------------------------------------------------
*/

  srand (SEED);
#ifndef ORA_NT
  srand48 (SEED);
#endif
  initperm ();


/*---------------------------------------------------------------+
 | Load the WAREHOUSE table.                                     |
 +---------------------------------------------------------------
*/

  if (do_A || do_w) {
    nrows = eware - bware + 1;

    fprintf (stderr, "Loading/generating
warehouse: w%d - w%d (%d rows)\n",
             bware, eware, nrows);

    begin_time = gettime ();
    begin_cpu = getcpu ();

    for (loop = bware; loop <= eware; loop++) {

      w_tax = (float) ((lrand48 () % 2001) *
0.0001);
      randstr (w_name, 6, 10);
      randstr (w_street_1, 10, 20);
      randstr (w_street_2, 10, 20);
      randstr (w_city, 10, 20);
      randstr (str2, 2, 2);
```

```
        randnum (num9, 9);
        num9[4] = num9[5] = num9[6] = num9[7] =
num9[8] = '1';

        if (gen) {
          printf ("%d
30000000 %6.4f %s %s %s %s %s %s\n", loop,
w_tax,
                  w_name, w_street_1, w_street_2,
w_city, str2, num9);
          fflush (stdout);
        }
        else {
          w_id = loop;
          strncpy (w_state, str2, 2);
          strncpy (w_zip, num9, 9);

        status = OCIStmtExecute(tpcsvc, curw,
errhp, (ub4) 1, (ub4) 0,
                    (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                    (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
        if (status != OCI_SUCCESS) {
          fprintf (stderr, "Error at ware %d\n", loop);
          OCIERROR(errhp, status);
          quit ();
            exit (1);
          }
        }
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows
loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
               nrows, end_time - begin_time, end_cpu
- begin_cpu);
    }

/*---------------------------------------------------------------+
 | Load the DISTRICT table.                 |
 +---------------------------------------------------------------
*/

    if (do_A || do_d) {
      nrows = (eware - bware + 1) * DISTFAC;

      fprintf (stderr, "Loading/generating district:
w%d - w%d (%d rows)\n",
               bware, eware, nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      dwid = bware - 1;

      for (row = 0; row < nrows; ) {
        dwid++;

        for (i = 0; i < DISTARR; i++, row++) {
          d_tax[i] = (float) ((lrand48 () % 2001) *
0.0001);
          randstr (d_name[i], 6, 10);
          randstr (d_street_1[i], 10, 20);
          randstr (d_street_2[i], 10, 20);
          randstr (d_city[i], 10, 20);
          randstr (str2, 2, 2);
          randnum (num9, 9);
          num9[4] = num9[5] = num9[6] = num9[7]
= num9[8] = '1';

          if (gen) {
            printf ("%d %d 3000000 %6.4f
3001 %s %s %s %s %s %s\n",
                    i + 1, dwid, d_tax[i], d_name[i],
d_street_1[i],
                    d_street_2[i], d_city[i], str2, num9 );
          }
          else {
            d_id[i] = i + 1;
            d_w_id[i] = dwid;
            strncpy (d_state[i], str2, 2);
            strncpy (d_zip[i], num9, 9);
          }
        }

        if (gen) {
          fflush (stdout);
        }
        else {
          status = OCIStmtExecute(tpcsvc, curd,
errhp, (ub4) DISTARR, (ub4) 0,
                    (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                    (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
          if (status != OCI_SUCCESS) {
         fprintf (stderr, "Aborted at ware %d, dist
1\n", dwid);
                OCIERROR(errhp, status);
                quit ();
                exit (1);
          }
        }
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows
loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
               nrows, end_time - begin_time, end_cpu
- begin_cpu);
    }

/*---------------------------------------------------------------+
 | Load the CUSTOMER table.                 |
 +---------------------------------------------------------------
*/

    if (do_A || do_c) {

      nrows = (eware - bware + 1) * CUSTFAC *
DISTFAC;

      fprintf (stderr, "Loading/generating customer:
w%d - w%d (%d rows)\n   ",
               bware, eware, nrows);

      if (getenv("tpcc_hash_overflow")) {
        fprintf(stderr,"Hash overflow is enabled\n");
        OCIHandleAlloc(tpcenv, (dvoid **)&curi,
OCI_HTYPE_STMT, 0, (dvoid**)0);
        sprintf ((char *) stmbuf, SQLTXTENHA);
        OCIStmtPrepare(curi, errhp, stmbuf,
strlen((char *)stmbuf),
                       OCI_NTV_SYNTAX,
OCI_DEFAULT);
        OCIERROR(errhp,OCIStmtExecute(tpcsvc,
curi, errhp,1,0,0,0,OCI_DEFAULT));
        OCIHandleFree(curi, OCI_HTYPE_STMT);
        fprintf (stderr,"Customer loaded for
horizontal partitioning\n");
      }
      else
      {
        fprintf (stderr,"Customer not loaded for
horizontal partitioning\n");
      }
      begin_time = gettime ();
      begin_cpu = getcpu ();

      s_c_id = 1;
      s_c_d_id = 1;
      s_c_w_id = bware;

      while (s_c_w_id <= eware) {
        status = OCIStmtExecute(tpcsvc, curcs,
errhp, (ub4) 1, (ub4) 0,
                (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
          if (status != OCI_SUCCESS) {
              OCIERROR(errhp, status);
              quit ();
              exit (1);
          }

          if (s_c_count == 0) {
            s_c_w_id--;
            break;
          }
          else s_c_w_id++;
      }

      if (s_c_w_id < bware ) s_c_w_id = bware;
      else {
        if (s_c_w_id > eware ) s_c_w_id = eware;
        while (s_c_d_id <= DISTFAC) {
          status = OCIStmtExecute(tpcsvc, curcs,
errhp, (ub4) 1, (ub4) 0,
                (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
            if (status != OCI_SUCCESS) {
            fprintf (stderr, "Select failed\n");
                OCIERROR(errhp, status);
                quit ();
                exit (1);
            }
            if (s_c_count == 0) {
              s_c_d_id--;
              break;
            }
            else s_c_d_id++;
        }
        if (s_c_d_id > DISTFAC) s_c_d_id =
DISTFAC;

        while (s_c_id <= CUSTFAC) {
          status = OCIStmtExecute(tpcsvc, curcs,
errhp, (ub4) 1, (ub4) 0,
                (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
            if (status != OCI_SUCCESS) {
                OCIERROR(errhp, status);
                quit ();
                exit (1);
            }
            if (s_c_count == 0) break;
            else s_c_id++;
        }
      }
      if (s_c_id > CUSTFAC) {
        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 = (eware - s_c_w_id + 1) * DISTFAC *
CUSTFAC - (s_c_d_id - 1) * CUSTFAC - s_c_id
+ 1;
    fprintf (stderr, "remaining rows: %d\n   ",
nrows);
    loopcount = 0;

    for (row = 0; row < nrows; ) {
      for (i = 0; i < CUSTARR && row < nrows;
i++, row++) {
        cid++;
        if (cid > CUSTFAC) {       /* cycle cust id
*/
          cid = 1;                 /* cheap mod */
          cdid++;                  /* shift dist cycle */
          if (cdid > DISTFAC) {
            cdid = 1;
            cwid++;                /* shift ware cycle */
          }
        }
        c_id[i] = cid;
        c_d_id[i] = cdid;
        c_w_id[i] = cwid;
        if (cid <= 1000)
          randlastname (c_last[i], cid - 1);
        else
          randlastname (c_last[i], NURand (255,
0, 999, CNUM1));
        c_credit[i][1] = 'C';
        if (lrand48 () % 10)
          c_credit[i][0] = 'G';
        else
          c_credit[i][0] = 'B';
        c_discount[i] = (float)((lrand48 () % 5001)
* 0.0001);
        randstr (c_first[i], 8, 16);
        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 %cC
5000000 %6.4f -1000 1000 1 0 %s\n",
                  cid, cdid, cwid, c_first[i], c_last[i],
                  c_street_1[i], c_street_2[i], c_city[i],
str2, num9,
                  num16, sdate, c_credit[i][0],
c_discount[i], c_data[i]);
        }
        else {
          strncpy (c_state[i], str2, 2);
          strncpy (c_zip[i], num9, 9);
          strncpy (c_phone[i], num16, 16);
        }
      }

      if (gen) {
        fflush (stdout);
      }
      else {
        status = OCIStmtExecute(tpcsvc, curc,
errhp, (ub4) i, (ub4) 0,
                (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);

        if (status != OCI_SUCCESS) {
       fprintf (stderr, "Aborted at w_id %d,
d_id %d, c_id %d\n",
                  c_w_id[0], c_d_id[0], c_id[0]);
          OCIERROR(errhp, status);
          quit ();
          exit (1);
        }
      }

      if ((++loopcount) % 50)
        fprintf (stderr, ".");
      else
        fprintf (stderr, " %d rows committed\n   ",
row);
    }

    end_time = gettime ();
    end_cpu = getcpu ();
    fprintf (stderr, "Done.  %d rows
loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
          nrows < 0 ? 0 : nrows, end_time -
begin_time, end_cpu - begin_cpu);
    if (getenv("tpcc_hash_overflow")) {
      fprintf(stderr,"Hash overflow is disabled\n");
      OCIHandleAlloc(tpcenv, (dvoid **)&curi,
OCI_HTYPE_STMT, 0, (dvoid**)0);
      sprintf ((char *) stmbuf, SQLTXTDIHA);
      OCIStmtPrepare(curi, errhp, stmbuf,
strlen((char *)stmbuf),
                OCI_NTV_SYNTAX,
OCI_DEFAULT);
      OCIERROR(errhp,OCIStmtExecute(tpcsvc,
curi, errhp,1,0,0,0,OCI_DEFAULT));
      OCIHandleFree(curi, OCI_HTYPE_STMT);
    }
  }


/*---------------------------------------------------------+
 | Load the CUSTOMER table (cluster around
c_id)       |
 +---------------------------------------------------------
*/

  if (do_C) {

    srand (bcid);
#ifndef 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, SQLTXTENHA);
      OCIStmtPrepare(curi, errhp, stmbuf,
strlen((char *)stmbuf),
                OCI_NTV_SYNTAX,
OCI_DEFAULT);
      OCIERROR(errhp,OCIStmtExecute(tpcsvc,
curi, errhp,1,0,0,0,OCI_DEFAULT));
      OCIHandleFree(curi, OCI_HTYPE_STMT);
      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 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_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 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_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 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_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 %cC
5000000 %6.4f -1000 1000 1 0 %s\n",
                  cid, cdid, cwid, c_first[i], c_last[i],
                  c_street_1[i], c_street_2[i], c_city[i],
str2, num9,
                  num16, sdate, c_credit[i][0],
c_discount[i], c_data[i]);
        }
        else {
          strncpy (c_state[i], str2, 2);
          strncpy (c_zip[i], num9, 9);
          strncpy (c_phone[i], num16, 16);
        }
      }

      if (gen) {
        fflush (stdout);
      }
      else {
        status = OCIStmtExecute(tpcsvc, curc,
errhp, (ub4) i, (ub4) 0,
                  (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                  (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);

        if (status != OCI_SUCCESS) {
       fprintf (stderr, "Aborted at w_id %d,
d_id %d, c_id %d\n",
                  c_w_id[0], c_d_id[0], c_id[0]);
          OCIERROR(errhp, status);
          quit ();
          exit (1);
        }
      }

      if ((++loopcount) % 50)
        fprintf (stderr, ".");
      else
        fprintf (stderr, " %d rows committed\n   ",
row);
    }

    end_time = gettime ();
    end_cpu = getcpu ();
    fprintf (stderr, "Done.  %d rows
loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
          nrows < 0 ? 0 : nrows, end_time -
begin_time, end_cpu - begin_cpu);
    if (getenv("tpcc_hash_overflow")) {
      fprintf(stderr,"Hash overflow is disabled\n");
      OCIHandleAlloc(tpcenv, (dvoid **)&curi,
OCI_HTYPE_STMT, 0, (dvoid**)0);
      sprintf ((char *) stmbuf, SQLTXTDIHA);
      OCIStmtPrepare(curi, errhp, stmbuf,
strlen((char *)stmbuf),
                  OCI_NTV_SYNTAX,
OCI_DEFAULT);
      OCIERROR(errhp,OCIStmtExecute(tpcsvc,
curi, errhp,1,0,0,0,OCI_DEFAULT));
      OCIHandleFree(curi, OCI_HTYPE_STMT);
    }
  }


/*---------------------------------------------------------+
 | Load the ITEM table.                  |
 +---------------------------------------------------------
*/

  if (do_A || do_i) {
    nrows = ITEMFAC;

    fprintf (stderr, "Loading/generating item: (%d
rows)\n   ", nrows);

    begin_time = gettime ();
    begin_cpu = getcpu ();

    loopcount = 0;

    for (row = 0; row < nrows; ) {
      for (i = 0; i < ITEMARR; i++, row++) {
        i_im_id[i] = (lrand48 () % 10000) + 1;
        i_price[i] = ((lrand48 () % 9901) + 100);
        randstr (i_name[i], 14, 24);
        randdatastr (i_data[i], 26, 50);

        if (gen) {
          printf ("%d %d %s %d %s\n", row + 1,
i_im_id[i], i_name[i],
                  i_price[i], i_data[i]);
        }
        else {
          i_id[i] = row + 1;
        }
      }

      if (gen) {
        fflush (stdout);
      }
      else {
        status = OCIStmtExecute(tpcsvc, curi,
errhp, (ub4) ITEMARR, (ub4) 0,
                  (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                  (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
        if (status != OCI_SUCCESS) {
       fprintf (stderr, "Aborted at i_id %d\n",
i_id[0]);
          OCIERROR(errhp, status);
          quit ();
          exit (1);
        }
      }

      if ((++loopcount) % 50)
        fprintf (stderr, ".");
      else
        fprintf (stderr, " %d rows committed\n   ",
row);
    }

    end_time = gettime ();
    end_cpu = getcpu ();
    fprintf (stderr, "Done.  %d rows
loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
          nrows, end_time - begin_time, end_cpu
- begin_cpu);
  }


/*---------------------------------------------------------+
 | Load the STOCK table.                 |
 +---------------------------------------------------------
*/

  if (do_A || do_s) {

    nrows = (eware - bware + 1) * STOCFAC;

    fprintf (stderr, "Loading/generating stock:
w%d - w%d (%d rows)\n   ",
          bware, eware, nrows);

    begin_time = gettime ();
    begin_cpu = getcpu ();

    s_s_i_id = 1;
```

```
    s_s_w_id = bware;

    while (s_s_w_id <= eware) {
      status = OCIStmtExecute(tpcsvc, curss,
errhp, (ub4) 1, (ub4) 0,
            (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
            (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
        if (status != OCI_SUCCESS) {
            OCIERROR(errhp, status);
            quit ();
            exit (1);
        }
        if (s_s_count == 0) {
          s_s_w_id--;
          break;
        }
        else s_s_w_id++;
    }

    if (s_s_w_id < bware ) s_s_w_id = bware;
    else {
      if (s_s_w_id > eware) s_s_w_id = eware;
      while (s_s_i_id<=STOCFAC) {
        status = OCIStmtExecute(tpcsvc, curss,
errhp, (ub4) 1, (ub4) 0,
            (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
            (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
          if (status != OCI_SUCCESS) {
            OCIERROR(errhp, status);
            quit ();
            exit (1);
          }
          if (s_s_count == 0) {
            break;
          }
          else s_s_i_id++;
      }
    }
    if (s_s_i_id > STOCFAC) {
      s_s_i_id=1;
      s_s_w_id++;
    }

    fprintf(stderr,"start at s_i_id: %d,
s_w_id: %d\n   ", s_s_i_id, s_s_w_id);

    sid = s_s_i_id - 1;
    swid = s_s_w_id;
    nrows = (eware - s_s_w_id + 1) * STOCFAC
- ( s_s_i_id - 1);
    fprintf (stderr, "remaining rows: %d\n   ",
nrows);
    loopcount = 0;

    for (row = 0; row < nrows; ) {
      /* added row < nrows condition on next
line - alex.ni */
   for (i = 0; (i < STOCARR) && (row < nrows);
i++, row++) {
        if (++sid > STOCFAC) {       /* cheap
mod */
          sid = 1;
          swid++;
        }
        s_quantity[i] = (lrand48 () % 91) + 10;
        randstr (s_dist_01[i], 24, 24);
        randstr (s_dist_02[i], 24, 24);
        randstr (s_dist_03[i], 24, 24);
        randstr (s_dist_04[i], 24, 24);
        randstr (s_dist_05[i], 24, 24);
        randstr (s_dist_06[i], 24, 24);
        randstr (s_dist_07[i], 24, 24);
        randstr (s_dist_08[i], 24, 24);
        randstr (s_dist_09[i], 24, 24);
        randstr (s_dist_10[i], 24, 24);
        randdatastr (s_data[i], 26, 50);

        if (gen) {
          printf
("%d %d %d %s %s %s %s %s %s %s %s %s %
s 0 0 0 %s\n",
                  sid, swid, s_quantity[i], s_dist_01[i],
s_dist_02[i],
                  s_dist_03[i], s_dist_04[i],
s_dist_05[i], s_dist_06[i],
                  s_dist_07[i], s_dist_08[i],
s_dist_09[i], s_dist_10[i],
                  s_data[i]);
        }
        else {
          s_i_id[i] = sid;
          s_w_id[i] = swid;
        }
      }

      if (gen) {
        fflush (stdout);
      }
      else {
    /* Changed to STOCKARR to i - alex.ni */
        status = OCIStmtExecute(tpcsvc, curs,
errhp, (ub4) i, (ub4) 0,
                (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
        if (status != OCI_SUCCESS) {
          fprintf (stderr, "Aborted at w_id %d,
s_i_id %d\n", s_w_id[0], s_i_id[0]);
          OCIERROR(errhp, status);
          quit ();
          exit (1);
        }
      }

      if ((++loopcount) % 50)
        fprintf (stderr, ".");
      else
        fprintf (stderr, " %d rows committed\n   ",
row);
    }

    end_time = gettime ();
    end_cpu = getcpu ();
    fprintf (stderr, "Done.  %d rows
loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
          nrows < 0 ? 0 : nrows, end_time -
begin_time, end_cpu - begin_cpu);
  }

/*-----------------------------------------------------------------+
 | Load the STOCK table (cluster around s_i_id).
|
 +-----------------------------------------------------------------
*/

  if (do_S) {

    nrows = (eitem - bitem + 1) * (eware - bware
+ 1);

    fprintf (stderr, "Loading/generating stock: i%d
- i%d, w%d - w%d (%d rows)\n   ",
          bitem, eitem, bware, eware, nrows);

    begin_time = gettime ();
    begin_cpu = getcpu ();

    s_s_i_id = bitem;
    s_s_w_id = bware;

    while (s_s_i_id <= eitem) {
      status = OCIStmtExecute(tpcsvc, curss,
errhp, (ub4) 1, (ub4) 0,
            (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
            (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
        if (status != OCI_SUCCESS) {
            OCIERROR(errhp, status);
            quit ();
            exit (1);
        }
        if (s_s_count == 0) {
          s_s_i_id--;
          break;
        }
        else s_s_i_id++;
    }

    if (s_s_i_id < bitem ) s_s_i_id = bitem;
    else {
      if (s_s_i_id > eitem) s_s_i_id = eitem;
      while (s_s_w_id <= eware) {
        status = OCIStmtExecute(tpcsvc, curss,
errhp, (ub4) 1, (ub4) 0,
            (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
            (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
          if (status != OCI_SUCCESS) {
            OCIERROR(errhp, status);
            quit ();
            exit (1);
          }
          if (s_s_count == 0) {
            break;
          }
          else s_s_w_id++;
      }
    }
    if (s_s_w_id > eware) {
      s_s_w_id=bware;
      s_s_i_id++;
    }

    fprintf(stderr,"start at s_i_id: %d,
s_w_id: %d\n   ", s_s_i_id, s_s_w_id);

    sid = s_s_i_id;
    swid = s_s_w_id - 1;
    nrows = (eitem - s_s_i_id + 1) * (eware -
bware + 1) - (s_s_w_id - bware);
    fprintf (stderr, "remaining rows: %d\n   ",
nrows);
    loopcount = 0;

    for (row = 0; row < nrows; ) {
      for (i = 0; i < STOCARR && row < nrows;
i++, row++) {
        if (++swid > eware) {        /* cheap mod
*/
          swid = bware;
          sid++;
        }
        s_quantity[i] = (lrand48 () % 91) + 10;
        randstr (s_dist_01[i], 24, 24);
        randstr (s_dist_02[i], 24, 24);
        randstr (s_dist_03[i], 24, 24);
        randstr (s_dist_04[i], 24, 24);
```

```
        randstr (s_dist_05[i], 24, 24);
        randstr (s_dist_06[i], 24, 24);
        randstr (s_dist_07[i], 24, 24);
        randstr (s_dist_08[i], 24, 24);
        randstr (s_dist_09[i], 24, 24);
        randstr (s_dist_10[i], 24, 24);
        randdatastr (s_data[i], 26, 50);

        if (gen) {
          printf
("%d %d %d %s %s %s %s %s %s %s %s %s %
s 0 0 0 %s\n",
                  sid, swid, s_quantity[i], s_dist_01[i],
s_dist_02[i],
                  s_dist_03[i], s_dist_04[i],
s_dist_05[i], s_dist_06[i],
                  s_dist_07[i], s_dist_08[i],
s_dist_09[i], s_dist_10[i],
                  s_data[i]);
        }
        else {
          s_i_id[i] = sid;
          s_w_id[i] = swid;
        }
      }

      if (gen) {
        fflush (stdout);
      }
      else {
        status = OCIStmtExecute(tpcsvc, curs,
errhp, (ub4) i, (ub4) 0,
                  (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                  (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
        if (status != OCI_SUCCESS) {
          fprintf (stderr, "Aborted at w_id %d,
s_i_id %d\n", s_w_id[0], s_i_id[0]);
          OCIERROR(errhp, status);
          quit ();
          exit (1);
        }
      }

      if ((++loopcount) % 50)
        fprintf (stderr, ".");
      else
        fprintf (stderr, " %d rows committed\n   ",
row);
    }

    end_time = gettime ();
    end_cpu = getcpu ();
    fprintf (stderr, "Done.  %d rows
loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
          nrows < 0 ? 0 : nrows, end_time -
begin_time, end_cpu - begin_cpu);
  }

/*-----------------------------------------------------------------+
 | Load the HISTORY table.                 |
 +-----------------------------------------------------------------
*/

  if (do_A || do_h) {
    nrows = (eware - bware + 1) * HISTFAC;

    fprintf (stderr, "Loading/generating history:
w%d - w%d (%d rows)\n   ",
          bware, eware, nrows);

    begin_time = gettime ();
    begin_cpu = getcpu ();

    cid = 0;
    cdid = 1;
    cwid = bware;
    loopcount = 0;

    for (row = 0; row < nrows; ) {
      for (i = 0; i < HISTARR; i++, row++) {
        cid++;
        if (cid > CUSTFAC) {       /* cycle cust id
*/
          cid = 1;               /* cheap mod */
          cdid++;                /* shift district cycle
*/
          if (cdid > DISTFAC) {
            cdid = 1;
            cwid++;              /* shift warehouse
cycle */
          }
        }
        h_c_id[i] = cid;
        h_d_id[i] = cdid;
        h_w_id[i] = cwid;
        randstr (h_data[i], 12, 24);
        if (gen) {
          printf ("%d %d %d %d %d %s
1000 %s\n", cid, cdid, cwid, cdid,
                  cwid, sdate, h_data[i]);
        }
      }

      if (gen) {
        fflush (stdout);
      }
      else {
        status = OCIStmtExecute(tpcsvc, curh,
errhp, (ub4) HISTARR, (ub4) 0,
                  (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                  (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
        if (status != OCI_SUCCESS) {
          fprintf (stderr, "Aborted at w_id %d,
d_id %d, c_id %d\n",
                  h_w_id[0], h_d_id[0], h_c_id[0]);
          OCIERROR(errhp, status);
          quit ();
          exit (1);
        }
      }

      if ((++loopcount) % 50)
        fprintf (stderr, ".");
      else
        fprintf (stderr, " %d rows committed\n   ",
row);
    }

    end_time = gettime ();
    end_cpu = getcpu ();
    fprintf (stderr, "Done.  %d rows
loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
          nrows, end_time - begin_time, end_cpu
- begin_cpu);
  }

/*-----------------------------------------------------------------+
 | Load the ORDERS and ORDER-LINE table.
|
 +-----------------------------------------------------------------
*/

  if (do_A || do_o) {
    int batch_olcnt;

    nrows = (eware - bware + 1) * ORDEFAC *
DISTFAC;

    fprintf (stderr, "Loading/generating orders
and order-line: w%d - w%d (%d ord, ~%d ordl)\n
",
          bware, eware, nrows, nrows * 10);

    begin_time = gettime ();
    begin_cpu = getcpu ();

    cid = 0;
    cdid = 1;
    cwid = bware;
    loopcount = 0;

    for (row = 0; row < nrows; ) {

      batch_olcnt = 0;

      for (i = 0; i < ORDEARR; i++, row++) {
        cid++;
        if (cid > ORDEFAC) {       /* cycle cust id
*/
          cid = 1;               /* cheap mod */
          cdid++;                /* shift district cycle
*/
          if (cdid > DISTFAC) {
            cdid = 1;
            cwid++;              /* shift warehouse
cycle */
          }
        }
        o_carrier_id[i] = lrand48 () % 10 + 1;
        o_ol_cnt[i] = olcnt = lrand48 () % 11 + 5;

        if (gen) {
          if (cid < 2101) {
            printf ("%d %d %d %d %s %d %d
1\n", cid, cdid, cwid,
                  randperm3000[cid - 1],
sdate,o_carrier_id[i],
                  o_ol_cnt[i]);
          }
          else {
           /* set carrierid to 11 instead of null */
            printf ("%d %d %d %d %s 11 %d 1\n",
cid, cdid, cwid,
                  randperm3000[cid - 1], sdate,
o_ol_cnt[i]);
          }
        }
        else {
          o_id[i] = cid;
          o_d_id[i] = cdid;
          o_w_id[i] = cwid;
          o_c_id[i] = randperm3000[cid - 1];
          if (cid >= 2101 ) {
            o_carrier_id[i] = 11;
          }
        }

        for (j = 0; j < o_ol_cnt[i]; j++,
batch_olcnt++ ) {
          ol_i_id[batch_olcnt] = sid = lrand48 () %
100000 + 1;
          if (cid < 2101)
            ol_amount[batch_olcnt] = 0;
          else
            ol_amount[batch_olcnt] = (lrand48
() % 999999 + 1) ;
          randstr (str24[j], 24, 24);
```

```
            if (gen) {
              if (cid < 2101) {
                fprintf (olfp,
"%d %d %d %d %s %d %d 5 %ld %s\n", cid,
                        cdid, cwid, j + 1, sdate,
ol_i_id[batch_olcnt], cwid,
                        ol_amount[batch_olcnt],
str24[j]);
              }
              else {
                /* Insert a default date instead of
null date */
                fprintf (olfp, "%d %d %d %d 01-Jan-
1811 %d %d 5 %ld %s\n", cid,
                        cdid, cwid, j + 1,
ol_i_id[batch_olcnt], cwid,
                        ol_amount[batch_olcnt],
str24[j]);
              }
            }
            else {
              ol_o_id[batch_olcnt] = cid;
              ol_d_id[batch_olcnt] = cdid;
              ol_w_id[batch_olcnt] = cwid;
              ol_number[batch_olcnt] = j + 1;
              ol_supply_w_id[batch_olcnt] = cwid;
              strncpy (ol_dist_info[batch_olcnt],
str24[j], 24);
            }
          }
          if (gen) {
            fflush (olfp);
          }
        }

        o_cnt =  ORDEARR;
        ol_cnt =  batch_olcnt;

        for (j = 0; j < batch_olcnt; j++) {
          ol_o_id_len[j] = sizeof(int);
          ol_d_id_len[j] = sizeof(int);
          ol_w_id_len[j] = sizeof(int);
          ol_number_len[j] = sizeof(int);
          ol_i_id_len[j] = sizeof(int);
          ol_supply_w_id_len[j] = sizeof(int);
          ol_dist_info_len[j] = 24;
          ol_amount_len[j] = sizeof(int);
        }
        for (j = batch_olcnt; j < 15*ORDEARR; j++)
{
          ol_o_id_len[j] = 0;
          ol_d_id_len[j] = 0;
          ol_w_id_len[j] = 0;
          ol_number_len[j] = 0;
          ol_i_id_len[j] = 0;
          ol_supply_w_id_len[j] = 0;
          ol_dist_info_len[j] = 0;
          ol_amount_len[j] = 0;
        }

        o_id_clen = ORDEARR;
        o_d_id_clen = ORDEARR;
        o_w_id_clen = ORDEARR;
        o_c_id_clen = ORDEARR;
        o_carrier_id_clen = ORDEARR;
        o_ol_cnt_clen = ORDEARR;

        ol_o_id_clen = batch_olcnt;
        ol_d_id_clen = batch_olcnt;
        ol_w_id_clen = batch_olcnt;
        ol_number_clen = batch_olcnt;
        ol_i_id_clen = batch_olcnt;
        ol_supply_w_id_clen = batch_olcnt;
        ol_dist_info_clen = batch_olcnt;
        ol_amount_clen = batch_olcnt;

        OCIERROR(errhp,
OCIStmtExecute(tpcsvc, curo1, errhp, (ub4) 1,
(ub4) 0,
                        (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                        (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS ));

        if ((++loopcount) % 50) {
          fprintf (stderr, ".");
        } else {
          fprintf (stderr, " %d orders committed\n
", row);
        }
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d orders
loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
              nrows, end_time - begin_time, end_cpu
- begin_cpu);
    }

/*-----------------------------------------------------------------+
 | Load the NEW-ORDER table.                       |
 +-----------------------------------------------------------------
*/

    if (do_A || do_n) {
      nrows = (eware - bware + 1) * NEWOFAC *
DISTFAC;

      fprintf (stderr, "Loading/generating new-
order: w%d - w%d (%d rows)\n   ",
              bware, eware, nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      cid = 0;
      cdid = 1;
      cwid = bware;
      loopcount = 0;

      for (row = 0; row < nrows; ) {
        for (i = 0; i < NEWOARR; i++, row++) {
          cid++;
          if (cid > NEWOFAC) {
            cid = 1;
            cdid++;
            if (cdid > DISTFAC) {
              cdid = 1;
              cwid++;
            }
          }

          if (gen) {
            printf ("%d %d %d\n", cid + 2100, cdid,
cwid);
          }
          else {
            no_o_id[i] = cid + 2100;
            no_d_id[i] = cdid;
            no_w_id[i] = cwid;
          }
        }

        if (gen) {
          fflush (stdout);
        }
        else {

          status = OCIStmtExecute(tpcsvc, curno,
errhp, (ub4) NEWOARR, (ub4) 0,
                        (CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
                        (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
          if (status != OCI_SUCCESS) {
        fprintf (stderr, "Aborted at w_id %d, d_id %d,
o_id %d\n", cwid, cdid, cid + 2100);
            OCIERROR(errhp, status);
            quit ();
            exit (1);
          }
        }

        if ((++loopcount) % 45)
          fprintf (stderr, ".");
        else
          fprintf (stderr, " %d rows committed\n   ",
row);
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows
loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
              nrows, end_time - begin_time, end_cpu
- begin_cpu);
    }

/*-----------------------------------------------------------------+
 | clean up and exit.                      |
 +-----------------------------------------------------------------
*/

    if (olfp)
      fclose (olfp);
    if (!gen)
      quit ();
    exit (0);

}

void initperm ()
{
    int i;
    int pos;
    int temp;

    /* init randperm3000 */

    for (i = 0; i < 3000; i++)
      randperm3000[i] = i + 1;
    for (i = 3000; i > 0; i--) {
      pos = lrand48 () % i;
      temp = randperm3000[i - 1];
      randperm3000[i - 1] = randperm3000[pos];
      randperm3000[pos] = temp;
    }
}

void randstr (str, x, y)
char *str;
int x;
int y;
{
    int i, j;
    int len;

    len = (lrand48 () % (y - x + 1)) + x;
    for (i = 0; i < len; i++) {
      j = lrand48 () % 62;
      if (j < 26)
        str[i] = (char) (j + 'a');
```

```
    else if (j < 52)
      str[i] = (char) (j - 26 + 'A');
    else
      str[i] = (char) (j - 52 + '0');
  }
  str[len] = '\0';

}

void randdatastr (str, x, y)
char *str;
int x;
int y;
{
  int i, j;
  int len;
  int pos;

  len = (lrand48 () % (y - x + 1)) + x;
  for (i = 0; i < len; i++) {
    j = lrand48 () % 62;
    if (j < 26)
      str[i] = (char) (j + 'a');
    else if (j < 52)
      str[i] = (char) (j - 26 + 'A');
    else
      str[i] = (char) (j - 52 + '0');
  }
  str[len] = '\0';
  if ((lrand48 () % 10) == 0) {
    pos = (lrand48 () % (len - 8));
    str[pos] = 'O';
    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 (RECOVERR);
}



::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::
 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;
```

Appendix F: 60 Day Space Calculation

TPM	1,354,086.50								
Warehouses	108,480								
SEGMENT	**TYPE**	**TSPACE**	**BLOCKS**	**BLOCK_SIZE**	**KB**	**FIVE_PCT (KB)**	**DAILY_GROW (KB)**	**TOTAL(KB)**	**TOTAL(Block)**
CUSTCLUSTER	CLUSTER	CUST	1,495,105,590	2,048	2,990,211,180	149,510,560	0	3,139,721,740	1,569,860,870
DB_STAT	SYS	SYSTEM	1,048,576	2,048	2,097,152	0		2,097,152	1,048,576
DCLUSTER	CLUSTER	DIST	1,199,424	2,048	2,398,848	119,942	0	2,518,790	1,259,395
HIST	TABLE	HIST	55,402,767	4,096	221,611,068	0	44,259,668	265,870,736	66,467,684
ICUST1	INDEX	ICUST1	4,627,450	16,384	74,039,200	3,701,968	0	77,741,168	4,858,823
ICUST2	INDEX	ICUST2	10,116,608	16,384	161,865,728	8,093,280	0	169,959,008	10,622,438
IDIST	INDEX	IDIST	280,512	2,048	561,024	28,052	0	589,076	294,538
IITEM	INDEX	ITEMS	5,632	2,048	11,264	564	0	11,828	5,914
IORDR2	INDEX	IORD2	7,256,039	16,384	116,096,624	5,804,832	0	121,901,456	7,618,841
ISTOK	INDEX	ISTK	13,941,542	16,384	223,064,672	11,153,232	0	234,217,904	14,638,619
ITEMCLUSTER	CLUSTER	ITEMS	8,446	2,048	16,892	844	0	17,736	8,868
IWARE	INDEX	WARE	70,512	2,048	141,024	7,052	0	148,076	74,038
NORDCLUSTER_QUEUE	CLUSTER	NORD	13,110,570	2,048	26,221,140	1,311,058	0	27,532,198	13,766,099
ORDRCLUSTER_QUEUE	CLUSTER	ORDR	172,764,416	16,384	2,764,230,656	0	552,065,984	3,316,296,640	207,268,540
STOCKCLUSTER	CLUSTER	STOK	1,601,590,636	2,048	3,203,181,272	160,159,064	0	3,363,340,336	1,681,670,168
SYS_IQ0000009617$$	INDEX	SYSTEM	684,548	2,048	1,369,096	68,454	0	1,437,550	718,775
SYS_IQ0000009622$$	INDEX	SYSTEM	490,314	16,384	7,845,024	392,256	0	8,237,280	514,830
SYSAUX	SYS	SYSTEM	61,440	2,048	122,880	0	0	122,880	61,440
SYSTEM	SYS	SYSTEM	204,800	2,048	409,600	0	0	409,600	204,800
WCLUSTER	CLUSTER	WARE	120,352	2,048	240,704	12,036	0	252,740	126,370
Total			3,378,090,174		9,795,735,048	340,363,194	596,325,652	10,732,423,894	3,581,089,626
Dynamic space(KB)			2,985,841,724						
Static space(KB)			7,150,256,518						
Free space(KB)			596,325,652						
Daily growth(KB)			596,325,652						
Daily spread		0	Oracle may be configured such that daily spread is 0						
60-day (GB)			40,941.04						
Log KB/tpmC		5.07	KB of log used per New-Order						
8-hour log (GB)			3,142.65						
		Capacity	**NumDisk**				**Total GB**		
Database Disks	ETERNUS8000	67.99	1,920				130,540.80		
							130,540.80		
		Capacity	**NumDisk(RAID0+1)**						
8-Hr Log Disks	ETERNUS8000	67.99	192				13,054.08		
							13,054.08		

Appendix G: Numerical Quantities Summary per Client

	Priced	Substituted										
	RX200 S4	F250								RX200 S2		
Client	cl069	cl039	cl040	cl041	cl042	cl043	cl044	cl045	cl046	cl061	cl062	cl063
tpmC	28581.49	27365.29	27394.31	27276.90	27341.89	27821.05	27909.68	27620.95	27642.09	28268.96	28250.43	28123.27
user	22600	22000	22000	22000	22000	22000	22000	22000	22000	22600	22600	22600
Menu												
average response	0.103	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.103	0.103	0.103
90%ile response	0.103	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104
New Order												
average response	0.207	1.196	1.182	1.273	1.227	0.832	0.738	0.982	0.966	0.446	0.460	0.587
90%ile response	0.309	1.416	1.374	1.489	1.418	1.063	0.967	1.175	1.170	0.625	0.646	0.770
average think time	12.023	12.017	12.009	12.023	12.019	12.007	12.021	12.005	12.010	12.014	12.020	12.009
average keying time	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012
mixture %	44.92	44.93	44.96	44.94	44.94	44.94	44.94	44.94	44.93	44.90	44.90	44.97
Payment												
average response	0.199	1.187	1.173	1.264	1.219	0.823	0.729	0.973	0.958	0.438	0.451	0.578
90%ile response	0.300	1.407	1.365	1.479	1.408	1.054	0.958	1.165	1.161	0.616	0.637	0.761
average think time	12.017	12.018	12.025	12.016	12.003	12.007	12.030	12.026	12.022	12.021	12.017	12.018
average keying time	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012
mixture %	43.03	43.02	42.99	43.00	42.99	43.01	43.01	43.03	43.00	43.03	43.04	42.97
Order Status												
average response	0.206	1.196	1.182	1.274	1.228	0.831	0.737	0.982	0.966	0.446	0.460	0.587
90%ile response	0.308	1.417	1.374	1.490	1.419	1.062	0.967	1.175	1.170	0.624	0.646	0.770
average think time	10.019	10.029	9.997	10.029	10.010	10.007	10.016	10.023	10.022	10.044	10.011	10.014
average keying time	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
mixture %	4.02	4.02	4.03	4.03	4.03	4.01	4.02	4.00	4.03	4.02	4.02	4.01
Delivery												
average response	0.103	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.103	0.103	0.103
90%ile response	0.103	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.103	0.103	0.103
average think time	5.016	5.025	5.007	5.015	5.007	5.020	5.032	5.015	5.014	5.029	5.015	5.030
average keying time	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
mixture %	4.01	4.01	4.01	4.02	4.02	4.01	4.01	4.02	4.03	4.03	4.01	4.03
Stock Level												
average response	0.191	1.179	1.166	1.257	1.211	0.816	0.722	0.966	0.950	0.430	0.444	0.571
90%ile response	0.293	1.399	1.357	1.472	1.401	1.046	0.950	1.158	1.154	0.608	0.628	0.754
average think time	5.016	5.025	4.998	5.014	5.001	5.016	5.023	5.016	5.000	5.017	5.008	5.016
average keying time	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
mixture %	4.02	4.02	4.01	4.01	4.02	4.03	4.02	4.01	4.01	4.02	4.03	4.02
# of New Order	3429779	3283835	3287318	3273228	3281027	3338526	3349162	3314514	3317051	3392276	3390052	3374793

	Substituted											
	RX200 S2											
Client	cl064	cl065	cl066	cl067	cl105	cl106	cl108	cl109	cl110	cl111	cl112	cl113
tpmC	28062.78	28572.52	28564.04	28442.60	28239.15	28223.45	28062.20	28160.05	28534.61	28572.10	28455.74	28421.75
user	22600	22600	22600	22600	22600	22600	22600	22600	22600	22600	22600	22600
Menu												
average response	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.104	0.104	0.104
90%ile response	0.104	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.104	0.104	0.104
New Order												
average response	0.621	0.229	0.238	0.325	0.480	0.496	0.625	0.553	0.252	0.227	0.314	0.336
90%ile response	0.808	0.349	0.365	0.488	0.669	0.678	0.810	0.735	0.394	0.345	0.479	0.506
average think time	12.010	12.009	12.002	12.012	12.018	12.021	12.016	12.017	12.008	12.014	12.025	12.013
average keying time	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012
mixture %	44.93	44.94	44.94	44.94	44.94	44.94	44.93	44.94	44.92	44.93	44.95	44.94
Payment												
average response	0.612	0.220	0.230	0.316	0.471	0.488	0.616	0.544	0.243	0.218	0.305	0.327
90%ile response	0.798	0.340	0.356	0.479	0.660	0.668	0.801	0.726	0.385	0.335	0.470	0.497
average think time	12.013	12.006	12.009	12.017	12.017	12.017	12.004	12.005	12.006	12.008	12.011	12.020
average keying time	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012
mixture %	43.04	43.02	43.00	43.03	43.03	43.02	43.04	43.02	43.03	42.99	42.99	43.01
Order Status												
average response	0.621	0.228	0.237	0.324	0.479	0.496	0.626	0.553	0.251	0.225	0.313	0.335
90%ile response	0.807	0.348	0.364	0.487	0.668	0.677	0.811	0.735	0.393	0.344	0.478	0.506
average think time	10.038	10.043	10.031	10.008	10.013	9.998	10.030	9.999	10.023	9.981	10.015	10.021
average keying time	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
mixture %	4.02	4.02	4.03	4.00	4.01	4.01	4.01	4.01	4.02	4.03	4.02	4.02
Delivery												
average response	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
90%ile response	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.104	0.104	0.104
average think time	5.008	5.025	5.014	4.997	5.005	5.012	5.027	5.026	5.023	5.012	5.013	5.019
average keying time	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
mixture %	4.01	4.02	4.02	4.01	4.02	4.02	4.02	4.01	4.01	4.03	4.02	4.01
Stock Level												
average response	0.605	0.213	0.222	0.309	0.463	0.480	0.609	0.537	0.235	0.210	0.298	0.320
90%ile response	0.791	0.332	0.349	0.471	0.653	0.661	0.794	0.718	0.377	0.327	0.463	0.490
average think time	5.025	5.028	5.022	5.042	5.013	5.005	5.003	5.020	5.017	5.018	5.016	5.034
average keying time	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
mixture %	4.00	4.00	4.01	4.02	4.00	4.01	4.00	4.02	4.02	4.02	4.02	4.02
# of New Order	3367534	3428703	3427685	3413113	3388698	3386815	3367464	3379206	3424154	3428653	3414689	3410611

	Substituted RX200 S2											
Client	cl114	cl115	cl116	cl117	cl118	cl119	cl121	cl122	cl124	cl125	cl126	cl127
tpmC	28309.30	28288.86	28055.97	28189.80	28505.27	28560.73	28420.86	28417.04	28263.77	28229.57	28040.30	28171.20
user	22600	22600	22600	22600	22600	22600	22600	22600	22600	22600	22600	22600
Menu												
average response	0.104	0.104	0.104	0.104	0.104	0.103	0.104	0.104	0.103	0.104	0.104	0.104
90%ile response	0.104	0.104	0.104	0.104	0.104	0.103	0.104	0.104	0.104	0.104	0.104	0.104
New Order												
average response	0.424	0.450	0.626	0.520	0.250	0.231	0.351	0.343	0.462	0.490	0.629	0.543
90%ile response	0.599	0.629	0.799	0.695	0.386	0.354	0.528	0.510	0.634	0.657	0.809	0.720
average think time	12.012	12.014	12.021	12.028	12.024	12.021	12.019	12.018	12.022	12.017	12.008	12.009
average keying time	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012
mixture %	44.94	44.96	44.93	44.94	44.89	44.96	44.95	44.94	44.95	44.95	44.91	44.95
Payment												
average response	0.414	0.441	0.616	0.511	0.241	0.222	0.342	0.334	0.453	0.481	0.620	0.534
90%ile response	0.590	0.619	0.789	0.686	0.376	0.344	0.518	0.501	0.625	0.647	0.800	0.710
average think time	12.027	12.019	12.016	12.015	12.017	12.018	11.998	12.015	12.017	12.028	12.026	12.021
average keying time	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012
mixture %	43.02	43.01	43.03	43.00	43.07	43.01	43.01	43.02	43.03	43.01	43.05	43.02
Order Status												
average response	0.423	0.449	0.626	0.519	0.249	0.230	0.349	0.342	0.462	0.489	0.629	0.542
90%ile response	0.599	0.628	0.800	0.695	0.385	0.353	0.526	0.509	0.635	0.656	0.810	0.720
average think time	10.032	10.043	9.986	10.016	10.020	10.021	10.011	9.978	10.031	10.026	9.982	10.000
average keying time	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
mixture %	4.01	4.00	4.01	4.02	4.01	4.01	4.00	4.02	4.00	4.01	4.02	4.01
Delivery												
average response	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
90%ile response	0.104	0.104	0.104	0.104	0.104	0.103	0.104	0.104	0.103	0.104	0.104	0.104
average think time	5.003	5.018	5.013	5.020	5.036	5.000	5.011	5.037	5.017	5.004	5.021	5.019
average keying time	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
mixture %	4.01	4.00	4.01	4.03	4.02	4.00	4.02	4.00	4.00	4.01	4.01	4.01
Stock Level												
average response	0.407	0.433	0.609	0.503	0.234	0.214	0.334	0.327	0.446	0.473	0.613	0.526
90%ile response	0.582	0.611	0.782	0.678	0.368	0.336	0.510	0.493	0.618	0.640	0.793	0.703
average think time	5.017	5.017	5.015	5.007	5.019	5.030	5.019	5.033	5.017	5.026	5.026	5.029
average keying time	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
mixture %	4.02	4.03	4.02	4.01	4.01	4.02	4.02	4.02	4.02	4.02	4.01	4.01
# of New Order	3397116	3394664	3366717	3382776	3420633	3427288	3410504	3410045	3391653	3387549	3364836	3380545

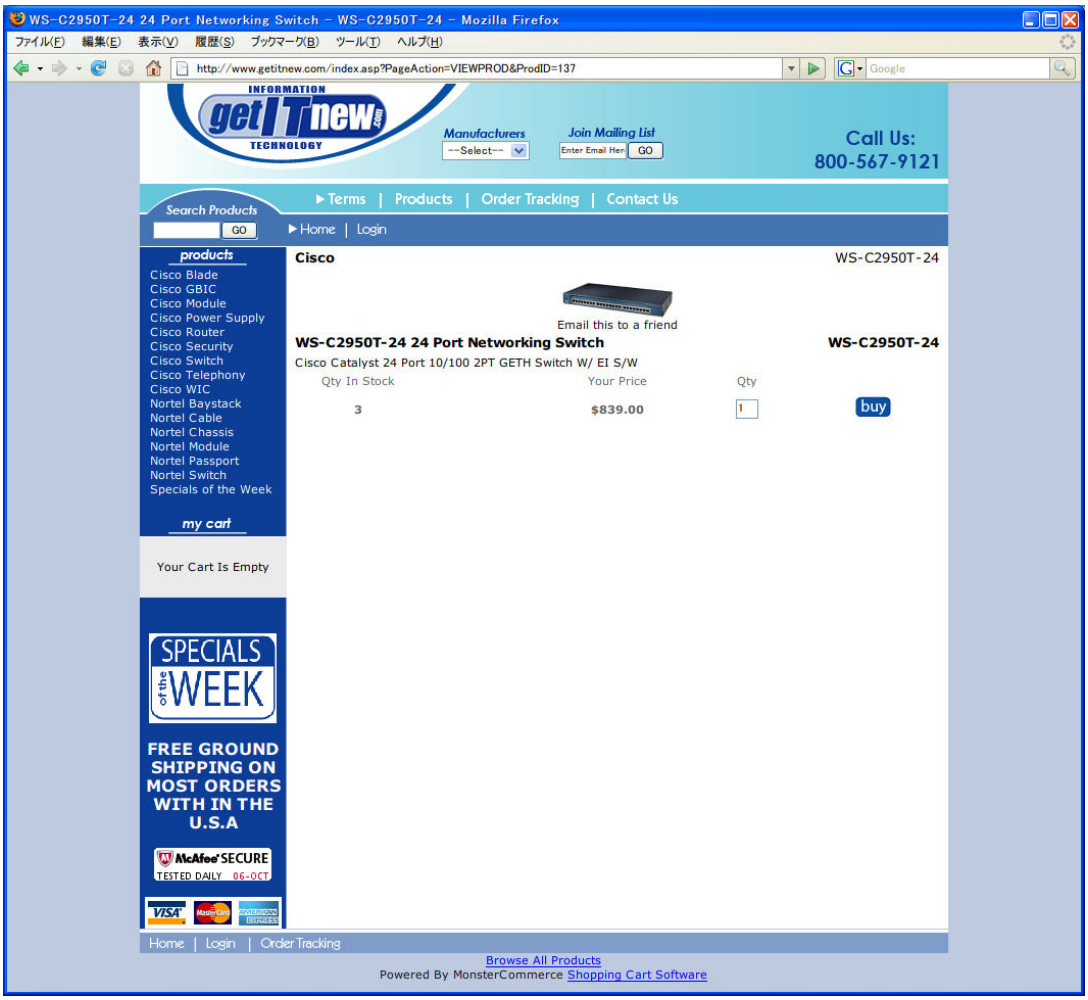
	Substituted											
	RX200 S2											
Client	cl128	cl129	cl130	cl131	cl132	cl133	cl134	cl135	cl136	cl137	cl138	cl139
tpmC	28558.21	28549.30	28441.24	28406.69	28214.40	28270.70	28133.55	28190.69	28565.01	28568.66	28431.88	28396.02
user	22600	22600	22600	22600	22600	22600	22600	22600	22600	22600	22600	22600
Menu												
average response	0.104	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
90%ile response	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.103
New Order												
average response	0.246	0.235	0.334	0.342	0.508	0.464	0.578	0.525	0.245	0.237	0.331	0.373
90%ile response	0.379	0.366	0.498	0.513	0.694	0.635	0.746	0.698	0.386	0.377	0.493	0.556
average think time	12.025	12.017	12.014	12.011	12.018	12.017	12.014	12.014	12.021	12.009	12.013	12.017
average keying time	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012
mixture %	44.99	44.92	44.97	44.90	44.95	44.97	44.96	44.94	44.98	44.99	44.94	44.95
Payment												
average response	0.237	0.226	0.325	0.333	0.499	0.455	0.569	0.516	0.236	0.228	0.323	0.364
90%ile response	0.370	0.357	0.489	0.503	0.684	0.626	0.737	0.688	0.377	0.367	0.484	0.547
average think time	12.006	12.014	12.023	12.016	12.014	12.024	12.006	12.015	11.998	12.024	12.016	12.003
average keying time	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012
mixture %	42.99	43.03	43.00	43.06	42.99	42.99	43.02	43.02	42.99	42.98	43.02	42.99
Order Status												
average response	0.246	0.234	0.333	0.341	0.508	0.464	0.578	0.525	0.244	0.236	0.330	0.372
90%ile response	0.378	0.365	0.497	0.511	0.694	0.635	0.746	0.698	0.386	0.375	0.491	0.556
average think time	10.039	10.038	10.002	9.996	10.024	10.029	10.036	9.994	10.023	10.048	10.003	9.982
average keying time	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
mixture %	4.00	4.02	4.01	4.00	4.02	4.01	4.00	4.02	4.01	4.01	4.02	4.02
Delivery												
average response	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
90%ile response	0.104	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
average think time	5.014	5.022	5.012	5.022	5.027	5.009	5.007	5.015	5.030	5.013	5.029	5.016
average keying time	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
mixture %	4.02	4.02	4.01	4.02	4.03	4.02	4.02	4.01	4.00	4.01	4.01	4.03
Stock Level												
average response	0.230	0.219	0.318	0.326	0.492	0.449	0.563	0.508	0.229	0.221	0.315	0.357
90%ile response	0.363	0.349	0.481	0.495	0.677	0.619	0.730	0.681	0.370	0.360	0.476	0.541
average think time	5.022	5.006	5.025	5.011	5.023	5.021	5.015	5.013	5.023	5.015	5.015	5.029
average keying time	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
mixture %	4.00	4.01	4.01	4.02	4.01	4.01	4.00	4.01	4.02	4.01	4.01	4.01
# of New Order	3426986	3425917	3412949	3408803	3385729	3392484	3376026	3382883	3427802	3428240	3411826	3407523

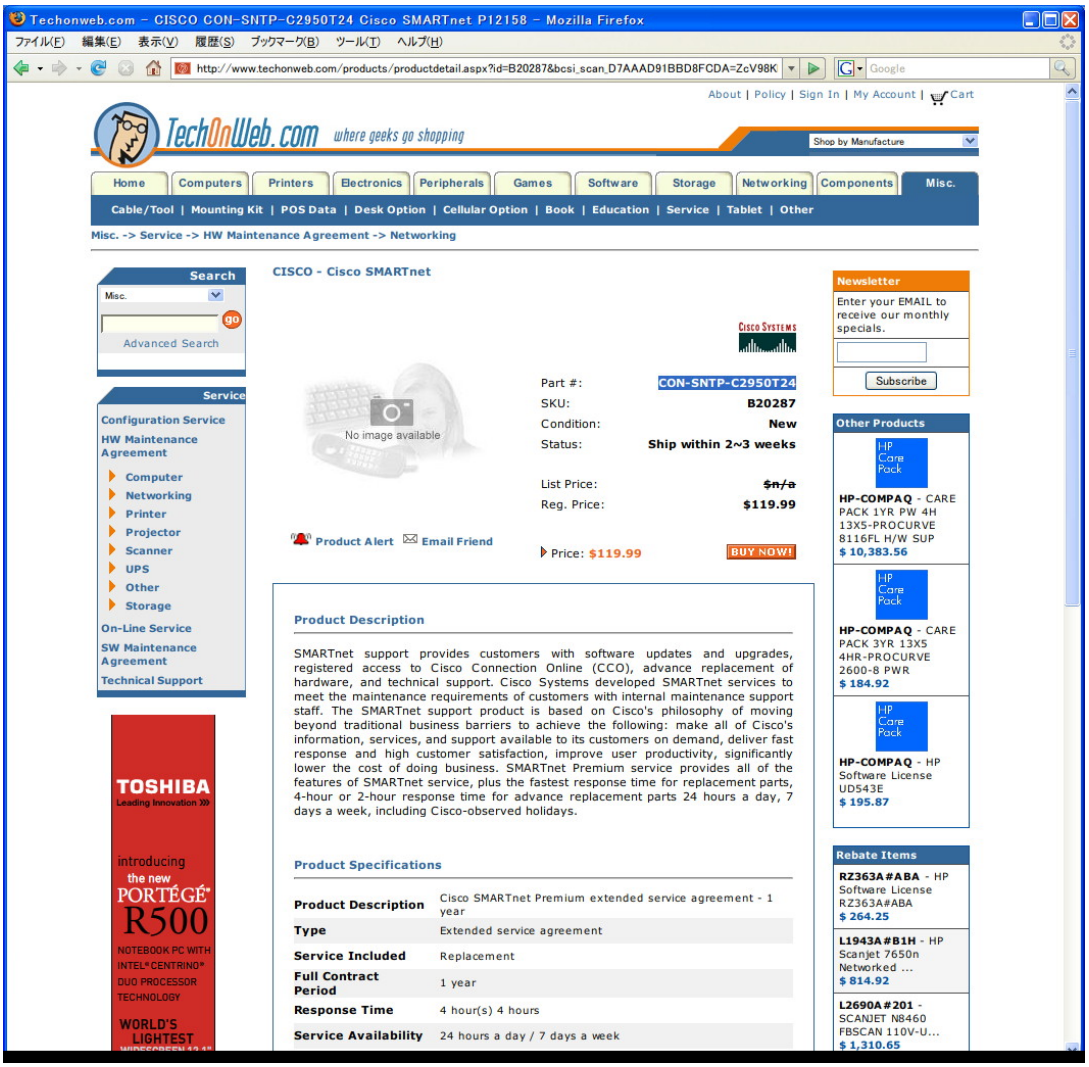
Appendix H: Price Quotes

From: MaryBeth Pierantoni [mailto:mary.beth.pierantoni@oracle.com]
Sent: Wednesday, November 19, 2008 9:42 AM
To: Shin'ichi Kurogi
Subject: Oracle Pricing

Product	Price	Quantity	Extended Price
Oracle Database 10g Enterprise Edition, Per Processor, Unlimited Users for 3 years	$23,750	16*	$380,000
Oracle Database Server Support for 3 years	$2,300	3	$6,900
Tuxedo CFS-R Tier 1	$1,800	48	$86,400
Oracle Premium Support for 3 years	$19,008	3	$57,024
Oracle Mandatory E-Business Discount			<$106,065>
Oracle TOTAL			$424,259

(*16 = 0.50 * 32. 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 0.50).
Oracle pricing contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081


WS-C2950T-24 24 Port Networking Switch - WS-C2950T-24 - Mozilla Firefox
http://www.getitnew.com/index.asp?PageAction=VIEWPROD&ProdID=137
Manufacturers
Join Mailing List
Call Us:
800-567-9121
Terms | Products | Order Tracking | Contact Us
Search Products
Home | Login
products
Cisco Blade
Cisco GBIC
Cisco Module
Cisco Power Supply
Cisco Router
Cisco Security
Cisco Switch
Cisco Telephony
Cisco WIC
Nortel Baystack
Nortel Cable
Nortel Chassis
Nortel Module
Nortel Passport
Nortel Switch
Specials of the Week
my cart
Your Cart Is Empty
SPECIALS of the WEEK
FREE GROUND SHIPPING ON MOST ORDERS WITH IN THE U.S.A
McAfee SECURE
Cisco
WS-C2950T-24
Email this to a friend
WS-C2950T-24 24 Port Networking Switch
WS-C2950T-24
Cisco Catalyst 24 Port 10/100 2PT GETH Switch W/ EI S/W
Qty In Stock
Your Price
Qty
3
$839.00
buy
Home | Login | Order Tracking
Browse All Products
Powered By MonsterCommerce Shopping Cart Software


Techonweb.com – CISCO CON-SNTP-C2950T24 Cisco SMARTnet P12158 – Mozilla Firefox
http://www.techonweb.com/products/productdetail.aspx?id=B20287&bcsi_scan_D7AAAD91BBD8FCDA=ZcV98K
About | Policy | Sign In | My Account | Cart
TechOnWeb.com where geeks go shopping
Home | Computers | Printers | Electronics | Peripherals | Games | Software | Storage | Networking | Components | Misc.
Cable/Tool | Mounting Kit | POS Data | Desk Option | Cellular Option | Book | Education | Service | Tablet | Other
Misc. -> Service -> HW Maintenance Agreement -> Networking
Search
Advanced Search
Service
Configuration Service
HW Maintenance Agreement
Computer
Networking
Printer
Projector
Scanner
UPS
Other
Storage
On-Line Service
SW Maintenance Agreement
Technical Support
CISCO - Cisco SMARTnet
No image available
Part #: CON-SNTP-C2950T24
SKU: B20287
Condition: New
Status: Ship within 2~3 weeks
List Price: $n/a
Reg. Price: $119.99
Product Alert
Email Friend
Price: $119.99
BUY NOW!
Product Description
SMARTnet support provides customers with software updates and upgrades, registered access to Cisco Connection Online (CCO), advance replacement of hardware, and technical support. Cisco Systems developed SMARTnet services to meet the maintenance requirements of customers with internal maintenance support staff. The SMARTnet support product is based on Cisco's philosophy of moving beyond traditional business barriers to achieve the following: make all of Cisco's information, services, and support available to its customers on demand, deliver fast response and high customer satisfaction, improve user productivity, significantly lower the cost of doing business. SMARTnet Premium service provides all of the features of SMARTnet service, plus the fastest response time for replacement parts, 4-hour or 2-hour response time for advance replacement parts 24 hours a day, 7 days a week, including Cisco-observed holidays.
Product Specifications
Product Description: Cisco SMARTnet Premium extended service agreement - 1 year
Type: Extended service agreement
Service Included: Replacement
Full Contract Period: 1 year
Response Time: 4 hour(s) 4 hours
Service Availability: 24 hours a day / 7 days a week
Newsletter
Enter your EMAIL to receive our monthly specials.
Subscribe
Other Products
HP-COMPAQ - CARE PACK 1YR PW 4H 13X5-PROCURVE 8116FL H/W SUP $ 10,383.56
HP-COMPAQ - CARE PACK 3YR 13X5 4HR-PROCURVE 2600-8 PWR $ 184.92
HP-COMPAQ - HP Software License UD543E $ 195.87
Rebate Items
RZ363A#ABA - HP Software License RZ363A#ABA $ 264.25
L1943A#B1H - HP Scanjet 7650n Networked ... $ 814.92
L2690A#201 - SCANJET N8460 FBSCAN 110V-U... $ 1,310.65
TOSHIBA
Leading Innovation
introducing the new PORTÉGÉ R500
NOTEBOOK PC WITH INTEL® CENTRINO® DUO PROCESSOR TECHNOLOGY
WORLD'S LIGHTEST

Appendix I: Auditor's attestation letter

INFO SIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL CERTIFIED AUDITOR

Benchmark Sponsor: Shin'ichi Kurogi
Manager, TRIOLE Technology Development Division
Software Unit
Fujitsu Limited
Shin-Yokohama TECH Bldg.
3-9-18 Shin-Yokohama, Kohoku-ku, Yokohama
Kanagawa Pref. 222-0033, Japan

November 21, 2008

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

Platform: Fujitsu PRIMEQUEST 540A c/s
Operating system: Red Hat Enterprise Linux 4 AS
Database Manager: Oracle Database 10g R2 Enterprise Edition
Transaction Manager: Tuxedo CFS-R tier-1

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	**tpmC**
Server: Fujitsu PRIMEQUEST 540A				
16 x Itanium Dual-Core 9150M (1.66GHz)	1024 GB (24 MB L3)	2112 x 73 GB 15Krpm 1 x 73 GB 10Krpm int.	0.992 Seconds	**1,354,086.5**
Forty eight (48) Client: Fujitsu PRIMERGY RX200 S4 (each with)				
1 x Xeon Dual-Core X5260 (3.33 GHz)	3.0 GB (6 MB L2)	1 x 73 GB 15Krpm SAS	n/a	n/a

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

125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com

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

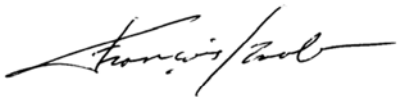
Additional Audit Notes:

The tested configuration included (1) priced ETERNUS8000 storage subsystem and (5) non-priced ETERNUS6000 storage subsystems. The priced configuration includes (6) 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 used a mix of PRIMERGY client systems, including one RX200 S4, thirty-nine (39) RX200 S2, and eight (8) F250. The RX200 S2 and the F250 clients were substituted one-for-one by the more recent PRIMERGY RX200 R4 in the priced configuration. Based on the specifications of these client systems, it is my opinion that this substitution has no significant effect on performance.

Respectfully Yours,

125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com



François Raab, President

125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com