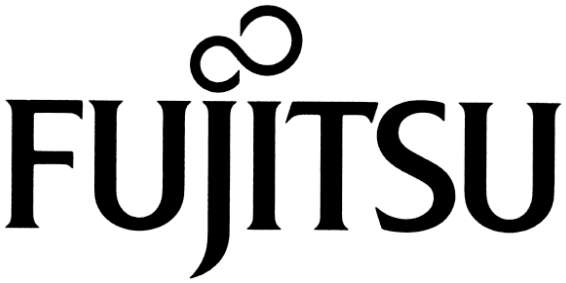


and

TPC Benchmark™C

Full Disclosure Report

Fujitsu
PRIMEQUEST 480 c/s
W/ 39 Front-Ends

running

Oracle Database 10g
Enterprise Edition

April 26,2006

First Edition - April 2006

The benchmark results contained in this document were submitted for compliance with version 5.6 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 April 2006.

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, April 2006.

ORACLE FUJITSU

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 5.6.

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.

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C test conducted by Fujitsu Ltd. on the Fujitsu PRIMEQUEST 480 w/ 39 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 Enterprise Edition.
The operating system on the clients was Red Hat Enterprise Linux 4 ES for x86.
Those clients ran Apache HTTP Server and BEA Tuxedo 8.1 CFS-R.
Two standard metrics, transaction-per-minute-C(tpmC) and price per tpmC($/tpmC) are reported, in accordance with the TPC Benchmark C Standard. The independent auditor's report by Francois Raab appears at the end of this report.

TPC Benchmark C Metrics

The standard TPC Benchmark C metrics, tpmC (transactions per minute), price per tpmC (three year capital cost per measured tpmC), and the availability date are reported as:

792,101.96 tpmC
$8.92 USD/tpmC
October 26, 2006

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.

FUJITSU ORACLE	**PRIMEQUEST 480 c/s w/39 Front-Ends**		**TPC-C Rev 5.6** **Report Date: April 26, 2006**
Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$7,060,571 USD	**792,101.96 tpmC**	**$8.92 USD/tpmC**	**October 26,2006**

Database Server Processors/Cores/Threads	Database Manager	Operating system	Other Software	Number of users
32/32/32 Intel Itanium 2 1.6GHz for Server	**Oracle Database 10g Enterprise Edition**	**Red Hat Enterprise Linux 4 AS**	**BEA Tuxedo 8.1**	**630,240**

System Component	Qty	Server:	Qty	Each of 39 Clients:
Processors/Cores/ Threads	32/32/ 32	Intel Itanium2 1.6GHz	2/2/ 4	Intel Xeon 3GHz
Cache Memory		9MB L3 Cache		2MB L2 Cache
Memory	64	16GB (4 x 4GB DDR2-400)	3	1GB (2 x 512MB PC-3200)
Disk Controllers	32	2G bps Fiber Chennel	1	SCSI controller
Disk Drives	1 1536 56	73GB 10K rpm 36GB 15K rpm 73GB 15K rpm	1	73 GB 10K rpm
Total Storage		59,457 GB		2,847 GB

FUJITSU ORACLE	PRIMEQUEST 480 c/s w/39 Front-Ends	TPC-C Rev 5.6 Report Date: April 26, 2006

Server Hardware	Part Number	Qty	Source	Unit Price	Ext. Price	3 Yr. Maint.
PRIMEQUEST 480 Base Unit	MC5A0P111U	1	1	65,870.00	65,870.00	109,632.00
System Board	MC-77SB11	8	1	17,500.00	140,000.00	
CPU Module (Itanium 2 1.6GHz/9MB L3)	MC-01AA11	32	1	13,680.00	437,760.00	347,136.00
16GB Memory Module (4x4GB DDR2-400)	MC-02A511	64	1	42,900.00	2,745,600.00	195,784.31
I/O Unit	MC-77UX11	8	1	14,240.00	113,920.00	
BMC Module	MC-77BM11	1	1	1,720.00	1,720.00	
Disk Drive Unit (3.5inch, 73GB, 10,000rpm, Ultra320)	MC-03D321	1	1	2,360.00	2,360.00	
Gigabit Switch Board (w/ 8 external 1000Base-T ports)	MC-77GE11	1	1	21,330.00	21,330.00	
Additional Power Supply	MC-57PS11U	1	1	17,100.00	17,100.00	
FibreChannel Card (2Gbps, PCI-X)	MC-08FC11	32	1	2,660.00	85,120.00	
FibreChannel Cable (15m, LC-LC)	CBL-MLLB15	32	1	330.00	10,560.00	
LCD (15inch)	MC-07FL21	1	1	4,830.00	4,830.00	
USB Keyboard	MC-07KB11	1	1	480.00	480.00	
USB Mouse	MC-07MU11	1	1	80.00	80.00	
				Server Hardware Subtotals	3,646,730.00	652,552.31
Storage						
ETERNUS6000 Model900 Base Unit w/ 2 Controllers, 8 Drive Enclosures, 4 Device Adapters	E690S01AU	4	1	195,000.00	780,000.00	104,448.00
Additional Expansion Raack	E600CR3U	8	1	8,000.00	64,000.00	8,640.00
Additional Controller w/ Power Supply	E600CJ3U	4	1	16,000.00	64,000.00	8,592.00
Additional Controller	E600CJ4U	4	1	16,000.00	64,000.00	8,592.00
Cashe Memory (4x2GB)	E600CM41	4	1	108,600.00	434,400.00	58,176.00
FibreChannel Host Interface (2Gbps, 2 x dual port)	E600CH14	16	1	12,800.00	204,800.00	27,648.00
Drive Enclosure Set (4 units) w/ 2 Device Adapters)	E690SE22U	8	1	39,400.00	315,200.00	42,144.00
Drive Enclosure Set (4 units)	E600CE21U	16	1	31,000.00	496,000.00	66,432.00
Disk Drive Unit (36GB, 15,000rpm)	E600CA2	1536	1	1,000.00	1,536,000.00	350,208.00
ETERNUS3000 Model700 Base Unit w/ 2 Controllers,4x2GB Cashe Memory,2 DriveEnclosures, 2 FibreChannel Host Interface (2 x dual port)	E370S21BUL	1	1	71,100.00	71,100.00	13,608.00
Disk Drive Unit (73GB, 15,000rpm)	E300CA7HL	56	1	1,350.00	75,600.00	18,144.00
				Strage Subtotals	4,105,100.00	706,632.00
Server Software						
Red Hat Enterprise Linux 4 AS (for Intel Itanium)		1	1	7,497.00	7,497.00	
Oracle Database 10g Enterprise Edition, Unlimited Users for 3 years		32	2	20,000.00	640,000.00	
Oracle Database Server Support Package for 3 years		3	2			6,000.00
				Server Software Subtotals	647,497.00	6,000.00
Client Hardware						
PRIMERGY RX200 S2 (Xeon 3GHz,1GB mem,2x1000Base-T)	S26361-K942-V211	39	1	2,017.20	78,670.80	37,800.00
CPU Module (Xeon 3GHz)	S26361-F3099-E400	39	1	523.20	20,404.80	
1GB Memory Module (2 x 512MB PC2-3200)	S26361-F3072-E521	78	1	295.20	23,025.60	
Disk Drive Unit (3..5inch, 73GB, 10,000rpm, Ultra320)	S26361-F3121-E173	39	1	279.60	10,904.40	
Internal CD-ROM Unit	SNP:SY-F2313E1-P	39	1	57.60	2,246.40	
19inch Rack (24U)	S26361-K826-V102	2	1	1,843.20	3,686.40	
LCD/KB/Pointing Device Unit (1U)	S26361-K1023-V200	2	1	1,874.40	3,748.80	
KVM Switch (8ports, 1U)	S26361-F2293-E801	6	1	750.00	4,500.00	
KVM Cable (1.8m)	S26361-F2293-L20	44	1	10.80	475.20	
				Client Hardware Subtotals	147,662.40	37,800.00
Client Software						
Red Hat Enterprise Linux 4 ES (for x86)	S26361-F2346-E212	39	1	1,366.00	53,274.00	
BEA TUXEDO 8.1 CFS-R (for RHEL4 x86)		39	3	1,140.00	44,460.00	29,484.00
				Client Software Subtotals	97,734.00	29,484.00
User Connectivity						
Cisco Catalyst 2950T-24 Switch		3	4	1,295.00	3,885.00	
Cisco SMARTnet 24x7x4 Maintenance		3	4			1,080.00
				User Connectivity Subtotals	3,885.00	1,080.00
Oracle Mandatory E-Business Discount			2		(129,200.00)	
Large Configuration Discount and Support Prepayment*			1		(2,616,769.96)	(275,616.86)
				Total	5,902,639.00	1,157,932.00

Pricing Sources: 1 = Fujitsu , 2 = Oracle , 3 = BEA , 4 = costcentral.com Audited by: Francois Raab, InfoSizing, Inc. (www.sizing.com) Oracle pricing contact: MaryBeth Pierantoni (mary.beth.pierantoni@oracle.com, 916-315-5081)	**Three-Year Cost of Ownership USD** **tpmC** **$ USD / tpmC**	**$7,060,571** **792,101.96** **$8.92**

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

PRIMEQUEST 480 c/s w/ 39 Front-Ends

Oracle Database 10g Enterprise Edition

MQTH, Computed Maximum Qualified Throughput	792,101.96 tpmC		
Response Times (in seconds)	**Average**	**90%**	**Max.**
New-Order	0.353	0.710	3.113
Payment	0.346	0.703	3.114
Order-Status	0.352	0.709	2.999
Delivery (interactive portion)	0.103	0.103	0.324
Delivery (deferred portion)	0.250	0.607	2.892
Stock-Level	0.340	0.696	3.099
Menu	0.104	0.103	74.414

Transaction Mix, in percent of total transaction	
New-Order	44.94
Payment	43.02
Order-Status	4.01
Delivery(interactive)	4.01
Stock-Level	4.02

Emulation Delay (in seconds)	Resp.Time	Menu
New-Order	.1	.1
Payment	.1	.1
Order-Status	.1	.1
Delivery (interactive)	.1	.1
Stock-Level	.1	.1

Keying/Think Times (in seconds)	Min.		Average		Max.	
New-Order	18.003	0.000	18.012	12.014	18.045	120.196
Payment	3.004	0.000	3.012	12.015	3.058	120.196
Order-Status	2.007	0.000	2.012	10.017	2.035	100.196
Delivery (interactive)	2.007	0.000	2.012	5.020	2.048	50.170
Stock-Level	2.006	0.000	2.012	5.024	2.043	50.198

Test Duration	
Ramp-up time (seconds)	11,525
Measurement interval	7,200
Transactions during measurement interval	211,598,853

Checkpointing	
Number of checkpoints	4
Checkpoint interval	1,701

Numerical Quantities for each client

	New clients (PRIMERGY RX200 S2)									
Client	cl105	cl106	cl107	cl108	cl109	cl110	cl111	cl112	cl113	cl114
tpmC	20371.36	20420.91	20416.33	20403.46	20328.91	20424.45	20417.95	20406.41	20349.32	20404.33
menu-ave	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104
menu-ninety	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
new-ave	0.291	0.229	0.246	0.257	0.327	0.234	0.245	0.240	0.306	0.230
new-ninety	0.514	0.375	0.396	0.431	0.585	0.381	0.398	0.389	0.546	0.372
new-think	12.006	12.017	11.996	12.011	12.015	12.004	12.010	12.024	12.016	12.012
new-keying	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012
new-mix	44.930	44.930	44.970	44.960	44.930	44.960	44.940	44.920	44.930	44.910
pay-ave	0.284	0.223	0.239	0.250	0.319	0.228	0.238	0.233	0.299	0.224
pay-ninety	0.506	0.368	0.388	0.422	0.576	0.374	0.389	0.381	0.538	0.366
pay-think	12.006	12.008	12.034	12.021	12.006	12.022	12.000	12.011	12.014	12.036
pay-keying	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012
pay-mix	43.020	43.050	43.000	43.010	43.040	42.990	43.020	43.030	43.030	43.050
ord-ave	0.290	0.229	0.245	0.256	0.324	0.233	0.244	0.239	0.304	0.229
ord-ninety	0.513	0.374	0.394	0.430	0.583	0.381	0.396	0.388	0.544	0.372
ord-think	10.021	9.991	10.025	10.010	10.042	10.010	10.016	9.977	10.020	10.034
ord-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
ord-mix	4.020	4.000	4.000	4.010	4.010	4.020	4.010	4.010	4.020	4.010
del-ave	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
del-ninety	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
del-think	5.034	5.017	5.022	5.017	5.038	5.032	5.031	5.014	5.028	5.004
del-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
del-mix	4.010	4.010	4.010	4.010	4.020	4.020	4.020	4.020	4.000	4.010
stk-ave	0.278	0.217	0.233	0.243	0.313	0.221	0.232	0.227	0.292	0.218
stk-ninety	0.501	0.361	0.381	0.415	0.570	0.366	0.383	0.375	0.532	0.359
stk-think	5.011	5.029	5.036	5.025	5.033	5.019	5.041	5.027	5.019	5.036
stk-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
stk-mix	4.020	4.010	4.020	4.010	4.000	4.010	4.010	4.020	4.020	4.020
ntxn	5442687	5456613	5450328	5447792	5432897	5454880	5453809	5453706	5436960	5454748

Numerical Quantities for each client

	New clients (PRIMERGY RX200 S2)									
Client	cl115	cl116	cl119	cl120	cl121	cl122	cl123	cl124	cl125	cl126
tpmC	20400.59	20390.15	20401.91	20409.73	20331.85	20437.90	20439.38	20283.00	20374.53	20404.63
menu-ave	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.105	0.105	0.104
menu-ninety	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
new-ave	0.251	0.250	0.247	0.243	0.335	0.238	0.226	0.370	0.282	0.244
new-ninety	0.403	0.414	0.407	0.399	0.597	0.391	0.368	0.633	0.459	0.387
new-think	12.022	12.025	12.011	12.010	12.012	12.001	12.002	12.019	12.023	12.022
new-keying	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012
new-mix	44.950	44.900	44.910	44.920	44.960	44.990	44.970	44.920	44.940	44.940
pay-ave	0.244	0.243	0.239	0.236	0.327	0.232	0.219	0.363	0.275	0.238
pay-ninety	0.397	0.407	0.399	0.392	0.589	0.384	0.362	0.626	0.451	0.380
pay-think	12.016	12.016	12.017	12.011	12.018	12.007	12.016	12.017	12.005	12.014
pay-keying	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012
pay-mix	43.020	43.050	43.020	43.030	43.020	42.980	42.990	43.050	43.010	43.010
ord-ave	0.250	0.249	0.245	0.242	0.332	0.238	0.226	0.371	0.281	0.243
ord-ninety	0.402	0.414	0.405	0.399	0.594	0.392	0.368	0.635	0.457	0.385
ord-think	10.027	10.022	10.046	10.033	10.013	9.996	9.994	10.050	10.025	10.015
ord-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
ord-mix	4.020	4.010	4.030	4.020	4.000	4.020	4.000	4.010	4.010	4.020
del-ave	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
del-ninety	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
del-think	5.035	5.014	4.995	5.018	5.025	5.023	5.040	5.031	5.018	5.004
del-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
del-mix	4.020	4.020	4.030	4.020	4.000	4.000	4.010	4.000	4.010	4.020
stk-ave	0.238	0.237	0.233	0.229	0.322	0.226	0.213	0.357	0.269	0.232
stk-ninety	0.389	0.400	0.392	0.385	0.584	0.377	0.355	0.620	0.445	0.373
stk-think	5.025	5.036	5.011	5.014	5.030	5.029	5.026	5.002	5.013	5.031
stk-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
stk-mix	3.990	4.020	4.010	4.010	4.020	4.010	4.030	4.020	4.030	4.010
ntxn	5448725	5450724	5453169	5454244	5429142	5454316	5457021	5420880	5442872	5451637

	New clients (PRIMERGY RX200 S2)									
Client	cl127	cl128	cl129	cl130	cl131	cl132	cl133	cl134	cl135	cl136
tpmC	20412.28	20302.04	20399.79	20401.60	20426.08	20270.70	20367.03	20388.15	20430.72	20343.19
menu-ave	0.105	0.105	0.104	0.104	0.104	0.104	0.104	0.104	0.105	0.105
menu-ninety	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
new-ave	0.236	0.357	0.269	0.252	0.238	0.389	0.288	0.263	0.216	0.305
new-ninety	0.385	0.615	0.442	0.412	0.396	0.667	0.478	0.430	0.337	0.525
new-think	12.016	12.016	12.012	12.012	12.014	12.020	12.020	12.013	12.018	12.026
new-keying	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012
new-mix	44.930	44.950	44.960	44.930	44.970	44.930	44.950	44.920	44.930	44.930
pay-ave	0.229	0.350	0.262	0.245	0.231	0.382	0.281	0.256	0.209	0.298
pay-ninety	0.377	0.608	0.435	0.405	0.388	0.660	0.472	0.423	0.329	0.518
pay-think	12.016	12.023	12.012	12.009	12.015	12.018	12.016	12.012	12.011	12.018
pay-keying	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012
pay-mix	43.020	43.020	42.980	43.040	42.990	43.040	43.010	43.030	43.040	43.010
ord-ave	0.236	0.358	0.268	0.251	0.238	0.389	0.286	0.262	0.215	0.306
ord-ninety	0.385	0.615	0.440	0.412	0.396	0.665	0.477	0.430	0.338	0.527
ord-think	10.037	10.039	10.032	10.025	10.014	10.033	9.987	10.014	10.007	10.025
ord-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
ord-mix	4.020	4.010	4.010	4.010	4.020	4.010	4.000	4.030	4.010	4.030
del-ave	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
del-ninety	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103	0.103
del-think	5.019	5.009	4.998	5.038	5.014	5.035	5.029	5.013	5.014	4.998
del-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
del-mix	4.010	4.010	4.030	4.010	4.020	4.010	4.010	4.010	4.010	4.000
stk-ave	0.223	0.344	0.256	0.239	0.225	0.375	0.274	0.250	0.203	0.293
stk-ninety	0.370	0.601	0.429	0.399	0.382	0.653	0.464	0.416	0.323	0.513
stk-think	5.028	5.022	4.997	5.028	5.037	5.034	5.044	5.008	5.020	5.021
stk-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
stk-mix	4.020	4.010	4.020	4.010	4.000	4.010	4.030	4.010	4.010	4.030
ntxn	5454015	5422821	5446267	5451186	5451826	5414953	5440584	5449042	5459781	5436652

	New clients (PRIMERGY RX200 S2)		Old clients (PRIMERGY F250)						
Client	cl137	cl138	cl033	cl035	cl036	cl037	cl038	cl039	cl040
tpmC	20353.45	20416.35	19617.99	20110.54	20080.49	20153.90	19660.08	19989.96	20160.35
menu-ave	0.104	0.103	0.103	0.104	0.104	0.104	0.104	0.104	0.104
menu-ninety	0.103	0.103	0.103	0.104	0.104	0.104	0.104	0.104	0.104
new-ave	0.294	0.254	1.136	0.572	0.602	0.523	1.095	0.714	0.539
new-ninety	0.507	0.415	1.602	0.961	1.016	0.892	1.445	1.153	0.937
new-think	12.016	12.005	12.016	12.020	12.017	12.019	12.021	12.015	12.004
new-keying	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012	18.012
new-mix	44.920	44.960	44.940	44.960	44.940	44.930	44.960	44.950	44.970
pay-ave	0.287	0.247	1.129	0.564	0.595	0.516	1.088	0.706	0.532
pay-ninety	0.499	0.408	1.595	0.953	1.009	0.885	1.437	1.146	0.929
pay-think	12.022	12.006	12.005	12.025	12.017	12.005	12.013	12.019	12.010
pay-keying	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012	3.012
pay-mix	43.060	42.980	43.030	43.020	43.010	43.040	43.000	43.010	42.970
ord-ave	0.292	0.252	1.135	0.570	0.602	0.523	1.096	0.711	0.537
ord-ninety	0.504	0.414	1.602	0.960	1.016	0.893	1.444	1.150	0.935
ord-think	9.989	10.054	9.996	10.009	10.049	10.023	9.991	9.985	9.982
ord-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
ord-mix	4.010	4.020	4.020	4.000	4.010	4.010	4.010	4.000	4.020
del-ave	0.103	0.103	0.103	0.103	0.104	0.103	0.103	0.103	0.104
del-ninety	0.103	0.103	0.104	0.104	0.104	0.104	0.104	0.104	0.104
del-think	5.014	5.004	5.037	5.016	5.001	5.031	5.030	5.018	5.022
del-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
del-mix	4.010	4.020	4.010	4.010	4.010	4.010	4.000	4.010	4.020
stk-ave	0.281	0.241	1.124	0.558	0.590	0.510	1.081	0.700	0.525
stk-ninety	0.493	0.401	1.591	0.946	1.003	0.880	1.430	1.141	0.922
stk-think	5.027	5.003	5.033	5.017	5.028	5.031	5.015	5.042	5.021
stk-keying	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012	2.012
stk-mix	4.000	4.020	4.000	4.010	4.030	4.010	4.030	4.030	4.020
ntxn	5440105	5451608	5241879	5370808	5365430	5385021	5249153	5338308	5382264

Table Of Contents

General Items

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 code implemented in this benchmark.

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.

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.

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.).*

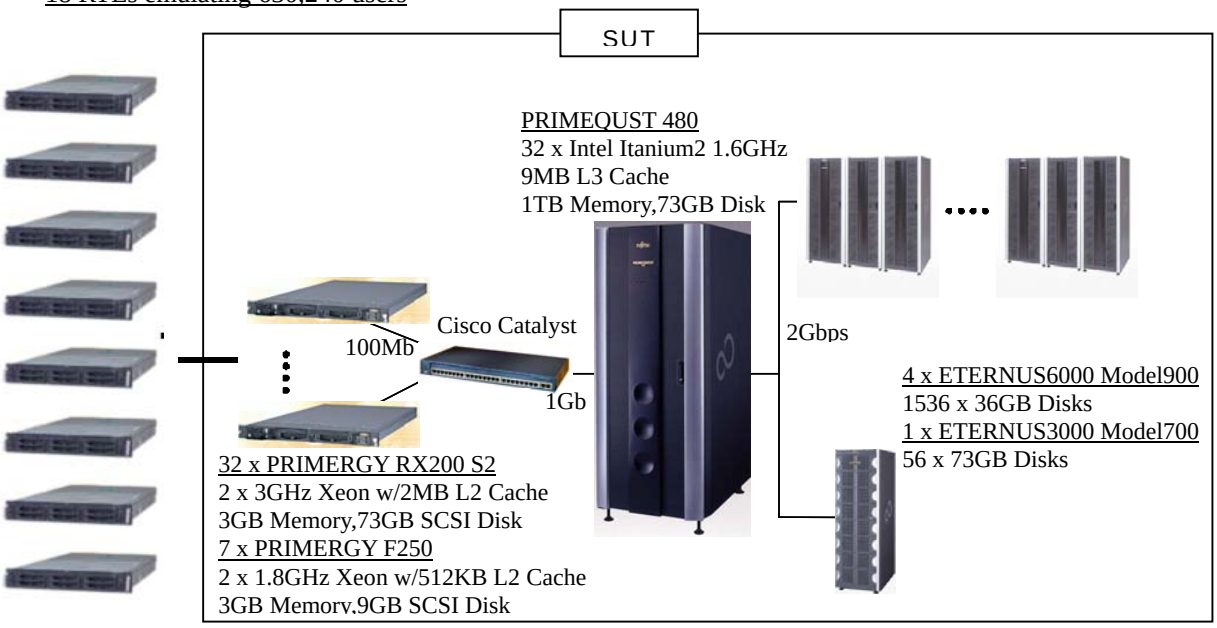
The System Under Test (SUT), a PRIMEQUEST 480 c/s w/ 39 Front-Ends, is depicted in the following diagrams.

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

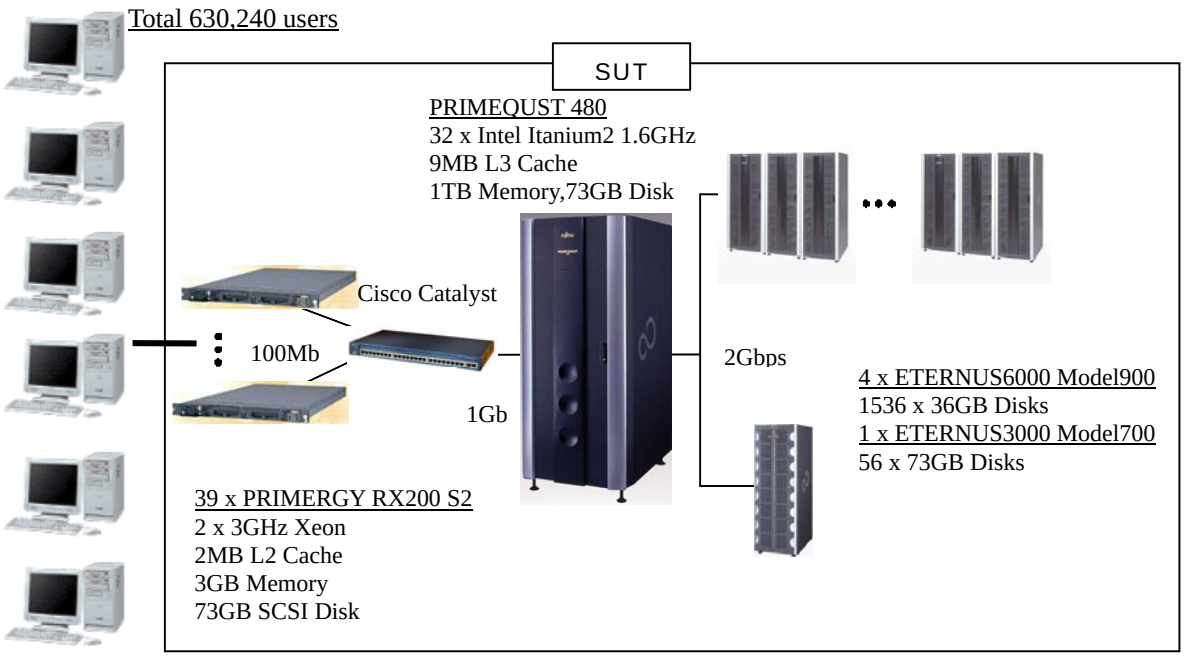
There were differences between the priced and measured configurations. The differences are:

- A RTE was used in the tested configuration.
- The clients that use Xeon@1.8GHz in the measurement were replaced by those that use Xeon@3.0GHz in the priced configuration.

PRIMEQUEST 480 Tested Configuration



PRIMEQUEST 480 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 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(see Clause 1.6), any such partitioning must be disclosed.

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	60.00%
Order Status	Accessed by last name	60.00%
Delivery	Skipped transactions	None
Transaction Mix	New Order	44.94%
	Payment	43.02%
	Order status	4.01%
	Delivery	4.01%
	Stock level	4.02%

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 And Loss of Data Disk

To demonstrate recovery from a permanent failure of durable media containing the Oracle recovery log data and TPC-C tables, the following steps were executed using 63,024 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 630,240 users.
4. The test was allowed to run for a minimum of 5 minutes.
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. The test was allowed to run for another 5 minutes and a disk array failure was caused by removing a disk from the disk array cabinet.
7. The RTE was shut down.
8. Oracle was shutdown abort.
9. New disks were returned into the disk cabinet to recover the RAID system.
10. Data from the backup disks was restored.
11. Oracle was restarted and the media recovery utility started.
12. Step 2 was repeated and the difference between the first and second counts was noted.
13. The success file was used to determine the number of NEW_ORDERS successfully returned to the RTE.
14. The counts in step 12 and 13 were compared, verifying that all committed transactions were successfully recovered.
15. 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.

3.5.2 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 63,960 warehouses under a full load of 639,600 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 639,600 users.
3. The test was allowed to run for a minimum of 5 minutes.
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 8 and 9 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.

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 68,000 warehouses.

Table 4.1 Number of Rows for Server

Table	Occurrences
Warehouse	68,000
District	680,000
Customer	2,040,000,000
History	2,040,000,000
Order	2,040,000,000
New Order	612,000,000
Order Line	20,399,769,632
Stock	6,800,000,000
Item	100,000

4.2 Constant Values

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

C_LAST (Build) **1**
C_LAST (RUN) **111**

4.3 Database Layout

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

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

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_0	raw1	sdb2		
cust_0_0	raw501	sdb3		
system_1	raw1001	sdb4		
stok_0_1	raw2	sdw2		
cust_0_1	raw502	sdw3		
tpccaux	raw1002	sdw4		
stok_0_2	raw3	sdar2		
cust_0_2	raw503	sdar3		
sp_0	raw1003	sdar4		
stok_0_3	raw4	sdbm2		
cust_0_3	raw504	sdbm3		
log_1_1	raw1004	sdbm4	0	16
stok_0_4	raw5	sdck2		
cust_0_4	raw505	sdck3		
log_1_2	raw1005	sdck4		
stok_0_5	raw6	sddp2		
cust_0_5	raw506	sddp3		
roll1	raw1006	sddp4		
stok_0_6	raw7	sdek2		
cust_0_6	raw507	sdek3		
ware_0_0	raw1007	sdek4		
stok_0_7	raw8	sdff2		
cust_0_7	raw508	sdff3		
dist_0_0	raw1008	sdff4		
stok_0_8	raw9	sdgk2		
cust_0_8	raw509	sdgk3		
item_0_0	raw1009	sdgk4		
stok_0_9	raw10	sdhf2		
cust_0_9	raw510	sdhf3		
iware_0_0	raw1010	sdhf4		
stok_0_10	raw11	sdia2		
cust_0_10	raw511	sdia3		
idist_0_0	raw1011	sdia4		
stok_0_11	raw12	sdiy2		
cust_0_11	raw512	sdiy3		
iitem_0_0	raw1012	sdiy4	0	16
stok_0_12	raw13	sdjt2		
cust_0_12	raw513	sdjt3		
temp_0_0	raw1013	sdjt4		
stok_0_13	raw14	sdko2		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_248	raw249	sduj2		
cust_0_248	raw749	sduj3		
stok_0_249	raw250	sdve2		
cust_0_249	raw750	sdve3		
stok_0_250	raw251	sdvz2		
cust_0_250	raw751	sdvz3		
stok_0_251	raw252	sdxe2		
cust_0_251	raw752	sdxe3		
stok_0_252	raw253	sdk2	0	16
cust_0_252	raw753	sdk3		
stok_0_253	raw254	sdaf2		
cust_0_253	raw754	sdaf3		
stok_0_254	raw255	sdba2		
cust_0_254	raw755	sdba3		
stok_0_255	raw256	sdbv2		
cust_0_255	raw756	sdbv3		
stok_0_256	raw257	sdct2		
cust_0_256	raw757	sdct3		
stok_0_257	raw258	sddy2		
cust_0_257	raw758	sddy3		
stok_0_258	raw259	sdet2		
cust_0_258	raw759	sdet3		
stok_0_259	raw260	sdfo2		
cust_0_259	raw760	sdfo3		
stok_0_260	raw261	sdgt2	0	16
cust_0_260	raw761	sdgt3		
stok_0_261	raw262	sdho2		
cust_0_261	raw762	sdho3		
stok_0_262	raw263	sdij2		
cust_0_262	raw763	sdij3		
stok_0_263	raw264	sdjh2		
cust_0_263	raw764	sdjh3		
stok_0_264	raw265	sdkc2		
cust_0_264	raw765	sdkc3		
stok_0_265	raw266	sdkx2		
cust_0_265	raw766	sdkx3		
stok_0_266	raw267	sdls2		
cust_0_266	raw767	sdls3		
stok_0_267	raw268	sdmn2		
cust_0_267	raw768	sdmn3		

datafile name	raw name	device name	RAID Level	# of Disk
cust_0_13	raw514	sdko3		
temp_0_1	raw1014	sdko4		
stok_0_14	raw15	sdlj2		
cust_0_14	raw515	sdlj3		
temp_0_2	raw1015	sdlj4		
stok_0_15	raw16	sdme2		
cust_0_15	raw516	sdme3		
temp_0_3	raw1016	sdme4		
stok_0_16	raw17	sdmz2		
cust_0_16	raw517	sdmz3		
temp_0_4	raw1017	sdmz4		
stok_0_17	raw18	sdoe2		
cust_0_17	raw518	sdoe3		
temp_0_5	raw1018	sdoe4		
stok_0_18	raw19	sdoz2		
cust_0_18	raw519	sdoz3		
temp_0_6	raw1019	sdoz4		
stok_0_19	raw20	sdpu2		
cust_0_19	raw520	sdpu3		
temp_0_7	raw1020	sdpu4	0	16
stok_0_20	raw21	sdqp2		
cust_0_20	raw521	sdqp3		
temp_0_8	raw1021	sdqp4		
stok_0_21	raw22	sdrn2		
cust_0_21	raw522	sdrn3		
temp_0_9	raw1022	sdrn4		
stok_0_22	raw23	sdsi2		
cust_0_22	raw523	sdsi3		
temp_0_10	raw1023	sdsi4		
stok_0_23	raw24	sdtd2		
cust_0_23	raw524	sdtd3		
temp_0_11	raw1024	sdtd4		
stok_0_24	raw25	sdub2		
cust_0_24	raw525	sdub3		
temp_0_12	raw1025	sdub4		
stok_0_25	raw26	sduw2		
cust_0_25	raw526	sduw3		
temp_0_13	raw1026	sduw4		
stok_0_26	raw27	sdvr2		
cust_0_26	raw527	sdvr3		
temp_0_14	raw1027	sdvr4		
stok_0_27	raw28	sdww2		
cust_0_27	raw528	sdww3		
temp_0_15	raw1028	sdww4	0	16
stok_0_28	raw29	sdc2		
cust_0_28	raw529	sdc3		
temp_0_16	raw1029	sdc4		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_268	raw269	sdni2	0	16
cust_0_268	raw769	sdni3		
stok_0_269	raw270	sdon2		
cust_0_269	raw770	sdon3		
stok_0_270	raw271	sdpi2		
cust_0_270	raw771	sdpi3		
stok_0_271	raw272	sdqd2		
cust_0_271	raw772	sdqd3		
stok_0_272	raw273	sdqy2		
cust_0_272	raw773	sdqy3		
stok_0_273	raw274	sdrw2		
cust_0_273	raw774	sdrw3		
stok_0_274	raw275	sdsr2		
cust_0_274	raw775	sdsr3		
stok_0_275	raw276	sdtm2		
cust_0_275	raw776	sdtm3		
stok_0_276	raw277	sduk2	0	16
cust_0_276	raw777	sduk3		
stok_0_277	raw278	sdvf2		
cust_0_277	raw778	sdvf3		
stok_0_278	raw279	sdwa2		
cust_0_278	raw779	sdwa3		
stok_0_279	raw280	sdxf2		
cust_0_279	raw780	sdxf3		
stok_0_280	raw281	sdl2		
cust_0_280	raw781	sdl3		
stok_0_281	raw282	sdag2		
cust_0_281	raw782	sdag3		
stok_0_282	raw283	sdbb2		
cust_0_282	raw783	sdbb3		
stok_0_283	raw284	sdbw2		
cust_0_283	raw784	sdbw3		
stok_0_284	raw285	sdcu2	0	16
cust_0_284	raw785	sdcu3		
stok_0_285	raw286	sddz2		
cust_0_285	raw786	sddz3		
stok_0_286	raw287	sdeu2		
cust_0_286	raw787	sdeu3		
stok_0_287	raw288	sdfp2		
cust_0_287	raw788	sdfp3		
stok_0_288	raw289	sdgu2		
cust_0_288	raw789	sdgu3		
stok_0_289	raw290	sdhp2		
cust_0_289	raw790	sdhp3		
stok_0_290	raw291	sdik2		
cust_0_290	raw791	sdik3		
stok_0_291	raw292	sdji2		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_29	raw30	sdx2		
cust_0_29	raw530	sdx3		
temp_0_17	raw1030	sdx4		
stok_0_30	raw31	sdas2		
cust_0_30	raw531	sdas3		
temp_0_18	raw1031	sdas4		
stok_0_31	raw32	sdbn2		
cust_0_31	raw532	sdbn3		
temp_0_19	raw1032	sdbn4		
stok_0_32	raw33	sdxp2		
cust_0_32	raw533	sdxp3		
temp_0_20	raw1033	sdxp4		
stok_0_33	raw34	sddq2		
cust_0_33	raw534	sddq3		
temp_0_21	raw1034	sddq4		
stok_0_34	raw35	sdel2		
cust_0_34	raw535	sdel3		
temp_0_22	raw1035	sdel4		
stok_0_35	raw36	sdfg2		
cust_0_35	raw536	sdfg3		
temp_0_23	raw1036	sdfg4	0	16
stok_0_36	raw37	sdgl2		
cust_0_36	raw537	sdgl3		
temp_0_24	raw1037	sdgl4		
stok_0_37	raw38	sdhg2		
cust_0_37	raw538	sdhg3		
temp_0_25	raw1038	sdhg4		
stok_0_38	raw39	sdib2		
cust_0_38	raw539	sdib3		
temp_0_26	raw1039	sdib4		
stok_0_39	raw40	sdiz2		
cust_0_39	raw540	sdiz3		
temp_0_27	raw1040	sdiz4		
stok_0_40	raw41	sdju2		
cust_0_40	raw541	sdju3		
temp_0_28	raw1041	sdju4		
stok_0_41	raw42	sdkp2		
cust_0_41	raw542	sdkp3		
temp_0_29	raw1042	sdkp4		
stok_0_42	raw43	sdlk2		
cust_0_42	raw543	sdlk3		
temp_0_30	raw1043	sdlk4		
stok_0_43	raw44	sdmf2		
cust_0_43	raw544	sdmf3		
temp_0_31	raw1044	sdmf4	0	16

datafile name	raw name	device name	RAID Level	# of Disk
cust_0_291	raw792	sdji3		
stok_0_292	raw293	sdkd2	0	16
cust_0_292	raw793	sdkd3		
stok_0_293	raw294	sdky2		
cust_0_293	raw794	sdky3		
stok_0_294	raw295	sdlt2		
cust_0_294	raw795	sdlt3		
stok_0_295	raw296	sdmo2		
cust_0_295	raw796	sdmo3		
stok_0_296	raw297	sdnj2		
cust_0_296	raw797	sdnj3		
stok_0_297	raw298	sdoo2		
cust_0_297	raw798	sdoo3		
stok_0_298	raw299	sdpj2		
cust_0_298	raw799	sdpj3		
stok_0_299	raw300	sdqe2		
cust_0_299	raw800	sdqe3		
stok_0_300	raw301	sdqz2	0	16
cust_0_300	raw801	sdqz3		
stok_0_301	raw302	sdrx2		
cust_0_301	raw802	sdrx3		
stok_0_302	raw303	sdss2		
cust_0_302	raw803	sdss3		
stok_0_303	raw304	sdtn2		
cust_0_303	raw804	sdtn3		
stok_0_304	raw305	sdul2		
cust_0_304	raw805	sdul3		
stok_0_305	raw306	sdvg2		
cust_0_305	raw806	sdvg3		
stok_0_306	raw307	sdwb2		
cust_0_306	raw807	sdwb3		
stok_0_307	raw308	sdxg2		
cust_0_307	raw808	sdxg3		
stok_0_308	raw309	sdm2	0	16
cust_0_308	raw809	sdm3		
stok_0_309	raw310	sdah2		
cust_0_309	raw810	sdah3		
stok_0_310	raw311	sdbc2		
cust_0_310	raw811	sdbc3		
stok_0_311	raw312	sdbx2		
cust_0_311	raw812	sdbx3		
stok_0_312	raw313	sdcv2		
cust_0_312	raw813	sdcv3		
stok_0_313	raw314	sdea2		
cust_0_313	raw814	sdea3		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_44	raw45	sdna2		
cust_0_44	raw545	sdna3		
temp_0_32	raw1045	sdna4		
stok_0_45	raw46	sdof2		
cust_0_45	raw546	sdof3		
temp_0_33	raw1046	sdof4		
stok_0_46	raw47	sdpa2		
cust_0_46	raw547	sdpa3		
temp_0_34	raw1047	sdpa4		
stok_0_47	raw48	sdpv2		
cust_0_47	raw548	sdpv3		
temp_0_35	raw1048	sdpv4		
stok_0_48	raw49	sdqq2		
cust_0_48	raw549	sdqq3		
temp_0_36	raw1049	sdqq4		
stok_0_49	raw50	sdro2		
cust_0_49	raw550	sdro3		
temp_0_37	raw1050	sdro4		
stok_0_50	raw51	sdsj2		
cust_0_50	raw551	sdsj3		
temp_0_38	raw1051	sdsj4		
stok_0_51	raw52	sdte2		
cust_0_51	raw552	sdte3		
temp_0_39	raw1052	sdte4	0	16
stok_0_52	raw53	sduc2		
cust_0_52	raw553	sduc3		
temp_0_40	raw1053	sduc4		
stok_0_53	raw54	sdux2		
cust_0_53	raw554	sdux3		
temp_0_41	raw1054	sdux4		
stok_0_54	raw55	sdvs2		
cust_0_54	raw555	sdvs3		
temp_0_42	raw1055	sdvs4		
stok_0_55	raw56	sdwx2		
cust_0_55	raw556	sdwx3		
temp_0_43	raw1056	sdwx4		
stok_0_56	raw57	sdd2		
cust_0_56	raw557	sdd3		
temp_0_44	raw1057	sdd4		
stok_0_57	raw58	sdy2		
cust_0_57	raw558	sdy3		
stok_0_58	raw59	sdat2		
cust_0_58	raw559	sdat3		
stok_0_59	raw60	sdbo2		
cust_0_59	raw560	sdbo3	0	16

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_314	raw315	sdev2		
cust_0_314	raw815	sdev3		
stok_0_315	raw316	sdfq2		
cust_0_315	raw816	sdfq3		
stok_0_316	raw317	sdgv2	0	16
cust_0_316	raw817	sdgv3		
stok_0_317	raw318	sdhq2		
cust_0_317	raw818	sdhq3		
stok_0_318	raw319	sdil2		
cust_0_318	raw819	sdil3		
stok_0_319	raw320	sdjj2		
cust_0_319	raw820	sdjj3		
stok_0_320	raw321	sdke2		
cust_0_320	raw821	sdke3		
stok_0_321	raw322	sdkz2		
cust_0_321	raw822	sdkz3		
stok_0_322	raw323	sdlu2		
cust_0_322	raw823	sdlu3		
stok_0_323	raw324	sdmp2		
cust_0_323	raw824	sdmp3		
stok_0_324	raw325	sdnk2	0	16
cust_0_324	raw825	sdnk3		
stok_0_325	raw326	sdop2		
cust_0_325	raw826	sdop3		
stok_0_326	raw327	sdpk2		
cust_0_326	raw827	sdpk3		
stok_0_327	raw328	sdqf2		
cust_0_327	raw828	sdqf3		
stok_0_328	raw329	sdra2		
cust_0_328	raw829	sdra3		
stok_0_329	raw330	sdry2		
cust_0_329	raw830	sdry3		
stok_0_330	raw331	sdst2		
cust_0_330	raw831	sdst3		
stok_0_331	raw332	sdto2		
cust_0_331	raw832	sdto3		
stok_0_332	raw333	sdum2	0	16
cust_0_332	raw833	sdum3		
stok_0_333	raw334	sdvh2		
cust_0_333	raw834	sdvh3		
stok_0_334	raw335	sdwc2		
cust_0_334	raw835	sdwc3		
stok_0_335	raw336	sdxh2		
cust_0_335	raw836	sdxh3		
stok_0_336	raw337	sdn2		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_60	raw61	sdcm2		
cust_0_60	raw561	sdcm3		
stok_0_61	raw62	sddr2		
cust_0_61	raw562	sddr3		
stok_0_62	raw63	sdem2		
cust_0_62	raw563	sdem3		
hist_0_0	raw1063	sdem4		
stok_0_63	raw64	sdfh2		
cust_0_63	raw564	sdfh3		
hist_0_1	raw1064	sdfh4		
stok_0_64	raw65	sdgm2		
cust_0_64	raw565	sdgm3		
hist_0_2	raw1065	sdgm4		
stok_0_65	raw66	sdhh2		
cust_0_65	raw566	sdhh3		
hist_0_3	raw1066	sdhh4		
stok_0_66	raw67	sdic2		
cust_0_66	raw567	sdic3		
hist_0_4	raw1067	sdic4		
stok_0_67	raw68	sdja2	0	16
cust_0_67	raw568	sdja3		
hist_0_5	raw1068	sdja4		
stok_0_68	raw69	sdjv2		
cust_0_68	raw569	sdjv3		
hist_0_6	raw1069	sdjv4		
stok_0_69	raw70	sdkq2		
cust_0_69	raw570	sdkq3		
hist_0_7	raw1070	sdkq4		
stok_0_70	raw71	sdll2		
cust_0_70	raw571	sdll3		
hist_0_8	raw1071	sdll4		
stok_0_71	raw72	sdmg2		
cust_0_71	raw572	sdmg3		
hist_0_9	raw1072	sdmg4		
stok_0_72	raw73	sdnb2		
cust_0_72	raw573	sdnb3		
hist_0_10	raw1073	sdnb4		
stok_0_73	raw74	sdog2		
cust_0_73	raw574	sdog3		
hist_0_11	raw1074	sdog4		
stok_0_74	raw75	sdpb2		
cust_0_74	raw575	sdpb3		
hist_0_12	raw1075	sdpb4		
stok_0_75	raw76	sdpw2		
cust_0_75	raw576	sdpw3	0	16

datafile name	raw name	device name	RAID Level	# of Disk
cust_0_336	raw837	sdn3		
stok_0_337	raw338	sdai2		
cust_0_337	raw838	sdai3		
stok_0_338	raw339	sdbd2		
cust_0_338	raw839	sdbd3		
stok_0_339	raw340	sdby2		
cust_0_339	raw840	sdby3		
stok_0_340	raw341	sdcw2	0	16
cust_0_340	raw841	sdcw3		
stok_0_341	raw342	sdeb2		
cust_0_341	raw842	sdeb3		
stok_0_342	raw343	sdew2		
cust_0_342	raw843	sdew3		
stok_0_343	raw344	sdfr2		
cust_0_343	raw844	sdfr3		
stok_0_344	raw345	sdgw2		
cust_0_344	raw845	sdgw3		
stok_0_345	raw346	sdhr2		
cust_0_345	raw846	sdhr3		
stok_0_346	raw347	sdim2		
cust_0_346	raw847	sdim3		
stok_0_347	raw348	sdjk2		
cust_0_347	raw848	sdjk3		
stok_0_348	raw349	sdkf2	0	16
cust_0_348	raw849	sdkf3		
stok_0_349	raw350	sdla2		
cust_0_349	raw850	sdla3		
stok_0_350	raw351	sdlv2		
cust_0_350	raw851	sdlv3		
stok_0_351	raw352	sdmq2		
cust_0_351	raw852	sdmq3		
stok_0_352	raw353	sdnl2		
cust_0_352	raw853	sdnl3		
stok_0_353	raw354	sdoq2		
cust_0_353	raw854	sdoq3		
stok_0_354	raw355	sdpl2		
cust_0_354	raw855	sdpl3		
stok_0_355	raw356	sdqg2		
cust_0_355	raw856	sdqg3		
stok_0_356	raw357	sdrb2	0	16
cust_0_356	raw857	sdrb3		
stok_0_357	raw358	sdrz2		
cust_0_357	raw858	sdrz3		
stok_0_358	raw359	sdsu2		
cust_0_358	raw859	sdsu3		

datafile name	raw name	device name	RAID Level	# of Disk
hist_0_13	raw1076	sdpw4		
stok_0_76	raw77	sdqr2		
cust_0_76	raw577	sdqr3		
hist_0_14	raw1077	sdqr4		
stok_0_77	raw78	sdrp2		
cust_0_77	raw578	sdrp3		
hist_0_15	raw1078	sdrp4		
stok_0_78	raw79	sdsk2		
cust_0_78	raw579	sdsk3		
hist_0_16	raw1079	sdsk4		
stok_0_79	raw80	sdtf2		
cust_0_79	raw580	sdtf3		
hist_0_17	raw1080	sdtf4		
stok_0_80	raw81	sdud2		
cust_0_80	raw581	sdud3		
hist_0_18	raw1081	sdud4		
stok_0_81	raw82	sduy2		
cust_0_81	raw582	sduy3		
hist_0_19	raw1082	sduy4		
stok_0_82	raw83	sdvt2		
cust_0_82	raw583	sdvt3		
hist_0_20	raw1083	sdvt4		
stok_0_83	raw84	sdwy2		
cust_0_83	raw584	sdwy3		
hist_0_21	raw1084	sdwy4	0	16
stok_0_84	raw85	sde2		
cust_0_84	raw585	sde3		
hist_0_22	raw1085	sde4		
stok_0_85	raw86	sdz2		
cust_0_85	raw586	sdz3		
hist_0_23	raw1086	sdz4		
stok_0_86	raw87	sdau2		
cust_0_86	raw587	sdau3		
hist_0_24	raw1087	sdau4		
stok_0_87	raw88	sdbp2		
cust_0_87	raw588	sdbp3		
hist_0_25	raw1088	sdbp4		
stok_0_88	raw89	sdcn2		
cust_0_88	raw589	sdcn3		
istok_0_0	raw1089	sdcn4		
stok_0_89	raw90	sdds2		
cust_0_89	raw590	sdds3		
istok_0_1	raw1090	sdds4		
stok_0_90	raw91	sden2		
cust_0_90	raw591	sden3		
istok_0_2	raw1091	sden4		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_359	raw360	sdtp2		
cust_0_359	raw860	sdtp3		
stok_0_360	raw361	sdun2		
cust_0_360	raw861	sdun3		
stok_0_361	raw362	sdvi2		
cust_0_361	raw862	sdvi3		
stok_0_362	raw363	sdwd2		
cust_0_362	raw863	sdwd3		
stok_0_363	raw364	sdxi2		
cust_0_363	raw864	sdxi3		
stok_0_364	raw365	sdo2	0	16
cust_0_364	raw865	sdo3		
stok_0_365	raw366	sdaj2		
cust_0_365	raw866	sdaj3		
stok_0_366	raw367	sdbe2		
cust_0_366	raw867	sdbe3		
stok_0_367	raw368	sdbz2		
cust_0_367	raw868	sdbz3		
stok_0_368	raw369	sdcx2		
cust_0_368	raw869	sdcx3		
stok_0_369	raw370	sdec2		
cust_0_369	raw870	sdec3		
stok_0_370	raw371	sdex2		
cust_0_370	raw871	sdex3		
stok_0_371	raw372	sdfs2		
cust_0_371	raw872	sdfs3		
stok_0_372	raw373	sdgx2	0	16
cust_0_372	raw873	sdgx3		
stok_0_373	raw374	sdhs2		
cust_0_373	raw874	sdhs3		
stok_0_374	raw375	sdin2		
cust_0_374	raw875	sdin3		
stok_0_375	raw376	sdjl2		
cust_0_375	raw876	sdjl3		
stok_0_376	raw377	sdkg2		
cust_0_376	raw877	sdkg3		
stok_0_377	raw378	sdlb2		
cust_0_377	raw878	sdlb3		
stok_0_378	raw379	sdlw2		
cust_0_378	raw879	sdlw3		
stok_0_379	raw380	sdmr2		
cust_0_379	raw880	sdmr3		
stok_0_380	raw381	sdnm2	0	16
cust_0_380	raw881	sdnm3		
stok_0_381	raw382	sdor2		
cust_0_381	raw882	sdor3		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_91	raw92	sdfi2		
cust_0_91	raw592	sdfi3		
istok_0_3	raw1092	sdfi4		
stok_0_92	raw93	sdgn2	0	16
cust_0_92	raw593	sdgn3		
istok_0_4	raw1093	sdgn4		
stok_0_93	raw94	sdhi2		
cust_0_93	raw594	sdhi3		
istok_0_5	raw1094	sdhi4		
stok_0_94	raw95	sdid2		
cust_0_94	raw595	sdid3		
istok_0_6	raw1095	sdid4		
stok_0_95	raw96	sdjb2		
cust_0_95	raw596	sdjb3		
istok_0_7	raw1096	sdjb4		
stok_0_96	raw97	sdjw2		
cust_0_96	raw597	sdjw3		
istok_0_8	raw1097	sdjw4		
stok_0_97	raw98	sdkr2		
cust_0_97	raw598	sdkr3		
istok_0_9	raw1098	sdkr4		
stok_0_98	raw99	sdlm2		
cust_0_98	raw599	sdlm3		
istok_0_10	raw1099	sdlm4		
stok_0_99	raw100	sdmh2	0	16
cust_0_99	raw600	sdmh3		
istok_0_11	raw1100	sdmh4		
stok_0_100	raw101	sdnc2		
cust_0_100	raw601	sdnc3		
stok_0_101	raw102	sdoh2		
cust_0_101	raw602	sdoh3		
stok_0_102	raw103	sdpc2		
cust_0_102	raw603	sdpc3		
stok_0_103	raw104	sdpx2		
cust_0_103	raw604	sdpx3		
stok_0_104	raw105	sdqs2		
cust_0_104	raw605	sdqs3		
stok_0_105	raw106	sdrq2		
cust_0_105	raw606	sdrq3		
stok_0_106	raw107	sdsl2		
cust_0_106	raw607	sdsl3		
stok_0_107	raw108	sdtg2		
cust_0_107	raw608	sdtg3		
stok_0_108	raw109	sdue2	0	16
cust_0_108	raw609	sdue3		
stok_0_109	raw110	sduz2		
cust_0_109	raw610	sduz3		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_382	raw383	sdpm2		
cust_0_382	raw883	sdpm3		
stok_0_383	raw384	sdqh2		
cust_0_383	raw884	sdqh3		
stok_0_384	raw385	sdrc2		
cust_0_384	raw885	sdrc3		
stok_0_385	raw386	sdsa2		
cust_0_385	raw886	sdsa3		
stok_0_386	raw387	sdsv2		
cust_0_386	raw887	sdsv3		
stok_0_387	raw388	sdtq2		
cust_0_387	raw888	sdtq3		
stok_0_388	raw389	sduo2	0	16
cust_0_388	raw889	sduo3		
stok_0_389	raw390	sdvj2		
cust_0_389	raw890	sdvj3		
stok_0_390	raw391	sdwe2		
cust_0_390	raw891	sdwe3		
stok_0_391	raw392	sdxj2		
cust_0_391	raw892	sdxj3		
stok_0_392	raw393	sdp2		
cust_0_392	raw893	sdp3		
stok_0_393	raw394	sdak2		
cust_0_393	raw894	sdak3		
stok_0_394	raw395	sdbf2		
cust_0_394	raw895	sdbf3		
stok_0_395	raw396	sdca2		
cust_0_395	raw896	sdca3		
stok_0_396	raw397	sdcy2	0	16
cust_0_396	raw897	sdcy3		
stok_0_397	raw398	sded2		
cust_0_397	raw898	sded3		
stok_0_398	raw399	sdey2		
cust_0_398	raw899	sdey3		
stok_0_399	raw400	sdft2		
cust_0_399	raw900	sdft3		
stok_0_400	raw401	sdgy2		
cust_0_400	raw901	sdgy3		
stok_0_401	raw402	sdht2		
cust_0_401	raw902	sdht3		
stok_0_402	raw403	sdio2		
cust_0_402	raw903	sdio3		
stok_0_403	raw404	sdjm2		
cust_0_403	raw904	sdjm3		
stok_0_404	raw405	sdkh2	0	16
cust_0_404	raw905	sdkh3		
stok_0_405	raw406	sdlc2		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_110	raw111	sdvu2		
cust_0_110	raw611	sdvu3		
istok_0_12	raw1111	sdvu4		
stok_0_111	raw112	sdwz2		
cust_0_111	raw612	sdwz3		
istok_0_13	raw1112	sdwz4		
stok_0_112	raw113	sdf2		
cust_0_112	raw613	sdf3		
istok_0_14	raw1113	sdf4		
stok_0_113	raw114	sdaa2		
cust_0_113	raw614	sdaa3		
istok_0_15	raw1114	sdaa4		
stok_0_114	raw115	sdav2		
cust_0_114	raw615	sdav3		
istok_0_16	raw1115	sdav4		
stok_0_115	raw116	sdbq2		
cust_0_115	raw616	sdbq3		
istok_0_17	raw1116	sdbq4	0	16
stok_0_116	raw117	sdco2		
cust_0_116	raw617	sdco3		
istok_0_18	raw1117	sdco4		
stok_0_117	raw118	sddt2		
cust_0_117	raw618	sddt3		
istok_0_19	raw1118	sddt4		
stok_0_118	raw119	sdeo2		
cust_0_118	raw619	sdeo3		
istok_0_20	raw1119	sdeo4		
stok_0_119	raw120	sdfj2		
cust_0_119	raw620	sdfj3		
istok_0_21	raw1120	sdfj4		
stok_0_120	raw121	sdgo2		
cust_0_120	raw621	sdgo3		
istok_0_22	raw1121	sdgo4		
stok_0_121	raw122	sdhj2		
cust_0_121	raw622	sdhj3		
istok_0_23	raw1122	sdhj4		
stok_0_122	raw123	sdie2		
cust_0_122	raw623	sdie3		
icust1_0_0	raw1123	sdie4		
stok_0_123	raw124	sdjc2		
cust_0_123	raw624	sdjc3	0	16
icust1_0_1	raw1124	sdjc4		
stok_0_124	raw125	sdjx2		
cust_0_124	raw625	sdjx3		
icust1_0_2	raw1125	sdjx4		

datafile name	raw name	device name	RAID Level	# of Disk
cust_0_405	raw906	sdlc3		
stok_0_406	raw407	sdlx2		
cust_0_406	raw907	sdlx3		
stok_0_407	raw408	sdms2		
cust_0_407	raw908	sdms3		
stok_0_408	raw409	sdnn2		
cust_0_408	raw909	sdnn3		
stok_0_409	raw410	sdos2		
cust_0_409	raw910	sdos3		
stok_0_410	raw411	sdpn2		
cust_0_410	raw911	sdpn3		
stok_0_411	raw412	sdqi2		
cust_0_411	raw912	sdqi3		
stok_0_412	raw413	sdrd2	0	16
cust_0_412	raw913	sdrd3		
stok_0_413	raw414	sdsb2		
cust_0_413	raw914	sdsb3		
stok_0_414	raw415	sdsw2		
cust_0_414	raw915	sdsw3		
stok_0_415	raw416	sdtr2		
cust_0_415	raw916	sdtr3		
stok_0_416	raw417	sdup2		
cust_0_416	raw917	sdup3		
stok_0_417	raw418	sdvk2		
cust_0_417	raw918	sdvk3		
stok_0_418	raw419	sdwf2		
cust_0_418	raw919	sdwf3		
stok_0_419	raw420	sdxk2		
cust_0_419	raw920	sdxk3		
stok_0_420	raw421	sdq2	0	16
cust_0_420	raw921	sdq3		
stok_0_421	raw422	sdal2		
cust_0_421	raw922	sdal3		
stok_0_422	raw423	sdbg2		
cust_0_422	raw923	sdbg3		
stok_0_423	raw424	sdcb2		
cust_0_423	raw924	sdcb3		
stok_0_424	raw425	sdcz2		
cust_0_424	raw925	sdcz3		
stok_0_425	raw426	sdee2		
cust_0_425	raw926	sdee3		
stok_0_426	raw427	sdez2		
cust_0_426	raw927	sdez3		
stok_0_427	raw428	sdfu2		
cust_0_427	raw928	sdfu3		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_125	raw126	sdks2		
cust_0_125	raw626	sdks3		
icust1_0_3	raw1126	sdks4		
stok_0_126	raw127	sdln2		
cust_0_126	raw627	sdln3		
icust1_0_4	raw1127	sdln4		
stok_0_127	raw128	sdmi2		
cust_0_127	raw628	sdmi3		
icust1_0_5	raw1128	sdmi4		
stok_0_128	raw129	sdnd2		
cust_0_128	raw629	sdnd3		
stok_0_129	raw130	sdoi2		
cust_0_129	raw630	sdoi3		
system_2	raw1130	sdoi4		
stok_0_130	raw131	sdpd2		
cust_0_130	raw631	sdpd3		
stok_0_131	raw132	sdpy2	0	16
cust_0_131	raw632	sdpy3		
stok_0_132	raw133	sdqt2		
cust_0_132	raw633	sdqt3		
stok_0_133	raw134	sdrr2		
cust_0_133	raw634	sdrr3		
stok_0_134	raw135	sdsm2		
cust_0_134	raw635	sdsm3		
stok_0_135	raw136	sdth2		
cust_0_135	raw636	sdth3		
stok_0_136	raw137	sduf2		
cust_0_136	raw637	sduf3		
stok_0_137	raw138	sdva2		
cust_0_137	raw638	sdva3		
stok_0_138	raw139	sdvv2		
cust_0_138	raw639	sdvv3		
stok_0_139	raw140	sdxa2		
cust_0_139	raw640	sdxa3		
stok_0_140	raw141	sdg2	0	16
cust_0_140	raw641	sdg3		
stok_0_141	raw142	sdab2		
cust_0_141	raw642	sdab3		
stok_0_142	raw143	sdaw2		
cust_0_142	raw643	sdaw3		
stok_0_143	raw144	sdbr2		
cust_0_143	raw644	sdbr3		
stok_0_144	raw145	sdcp2		
cust_0_144	raw645	sdcp3		
stok_0_145	raw146	sddu2		
cust_0_145	raw646	sddu3		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_428	raw429	sdgz2	0	16
cust_0_428	raw929	sdgz3		
stok_0_429	raw430	sdhu2		
cust_0_429	raw930	sdhu3		
stok_0_430	raw431	sdip2		
cust_0_430	raw931	sdip3		
stok_0_431	raw432	sdjn2		
cust_0_431	raw932	sdjn3		
stok_0_432	raw433	sdki2		
cust_0_432	raw933	sdki3		
stok_0_433	raw434	sdld2		
cust_0_433	raw934	sdld3		
stok_0_434	raw435	sdly2		
cust_0_434	raw935	sdly3		
stok_0_435	raw436	sdmt2		
cust_0_435	raw936	sdmt3		
stok_0_436	raw437	sdno2	0	16
cust_0_436	raw937	sdno3		
stok_0_437	raw438	sdot2		
cust_0_437	raw938	sdot3		
stok_0_438	raw439	sdpo2		
cust_0_438	raw939	sdpo3		
stok_0_439	raw440	sdqj2		
cust_0_439	raw940	sdqj3		
stok_0_440	raw441	sdre2		
cust_0_440	raw941	sdre3		
stok_0_441	raw442	sdsc2		
cust_0_441	raw942	sdsc3		
stok_0_442	raw443	sdsx2		
cust_0_442	raw943	sdsx3		
stok_0_443	raw444	sdts2		
cust_0_443	raw944	sdts3		
stok_0_444	raw445	sduq2	0	16
cust_0_444	raw945	sduq3		
stok_0_445	raw446	sdvl2		
cust_0_445	raw946	sdvl3		
stok_0_446	raw447	sdwg2		
cust_0_446	raw947	sdwg3		
stok_0_447	raw448	sdxl2		
cust_0_447	raw948	sdxl3		
stok_0_448	raw449	sdr2		
cust_0_448	raw949	sdr3		
stok_0_449	raw450	sdam2		
cust_0_449	raw950	sdam3		
stok_0_450	raw451	sdbh2		
cust_0_450	raw951	sdbh3		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_146	raw147	sdep2		
cust_0_146	raw647	sdep3		
stok_0_147	raw148	sdfk2		
cust_0_147	raw648	sdfk3		
stok_0_148	raw149	sdgp2	0	16
cust_0_148	raw649	sdgp3		
stok_0_149	raw150	sdhk2		
cust_0_149	raw650	sdhk3		
stok_0_150	raw151	sdif2		
cust_0_150	raw651	sdif3		
stok_0_151	raw152	sdjd2		
cust_0_151	raw652	sdjd3		
stok_0_152	raw153	sdjy2		
cust_0_152	raw653	sdjy3		
stok_0_153	raw154	sdkt2		
cust_0_153	raw654	sdkt3		
stok_0_154	raw155	sdlo2		
cust_0_154	raw655	sdlo3		
stok_0_155	raw156	sdmj2		
cust_0_155	raw656	sdmj3		
stok_0_156	raw157	sdne2	0	16
cust_0_156	raw657	sdne3		
stok_0_157	raw158	sdoj2		
cust_0_157	raw658	sdoj3		
stok_0_158	raw159	sdpe2		
cust_0_158	raw659	sdpe3		
stok_0_159	raw160	sdpz2		
cust_0_159	raw660	sdpz3		
stok_0_160	raw161	sdqu2		
cust_0_160	raw661	sdqu3		
stok_0_161	raw162	sdrs2		
cust_0_161	raw662	sdrs3		
stok_0_162	raw163	sdsn2		
cust_0_162	raw663	sdsn3		
stok_0_163	raw164	sdti2		
cust_0_163	raw664	sdti3		
stok_0_164	raw165	sdug2	0	16
cust_0_164	raw665	sdug3		
stok_0_165	raw166	sdvb2		
cust_0_165	raw666	sdvb3		
stok_0_166	raw167	sdvw2		
cust_0_166	raw667	sdvw3		
stok_0_167	raw168	sdxb2		
cust_0_167	raw668	sdxb3		
stok_0_168	raw169	sdh2		
cust_0_168	raw669	sdh3		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_451	raw452	sdda2		
cust_0_451	raw952	sdda3		
stok_0_452	raw453	sdef2	0	16
cust_0_452	raw953	sdef3		
stok_0_453	raw454	sdfa2		
cust_0_453	raw954	sdfa3		
stok_0_454	raw455	sdfv2		
cust_0_454	raw955	sdfv3		
stok_0_455	raw456	sdha2		
cust_0_455	raw956	sdha3		
stok_0_456	raw457	sdhv2		
cust_0_456	raw957	sdhv3		
stok_0_457	raw458	sdjo2		
cust_0_457	raw958	sdjo3		
stok_0_458	raw459	sdkj2		
cust_0_458	raw959	sdkj3		
stok_0_459	raw460	sdle2		
cust_0_459	raw960	sdle3		
stok_0_460	raw461	sdlz2	0	16
cust_0_460	raw961	sdlz3		
stok_0_461	raw462	sdmu2		
cust_0_461	raw962	sdmu3		
stok_0_462	raw463	sdnp2		
cust_0_462	raw963	sdnp3		
stok_0_463	raw464	sdou2		
cust_0_463	raw964	sdou3		
stok_0_464	raw465	sdpp2		
cust_0_464	raw965	sdpp3		
stok_0_465	raw466	sdqk2		
cust_0_465	raw966	sdqk3		
stok_0_466	raw467	sdsd2		
cust_0_466	raw967	sdsd3		
stok_0_467	raw468	sdsy2		
cust_0_467	raw968	sdsy3		
stok_0_468	raw469	sdur2	0	16
cust_0_468	raw969	sdur3		
stok_0_469	raw470	sdvm2		
cust_0_469	raw970	sdvm3		
stok_0_470	raw471	sdwh2		
cust_0_470	raw971	sdwh3		
stok_0_471	raw472	sdxm2		
cust_0_471	raw972	sdxm3		
stok_0_472	raw473	sds2		
cust_0_472	raw973	sds3		
stok_0_473	raw474	sdan2		
cust_0_473	raw974	sdan3		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_169	raw170	sdac2		
cust_0_169	raw670	sdac3		
stok_0_170	raw171	sdax2		
cust_0_170	raw671	sdax3		
stok_0_171	raw172	sdbs2		
cust_0_171	raw672	sdbs3		
stok_0_172	raw173	sdcq2	0	16
cust_0_172	raw673	sdcq3		
stok_0_173	raw174	sddv2		
cust_0_173	raw674	sddv3		
stok_0_174	raw175	sdeq2		
cust_0_174	raw675	sdeq3		
stok_0_175	raw176	sdfl2		
cust_0_175	raw676	sdfl3		
stok_0_176	raw177	sdgq2		
cust_0_176	raw677	sdgq3		
stok_0_177	raw178	sdhl2		
cust_0_177	raw678	sdhl3		
stok_0_178	raw179	sdig2		
cust_0_178	raw679	sdig3		
stok_0_179	raw180	sdje2		
cust_0_179	raw680	sdje3		
stok_0_180	raw181	sdjz2	0	16
cust_0_180	raw681	sdjz3		
stok_0_181	raw182	sdku2		
cust_0_181	raw682	sdku3		
stok_0_182	raw183	sdlp2		
cust_0_182	raw683	sdlp3		
stok_0_183	raw184	sdmk2		
cust_0_183	raw684	sdmk3		
stok_0_184	raw185	sdnf2		
cust_0_184	raw685	sdnf3		
stok_0_185	raw186	sdok2		
cust_0_185	raw686	sdok3		
stok_0_186	raw187	sdpf2		
cust_0_186	raw687	sdpf3		
stok_0_187	raw188	sdqa2		
cust_0_187	raw688	sdqa3		
stok_0_188	raw189	sdqv2	0	16
cust_0_188	raw689	sdqv3		
stok_0_189	raw190	sdrt2		
cust_0_189	raw690	sdrt3		
stok_0_190	raw191	sdso2		
cust_0_190	raw691	sdso3		
stok_0_191	raw192	sdtj2		
cust_0_191	raw692	sdtj3		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_474	raw475	sdbi2		
cust_0_474	raw975	sdbi3		
stok_0_475	raw476	sddb2		
cust_0_475	raw976	sddb3		
stok_0_476	raw477	sdeg2	0	16
cust_0_476	raw977	sdeg3		
stok_0_477	raw478	sdfb2		
cust_0_477	raw978	sdfb3		
stok_0_478	raw479	sdfw2		
cust_0_478	raw979	sdfw3		
stok_0_479	raw480	sdhb2		
cust_0_479	raw980	sdhb3		
stok_0_480	raw481	sdhw2		
cust_0_480	raw981	sdhw3		
stok_0_481	raw482	sdjp2		
cust_0_481	raw982	sdjp3		
stok_0_482	raw483	sdkk2		
cust_0_482	raw983	sdkk3		
stok_0_483	raw484	sdlf2		
cust_0_483	raw984	sdlf3		
stok_0_484	raw485	sdma2	0	16
cust_0_484	raw985	sdma3		
stok_0_485	raw486	sdmv2		
cust_0_485	raw986	sdmv3		
stok_0_486	raw487	sdnq2		
cust_0_486	raw987	sdnq3		
stok_0_487	raw488	sdov2		
cust_0_487	raw988	sdov3		
stok_0_488	raw489	sdpq2		
cust_0_488	raw989	sdpq3		
stok_0_489	raw490	sdql2		
cust_0_489	raw990	sdql3		
stok_0_490	raw491	sdse2		
cust_0_490	raw991	sdse3		
stok_0_491	raw492	sdsz2		
cust_0_491	raw992	sdsz3		
stok_0_492	raw493	sdus2	0	16
cust_0_492	raw993	sdus3		
stok_0_493	raw494	sdvn2		
cust_0_493	raw994	sdvn3		
stok_0_494	raw495	sdwi2		
cust_0_494	raw995	sdwi3		
stok_0_495	raw496	sdxn2		
cust_0_495	raw996	sdxn3		
stok_0_496	raw497	sdt2		
cust_0_496	raw997	sdt3		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_192	raw193	sduh2		
cust_0_192	raw693	sduh3		
stok_0_193	raw194	sdvc2		
cust_0_193	raw694	sdvc3		
stok_0_194	raw195	sdvx2		
cust_0_194	raw695	sdvx3		
stok_0_195	raw196	sdxc2		
cust_0_195	raw696	sdxc3		
stok_0_196	raw197	sdi2	0	16
cust_0_196	raw697	sdi3		
stok_0_197	raw198	sdad2		
cust_0_197	raw698	sdad3		
stok_0_198	raw199	sday2		
cust_0_198	raw699	sday3		
stok_0_199	raw200	sdbt2		
cust_0_199	raw700	sdbt3		
stok_0_200	raw201	sdcr2		
cust_0_200	raw701	sdcr3		
stok_0_201	raw202	sddw2		
cust_0_201	raw702	sddw3		
stok_0_202	raw203	sder2		
cust_0_202	raw703	sder3		
stok_0_203	raw204	sdfm2		
cust_0_203	raw704	sdfm3		
stok_0_204	raw205	sdgr2	0	16
cust_0_204	raw705	sdgr3		
stok_0_205	raw206	sdhm2		
cust_0_205	raw706	sdhm3		
stok_0_206	raw207	sdih2		
cust_0_206	raw707	sdih3		
stok_0_207	raw208	sdjf2		
cust_0_207	raw708	sdjf3		
stok_0_208	raw209	sdka2		
cust_0_208	raw709	sdka3		
stok_0_209	raw210	sdkv2		
cust_0_209	raw710	sdkv3		
stok_0_210	raw211	sdlq2		
cust_0_210	raw711	sdlq3		
stok_0_211	raw212	sdml2		
cust_0_211	raw712	sdml3		
stok_0_212	raw213	sdng2	0	16
cust_0_212	raw713	sdng3		
stok_0_213	raw214	sdol2		
cust_0_213	raw714	sdol3		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_497	raw498	sdao2		
cust_0_497	raw998	sdao3		
stok_0_498	raw499	sdbj2		
cust_0_498	raw999	sdbj3		
stok_0_499	raw500	sddc2	0	16
cust_0_499	raw1000	sddc3		
nord_0_0	raw1058	sdmc4		
nord_0_1	raw1059	sdmx4		
nord_0_2	raw1060	sdns4		
nord_0_3	raw1061	sdox4		
nord_0_4	raw1062	sdps4		
iordr2_0_0	raw1501	sdeh2		
iordr2_0_1	raw1502	sdfc2		
iordr2_0_2	raw1503	sdfx2	0	16
iordr2_0_3	raw1504	sdhc2		
iordr2_0_4	raw1505	sdhx2		
iordr2_0_5	raw1506	sdjq2		
iordr2_0_6	raw1507	sdkl2		
iordr2_0_7	raw1508	sdlg2		
iordr2_0_8	raw1509	sdmb2		
iordr2_0_9	raw1510	sdmw2		
iordr2_0_10	raw1511	sdnr2	0	16
iordr2_0_11	raw1512	sdow2		
iordr2_0_12	raw1513	sdpr2		
iordr2_0_13	raw1514	sdqm2		
iordr2_0_14	raw1515	sdsf2		
iordr2_0_15	raw1516	sdta2		
icust2_0_0	raw1517	sdut2		
icust2_0_1	raw1518	sdvo2		
icust2_0_2	raw1519	sdwj2	0	16
icust2_0_3	raw1520	sdxo2		
icust2_0_4	raw1521	sdu2		
icust2_0_5	raw1522	sdap2		
icust2_0_6	raw1523	sdbk2		
icust2_0_7	raw1524	sddd2		
icust2_0_8	raw1525	sdei2		
icust2_0_9	raw1526	sdfd2		
icust2_0_10	raw1527	sdfy2	0	16
icust2_0_11	raw1528	sdhd2		
icust2_0_12	raw1529	sdhy2		
icust2_0_13	raw1530	sdjr2		
icust2_0_14	raw1531	sdkm2		
icust2_0_15	raw1532	sdlh2	0	16
ordr_0_0	raw1533	sdv2		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_214	raw215	sdpg2		
cust_0_214	raw715	sdpg3		
stok_0_215	raw216	sdqb2		
cust_0_215	raw716	sdqb3		
stok_0_216	raw217	sdqw2		
cust_0_216	raw717	sdqw3		
stok_0_217	raw218	sdru2		
cust_0_217	raw718	sdru3		
stok_0_218	raw219	sdsp2		
cust_0_218	raw719	sdsp3		
stok_0_219	raw220	sdtk2		
cust_0_219	raw720	sdtk3		
stok_0_220	raw221	sdui2	0	16
cust_0_220	raw721	sdui3		
stok_0_221	raw222	sdvd2		
cust_0_221	raw722	sdvd3		
stok_0_222	raw223	sdvy2		
cust_0_222	raw723	sdvy3		
stok_0_223	raw224	sdxd2		
cust_0_223	raw724	sdxd3		
stok_0_224	raw225	sdj2		
cust_0_224	raw725	sdj3		
stok_0_225	raw226	sdae2		
cust_0_225	raw726	sdae3		
stok_0_226	raw227	sdaz2		
cust_0_226	raw727	sdaz3		
stok_0_227	raw228	sdbu2		
cust_0_227	raw728	sdbu3		
stok_0_228	raw229	sdcs2	0	16
cust_0_228	raw729	sdcs3		
stok_0_229	raw230	sddx2		
cust_0_229	raw730	sddx3		
stok_0_230	raw231	sdes2		
cust_0_230	raw731	sdes3		
stok_0_231	raw232	sdfn2		
cust_0_231	raw732	sdfn3		

datafile name	raw name	device name	RAID Level	# of Disk
ordr_0_1	raw1534	sdaq2	0	16
ordr_0_2	raw1535	sdbl2		
ordr_0_3	raw1536	sdcc2	0	16
ordr_0_4	raw1537	sdcd2		
ordr_0_5	raw1538	sdce2	0	16
ordr_0_6	raw1539	sdcf2		
ordr_0_7	raw1540	sdcg2	0	16
ordr_0_8	raw1541	sdch2		
ordr_0_9	raw1542	sdci2	0	16
ordr_0_10	raw1543	sdcj2		
ordr_0_11	raw1544	sdde2	0	16
ordr_0_12	raw1545	sdej2		
ordr_0_13	raw1546	sdfe2	0	16
ordr_0_14	raw1547	sdfz2		
ordr_0_15	raw1548	sdhe2	0	16
ordr_0_16	raw1549	sdhz2		
ordr_0_17	raw1550	sdiq2	0	16
ordr_0_18	raw1551	sdir2		
ordr_0_19	raw1552	sdis2	0	16
ordr_0_20	raw1553	sdit2		
ordr_0_21	raw1554	sdiu2	0	16
ordr_0_22	raw1555	sdiv2		
ordr_0_23	raw1556	sdiw2	0	16
ordr_0_24	raw1557	sdix2		
ordr_0_25	raw1558	sdjs2	0	16
ordr_0_26	raw1559	sdkn2		
ordr_0_27	raw1560	sdli2	0	16
ordr_0_28	raw1561	sdmd2		
ordr_0_29	raw1562	sdmy2	0	16
ordr_0_30	raw1563	sdnt2		
ordr_0_31	raw1564	sdoy2	0	16
ordr_0_32	raw1565	sdpt2		
ordr_0_33	raw1566	sdqo2	0	16
ordr_0_34	raw1567	sdrf2		
ordr_0_35	raw1568	sdrg2	0	16
ordr_0_36	raw1569	sdrh2		

datafile name	raw name	device name	RAID Level	# of Disk
stok_0_232	raw233	sdgs2		
cust_0_232	raw733	sdgs3		
stok_0_233	raw234	sdhn2		
cust_0_233	raw734	sdhn3		
stok_0_234	raw235	sdii2		
cust_0_234	raw735	sdii3		
stok_0_235	raw236	sdjg2		
cust_0_235	raw736	sdjg3		
stok_0_236	raw237	sdkb2	0	16
cust_0_236	raw737	sdkb3		
stok_0_237	raw238	sdkw2		
cust_0_237	raw738	sdkw3		
stok_0_238	raw239	sdlr2		
cust_0_238	raw739	sdlr3		
stok_0_239	raw240	sdmm 2		
cust_0_239	raw740	sdmm 3		
stok_0_240	raw241	sdnh2		
cust_0_240	raw741	sdnh3		
stok_0_241	raw242	sdom2		
cust_0_241	raw742	sdom3		
stok_0_242	raw243	sdph2		
cust_0_242	raw743	sdph3		
stok_0_243	raw244	sdqc2		
cust_0_243	raw744	sdqc3		
stok_0_244	raw245	sdqx2	0	16
cust_0_244	raw745	sdqx3		
stok_0_245	raw246	sdrv2		
cust_0_245	raw746	sdrv3		
stok_0_246	raw247	sdsq2		
cust_0_246	raw747	sdsq3		
stok_0_247	raw248	sdtl2		
cust_0_247	raw748	sdtl3		

datafile name	raw name	device name	RAID Level	# of Disk
ordr_0_37	raw1570	sdri2	0	16
ordr_0_38	raw1571	sdrj2		
ordr_0_39	raw1572	sdrk2	0	16
ordr_0_40	raw1573	sdrl2		
ordr_0_41	raw1574	sdrm2	0	16
ordr_0_42	raw1575	sdsh2		
ordr_0_43	raw1576	sdtc2	0	16
ordr_0_44	raw1577	sdtt2		
ordr_0_45	raw1578	sdtu2	0	16
ordr_0_46	raw1579	sdtv2		
ordr_0_47	raw1580	sdtw2	0	16

Table 4.2 Database Layout

The Database tables were configured with 92 RAID0 volumes. Each RAID0 volume consisted of 16 Fible Channel disks with 36GB capacity and it had either of 2 Logical Unit(LUs) or 8LUs.

The Database logs were configured with 4 RAID0+1 volumes. Each RAID0+1 volume consisted of 14 disks (7 disks + 7 disks mirrored) with 73GB capacity. A log file was configured with 4 LUs (1 LU from each RAID0+1 volume) using Linux mdadm software RAID utility to spread accesses across all 4 volumes.

4.4 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.

4.5 Database Mapping

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

The database was neither partitioned nor replicated.

4.6 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:	792,101.96 tpmC
Price per tpmC:	$8.92 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	0.353	0.710	3.113
Payment	0.346	0.703	3.114
Order-Status	0.352	0.709	2.999
Interactive Delivery	0.103	0.103	0.324
Deferred Delivery	0.250	0.607	2.892
Stock-Level	0.340	0.696	3.099
Menu	0.104	0.103	74.414

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	18.003	18.012	18.045
Payment	3.004	3.012	3.058
Order-Status	2.007	2.012	2.035
Interactive Delivery	2.007	2.012	2.048
Stock-Level	2.006	2.012	2.043

Table 5.3 Think Times

Type	Minimum	Average	Maximum
New-Order	0.000	12.014	120.196
Payment	0.000	12.015	120.199
Order-Status	0.000	10.017	100.96
Interactive Delivery	0.000	5.020	50.170
Stock-Level	0.000	5.024	50.198

5.4 Response Time Frequency Distribution Curves and Other Graphs

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

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

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

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

Figure 5.1: New Order Response Time Distribution

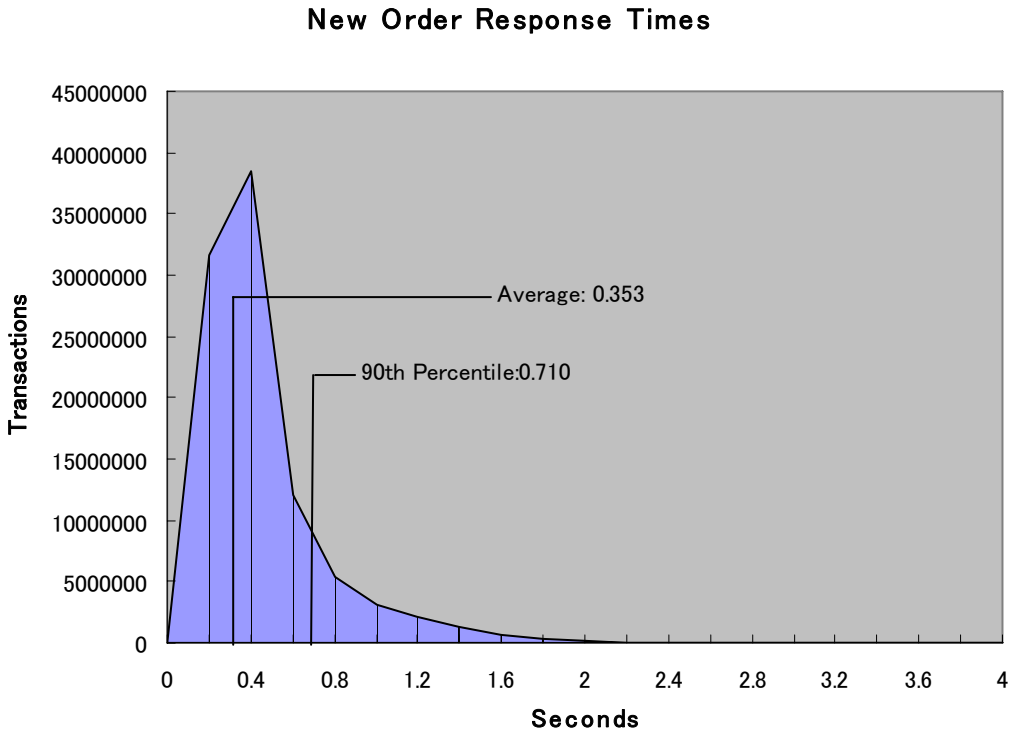


Figure 5.2: Payment Response Time Distribution

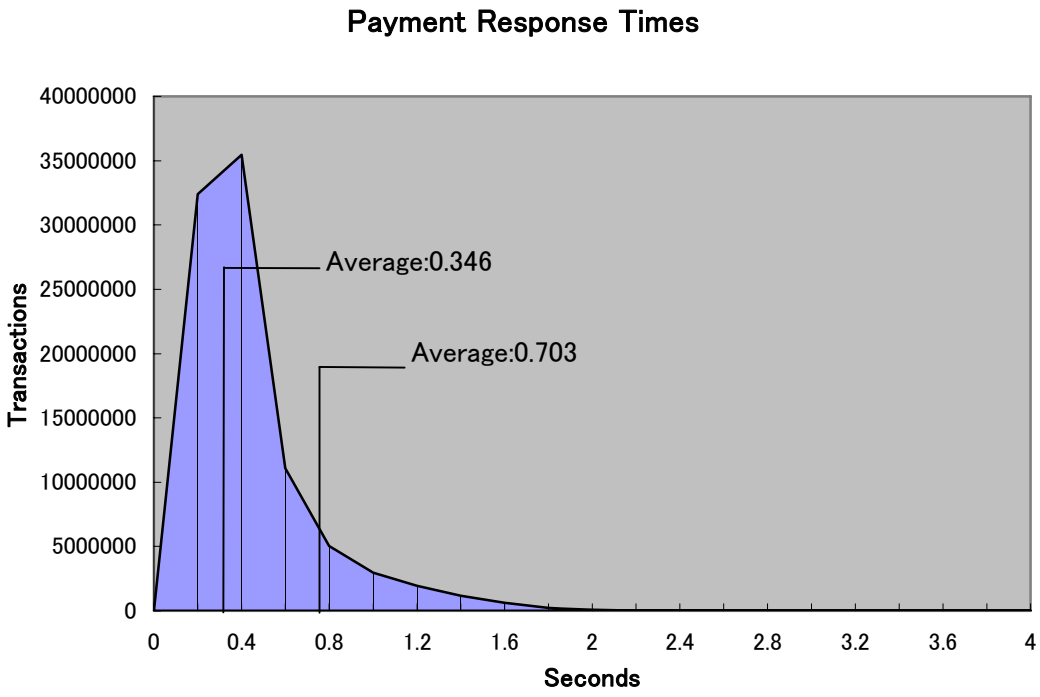


Figure 5.3: Order Status Response Time Distribution

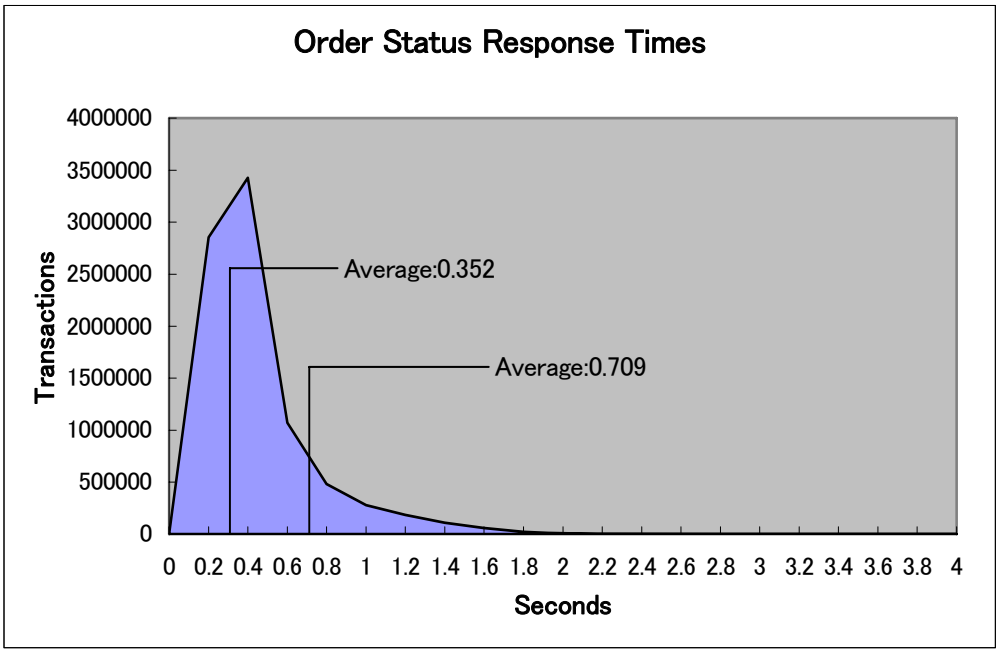


Figure 5.4: Delivery Response Time Distribution

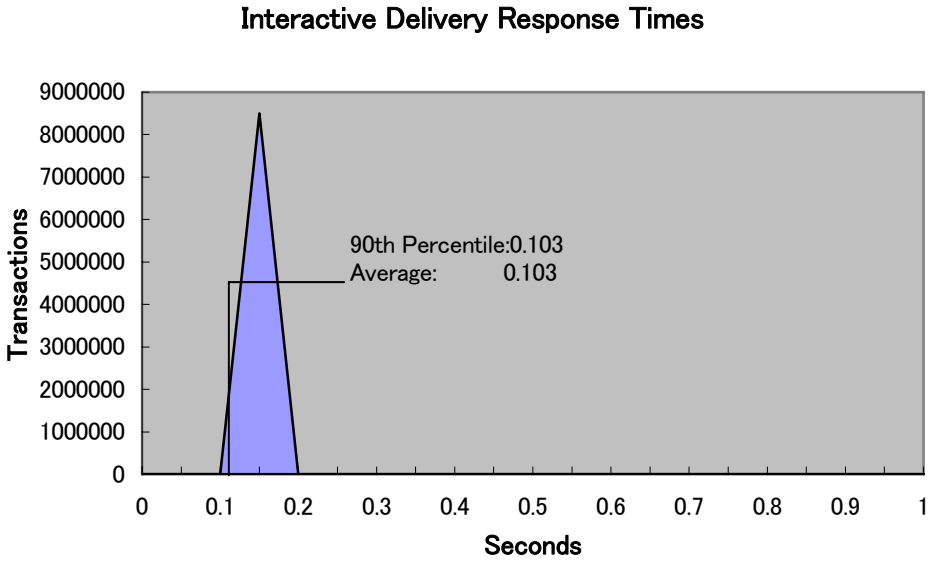


Figure 5.5: Stock Level Response Time Distribution

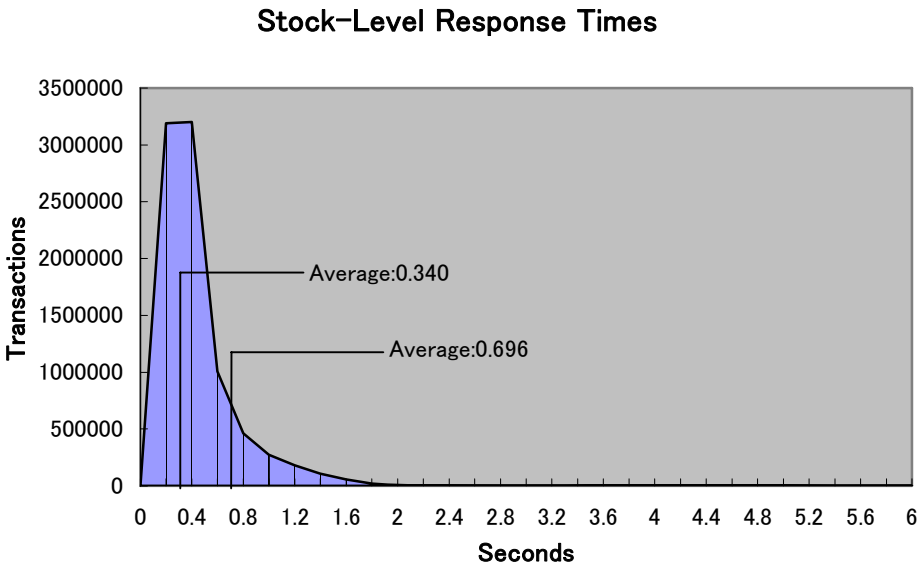


Figure 5.6: New Order Think Time Frequency Distribution

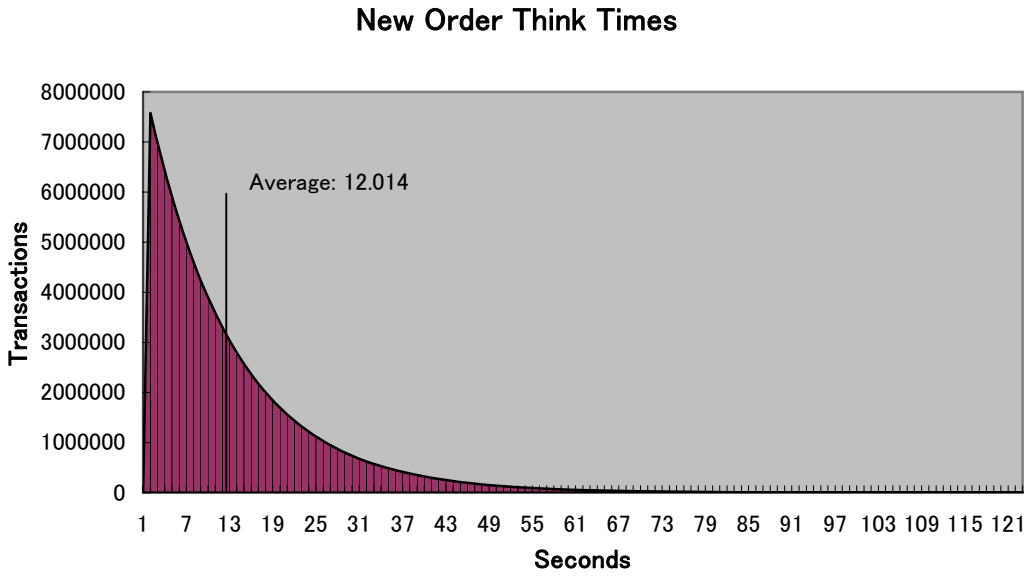


Figure 5.7: New-Order Response time vs. Throughput

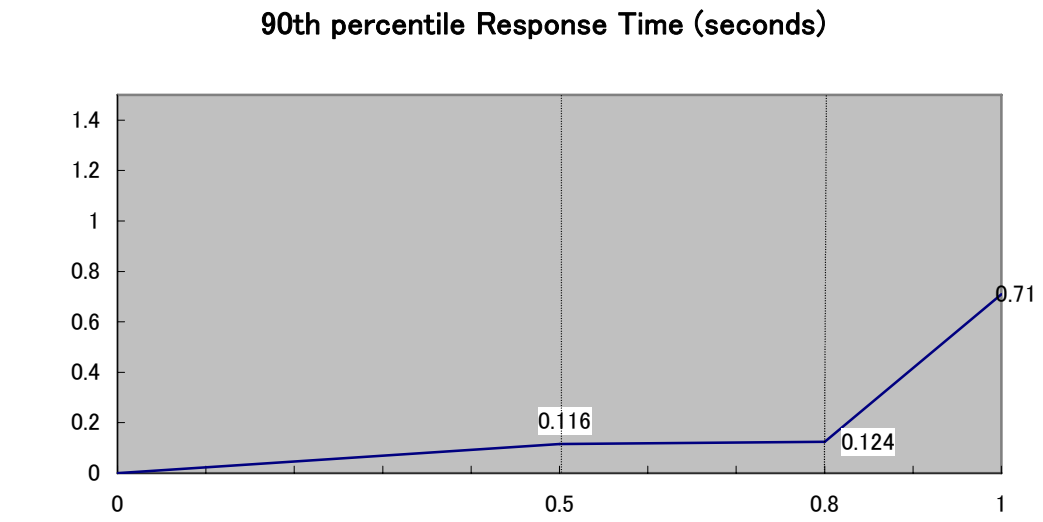
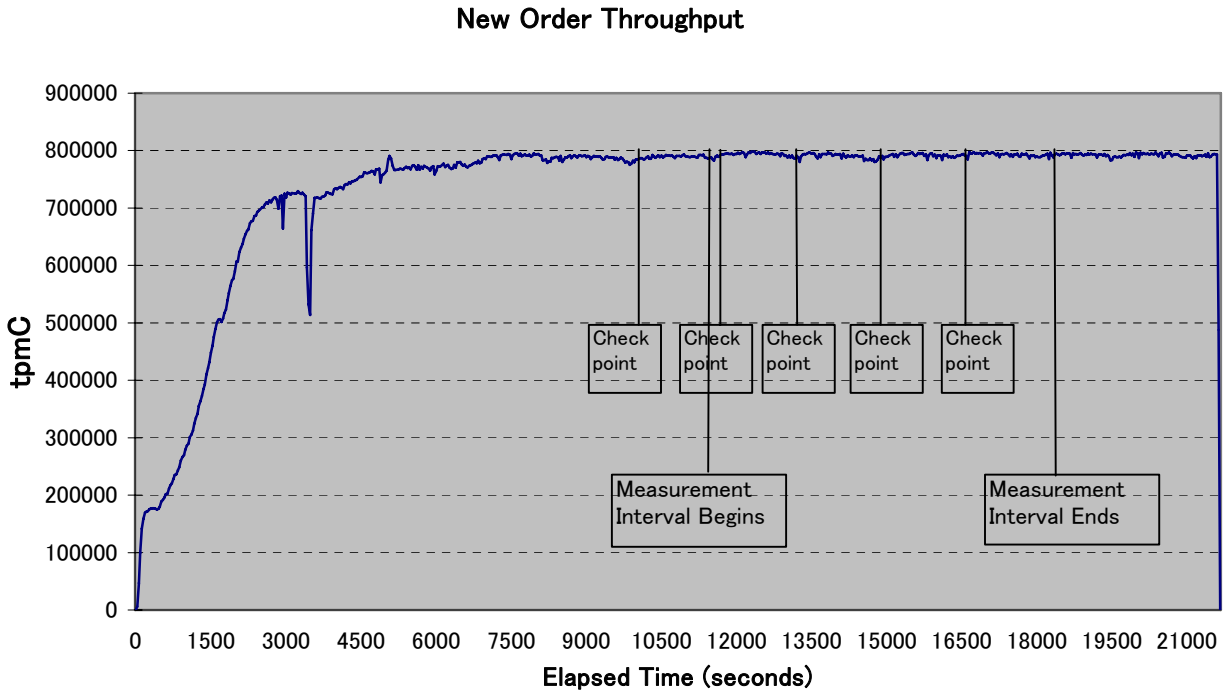


Figure 5.8: New Order Sustained Throughput



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	44.94%
	Payment	43.02%
	Order status	4.01%
	Delivery	4.01%
	Stock level	4.02%
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse	85.00%
	Remote warehouse	15.00%
	Accessed by last name	60.00%
Order Status	Accessed by last name	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	16:55:32	18:55:32	120 minutes
	start	End	duration
checkpoint 0	16:27:36	16:54:35	26:59
checkpoint 1	16:55:38	17:22:39	27:01
checkpoint 2	17:23:59	17:51:00	27:01
checkpoint 3	17:52:05	18:19:06	27:01
checkpoint 4	18:20:23	18:47:24	27:01

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.

Three 1Gbps ethernet LAN connections were used between the server and three switches, to which clients were connected by thirty nine 100Mbps ethernet LAN connections. Thirty nine 100Mbps ethernet LAN connections were used between the client machines 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 October 26, 2006.

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 :	792,101.96 tpmC
Price per tpmC :	$8.92 per tpmC
Three-year cost of ownership :	$7,060,571 USD

7.4 Country Specific Pricing

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

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

7.5 Usage Pricing

For any usage pricing, the sponsor must disclose:

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

The component pricing based on usage is shown below:

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

7.6 System Pricing

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

System pricing must include line item indication where non-sponsoring companies' part numbers are used. System pricing must also include 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 E at the end of this document.

Clause 8 Related Items

8.1 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.

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

::::::::::::::::::
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/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] = '-';
!
!   /* Q */
```

```
!   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/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/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/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/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);

// G[y[W\OracleASymfogp
// 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 : DLLeXgpf[^f[^mFpTu
//
//   f[^   FSCRTEST#defineL
//   f[^mFFDBPRT#defineL
//

//
// pf[^`
//

#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

//
// f[^mFpTu`
//

#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:  /* L[t */
      case TPETIME:   /* gUNV^CAEg */
      case TPGOTSIG:  /* VOi */
         /* 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]), 5, (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]), 5, (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 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_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 : f[^`FbN (-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 : 1sf[^ */
      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],
5)) < 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 [obNf[^p 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]), 5, (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]), 5, (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]), 5,
(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,
5)) < 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]), 5, (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), 5,
(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");*/

      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\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]), 5, (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 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_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/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,st
ay_Del,stay_Sto,"\

"resp_New,num_New,resp_Pay,num_Pay,resp_O
dr,num_Odr,resp_Del,num_Del,resp_Sto,num_Sto
,"\

"imax_New,imax_Pay,imax_Odr,imax_Del,imax_S
to,"\

"max_New,max_Pay,max_Odr,max_Del,max_Sto,
"\

"ans_New,nas_Pay,ans_Odr,ans_Del,ans_Sto,co
nnect\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 :L */
            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 :L */
            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 :L */
            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.: _P */
        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)wf~^AhXT[`

   Al: f~^AhX.Af~^NULL
         f~^ANULL

   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 */
    /* AhX */
//    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);   /* f[^AhX */
}

/*
    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 :L */
            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 */

    /* lenl*/
    /* 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/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, 0x3b }; /* w_id, carrier
number, status */

::::::::::::::::::::
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"OracleSymfoX(set_errHTML)B
!   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 : bZ[WeX */
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:

  RTEwf[^G[tH[

  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 : bZ[WeXC
 */
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 */
TPCONTEXT_T         GLB_TpContext = 0;

::::::::::::::::::::
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 */
extern  TPCONTEXT_T         GLB_TpContext;

#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 "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                            *

***********************************************************
**/
THREAD_CNTL_INFO* CreateThreadEnv() {

    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(GLB
_ThreadKey)) == NULL) {

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

        /* First execution in this thread */
#ifndef SCRTEST
        /* Regist context */
        if (tpsetctxt(GLB_TpContext, 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(GLB
_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                                 *

*************************************************************
**/
TPCONTEXT_T RegistTuxApl() {

  TPCONTEXT_T ctx = 0;

  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 (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;
  }

  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/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/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/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/Maketpapl
::::::::::::::::::::
#! /bin/sh
#-----------------------------------------------------------------------
----
#  Make TPAPL for Linux.
#
#   Created by TSL 2003.12.22
```

```
#
#   All Right Rserverd, Copyright Co, FUJITSU
LIMITED 2003-2004.
#--------------------------------------------------------------
----
TARGETDIR=/home/tpc/client_apl
MAKETXT=${TARGETDIR}/tpapl/make.txt
TPAPLDIR=${TARGETDIR}/tpapl
TRSEXDIR=${TARGETDIR}/tpapl/trnexe
COMDIR=${TARGETDIR}/common

#
# Delete objests , libralys and others
#
if [ "$1" = "clean" ] ; then
   cd ${TPAPLDIR}
   make clean
   cd ${TRSEXDIR}
   make -f Makefile_lib clean
   make clean
   exit
fi

if [ $# -gt 0 ]; then
  echo
"#####################################
#########################"  >  ${MAKETXT}
  echo "##################  begin to make all
TPAPL ###################"  >> ${MAKETXT}
  echo
"#####################################
#########################"  >> ${MAKETXT}
else
  echo
"#####################################
#########################"
  echo "##################  begin to make all
TPAPL ###################"
  echo
"#####################################
#########################"
fi
#====================================
====================
# make a common library for client applications.
#====================================
====================
cd ${COMDIR}
if [ $# -gt 0 ]; then
  echo "-----------------------------------------------------
-------" >> ${MAKETXT}
  echo "---------------- begin to make a common
library ----------------" >> ${MAKETXT}
  echo "-----------------------------------------------------
-------" >> ${MAKETXT}
  make >> ${MAKETXT} 2>&1
else
  echo "-----------------------------------------------------
-------"
  echo "---------------- begin to make a common
library ----------------"
  echo "-----------------------------------------------------
-------"
  make
fi
#====================================
====================
# make TPAPL main library
#====================================
====================
  cd ${TPAPLDIR}
if [ $# -gt 0 ]; then
  echo "-----------------------------------------------------
-------" >> ${MAKETXT}
  echo "---------------- begin to make a tpapl library
----------------" >> ${MAKETXT}
  echo "-----------------------------------------------------
-------" >> ${MAKETXT}
  make >> ${MAKETXT} 2>&1
else
  echo "-----------------------------------------------------
-------"
  echo "---------------- begin to make a tpapl library
----------------"
  echo "-----------------------------------------------------
-------"
  make
fi
#====================================
====================
# make TPAPL for WARE HOUSE BIND
#====================================
====================
  cd ${TRSEXDIR}
  make clean
if [ $# -gt 0 ]; then
  echo "-----------------------------------------------------
-------" >> ${MAKETXT}
  echo "---------- begin to make a archive for
WARE HOUSE BIND ---------" >> ${MAKETXT}
  echo "-----------------------------------------------------
-------" >> ${MAKETXT}
  make BIND_TYPE="WH_BIND" >>
${MAKETXT} 2>&1
else
  echo "-----------------------------------------------------
-------"
  echo "---------- begin to make a archive for
WARE HOUSE BIND ---------"
  echo "-----------------------------------------------------
-------"
  make BIND_TYPE="WH_BIND"
fi
  cd ${TPAPLDIR}
if [ $# -gt 0 ]; then
  echo "-----------------------------------------------------
-------" >> ${MAKETXT}
  echo "------------ begin to make WARE HOUSE
BIND TPAPL -------------" >> ${MAKETXT}
  echo "-----------------------------------------------------
-------" >> ${MAKETXT}
  make >> ${MAKETXT} 2>&1
  mv mod_tpapl.so mod_tpapl.so.WH >>
${MAKETXT}
else
  echo "-----------------------------------------------------
-------"
  echo "---------- begin to make a archive for
WARE HOUSE BIND ---------"
  echo "-----------------------------------------------------
-------"
  make
  mv mod_tpapl.so mod_tpapl.so.WH
fi
#====================================
====================
# make TPAPL for TRANSACTION BIND
#====================================
====================
  cd ${TRSEXDIR}
  make clean
if [ $# -gt 0 ]; then
  echo "-----------------------------------------------------
-------" >> ${MAKETXT}
  echo "---------- begin to make a archive for
WARE HOUSE BIND ---------" >> ${MAKETXT}
  echo "-----------------------------------------------------
-------" >> ${MAKETXT}
  make BIND_TYPE="TRNS_BIND" >>
${MAKETXT} 2>&1
else
  echo "-----------------------------------------------------
-------"
  echo "---------- begin to make a archive for
WARE HOUSE BIND ---------"
  echo "-----------------------------------------------------
-------"
  make BIND_TYPE="TRNS_BIND"
fi
  cd ${TPAPLDIR}
if [ $# -gt 0 ]; then
  echo "-----------------------------------------------------
-------" >> ${MAKETXT}
  echo "------------ begin to make WARE HOUSE
BIND TPAPL -------------" >> ${MAKETXT}
  echo "-----------------------------------------------------
-------" >> ${MAKETXT}
  make >> ${MAKETXT} 2>&1
  mv mod_tpapl.so mod_tpapl.so.TR >>
${MAKETXT}
else
  echo "-----------------------------------------------------
-------"
  echo "---------- begin to make a archive for
WARE HOUSE BIND ---------"
  echo "-----------------------------------------------------
-------"
  make
  mv mod_tpapl.so mod_tpapl.so.TR
fi
if [ $# -gt 0 ]; then
  echo
"#####################################
#########################"  >  ${MAKETXT}
  echo "##################  end to make all
TPAPL ###################"  >>
${MAKETXT}
  echo
"#####################################
#########################"  >> ${MAKETXT}
else
  echo
"#####################################
#########################"
  echo "##################  end to make all
TPAPL ###################"
  echo
"#####################################
#########################"
fi


::::::::::::::::::::
::::::::::::::::::::
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\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/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/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/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\
                    $    .        -  -   \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, 0x157,
0x163, 0x174, 0x177, 0x194, /* 18 - 21 */
0x1a8, 0x1d9,          /* 22, 23 */
0x1e5, 0x216,          /* 24, 25 */
0x225, 0x23a, 0x23d, 0x256,
0x284, 0x2a4,
0x2c5,
0x2e1,            /* offset 0x3e */
0x320,
0x35f,
0x39e};

::::::::::::::::::::
::::::::::::::::::::
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_SHM
```

```
PATH, 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_SH
MPATH,
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/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"

/* 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/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/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/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/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 :
  QueryStringl
  YlAhXp\

  Al: v[
```

```
   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:

   RTEwgUNVs
   Al: I[m

   s_bufgUNVHTMLtH[

   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'.

   L[MNeBJZNV
  ---------------------------------------------------------------- */
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/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 */
    ThreadCntlInfo = CreateThreadEnv();
    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 ((GLB_TpContext = RegistTuxApl()) == 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();
THREAD_CNTL_INFO*   CreateThreadEnv();
void            FreeThreadKey();
void            GetConfFileInfo();
TPCONTEXT_T       RegistTuxApl();
apr_status_t      TermChildProcess(void* p);
apr_status_t      PlainCleanup(void* p);

::::::::::::::::::::
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:%5d\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:%5d\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:%5d   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=5>
<INPUT NAME=\"OI01\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ01\" SIZE=2
maxlength=2>\r\n\
```

```
  <INPUT NAME=\"OS02\" SIZE=5 maxlength=5>
<INPUT NAME=\"OI02\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ02\" SIZE=2
maxlength=2>\r\n\
  <INPUT NAME=\"OS03\" SIZE=5 maxlength=5>
<INPUT NAME=\"OI03\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ03\" SIZE=2
maxlength=2>\r\n\
  <INPUT NAME=\"OS04\" SIZE=5 maxlength=5>
<INPUT NAME=\"OI04\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ04\" SIZE=2
maxlength=2>\r\n\
  <INPUT NAME=\"OS05\" SIZE=5 maxlength=5>
<INPUT NAME=\"OI05\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ05\" SIZE=2
maxlength=2>\r\n\
  <INPUT NAME=\"OS06\" SIZE=5 maxlength=5>
<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=5>
<INPUT NAME=\"OI07\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ07\" SIZE=2
maxlength=2>\r\n\
  <INPUT NAME=\"OS08\" SIZE=5 maxlength=5>
<INPUT NAME=\"OI08\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ08\" SIZE=2
maxlength=2>\r\n\
  <INPUT NAME=\"OS09\" SIZE=5 maxlength=5>
<INPUT NAME=\"OI09\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ09\" SIZE=2
maxlength=2>\r\n\
  <INPUT NAME=\"OS10\" SIZE=5 maxlength=5>
<INPUT NAME=\"OI10\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ10\" SIZE=2
maxlength=2>\r\n\
  <INPUT NAME=\"OS11\" SIZE=5 maxlength=5>
<INPUT NAME=\"OI11\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ11\" SIZE=2
maxlength=2>\r\n\
  <INPUT NAME=\"OS12\" SIZE=5 maxlength=5>
<INPUT NAME=\"OI12\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ12\" SIZE=2
maxlength=2>\r\n\
  <INPUT NAME=\"OS13\" SIZE=5 maxlength=5>
<INPUT NAME=\"OI13\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ13\" SIZE=2
maxlength=2>\r\n\
  <INPUT NAME=\"OS14\" SIZE=5 maxlength=5>
<INPUT NAME=\"OI14\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ14\" SIZE=2
maxlength=2>\r\n\
  <INPUT NAME=\"OS15\" SIZE=5 maxlength=5>
<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:%5d   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:%5d       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:%5d                    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=5>  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:%5d
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:%5d        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"gXB*/
!/* 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;
```

Appendix B: Server Source Code

```
::::::::::::::::::
blocks/load_ordordl.sql
::::::::::::::::::
-- anonymous block for loading order/orderline

DECLARE
  order_idx      PLS_INTEGER;
  order_rows     PLS_INTEGER;
  ordl_rows      PLS_INTEGER;
  ordl_idx       PLS_INTEGER;
  ordl_idx_hi    PLS_INTEGER;
  local_idx      PLS_INTEGER;
BEGIN
  order_rows := :order_rows;
  ordl_rows  := :ordl_rows;
  order_idx  := 1;
  ordl_idx   := 1;

  WHILE (order_idx <= order_rows) LOOP

    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(order_idx), :o_d_id(order_idx), :o_w_id(orde
r_idx),
               :o_c_id(order_idx),
SYSDATE, :o_carrier_id(order_idx),
               :o_ol_cnt(order_idx), 1);

    ordl_idx_hi := ordl_idx + :o_ol_cnt(order_idx) -
1;

    IF ( :o_id(order_idx) < 2101 ) THEN
      FORALL local_idx IN ordl_idx .. ordl_idx_hi
        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(local_idx), :ol_d_id(local_idx),
                    :ol_w_id(local_idx), :ol_number(l
ocal_idx),
                    SYSDATE, :ol_i_id(local_idx),
                    :ol_supply_w_id(local_idx), 5,
0, :ol_dist_info(local_idx));
    ELSE
      FORALL local_idx IN ordl_idx .. ordl_idx_hi
        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(local_idx), :ol_d_id(local_idx),
                    :ol_w_id(local_idx), :ol_number(l
ocal_idx),
                    to_date('01-Jan-
1811'), :ol_i_id(local_idx),
                    :ol_supply_w_id(local_idx), 5,
                    :ol_amount(local_idx), :ol_dist_i
nfo(local_idx));
    END IF;
    ordl_idx := ordl_idx_hi + 1;
    order_idx := order_idx + 1;
  END LOOP;
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;

::::::::::::::::::
::::::::::::::::::
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_st
reet_1,
               :c_street_2, :c_city, :c_state, :c_zip, :c
_phone,
               :c_since, :c_credit, :c_credit_lim,
               :c_discount, :c_balance;
         IF SQL%NOTFOUND THEN
          raise NO_DATA_FOUND;
         END IF;


       IF :c_credit = 'BC' THEN
         UPDATE cust
        SET c_data = substr ((to_char (:c_id) || ' ' ||
                     to_char (:c_d_id) || ' ' ||
                     to_char (:c_w_id) || ' ' ||
                     to_char (:d_id) || ' ' ||
```

```
                        to_char (:w_id) || ' ' ||
                        to_char (:h_amount/100,
'9999.99') || ' ')
                        || c_data, 1, 500)
          WHERE rowid = inittpcc.cust_rowid
        RETURNING substr(c_data,1, 200)
          INTO :c_data;

        END IF;

          UPDATE dist
            SET d_ytd = d_ytd + :h_amount
          WHERE d_id = :d_id
            AND d_w_id = :w_id
        RETURNING d_name, d_street_1,
d_street_2, d_city,d_state, d_zip
          INTO
inittpcc.dist_name,:d_street_1,:d_street_2,:d_city,:
d_state,
               :d_zip;
         IF SQL%NOTFOUND THEN
           raise NO_DATA_FOUND;
         END IF;


        INSERT INTO hist  (h_c_id, h_c_d_id,
h_c_w_id, h_d_id, h_w_id,
                    h_amount, h_date, h_data)
        VALUES

(:c_id, :c_d_id, :c_w_id, :d_id, :w_id, :h_amount,
          :cr_date, inittpcc.ware_name || '    ' ||
inittpcc.dist_name);
        EXIT;

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

    END LOOP;
  END;

::::::::::::::::::::
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_s
tate, :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_stat
e,
            :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_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/views.sql
::::::::::::::::::::

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(stok) USE_NL(stok) */ 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
/


::::::::::::::::::::
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;

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

::::::::::::::::::
::::::::::::::::::
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/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/MakeShell
::::::::::::::::::::
#! /bin/sh
cd /home/tpc/client_apl/svrapl/3tier
make > make_result.txt 2>&1

::::::::::::::::::::
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_DEFA
ULT);
    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_DEFA
ULT);
    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_DEFA
ULT);
    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,(dvoid
**)(&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_DEFA
ULT);
    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_INT
);
  OCIDEF(octx->curo1,octx-
>o_entry_d_dp[0],errhp,7,

&o_entry_d_base,SIZ(OCIDate),SQLT_ODT);
  OCIDEF(octx->curo1,octx-
>o_cr_id_dp[0],errhp,8,ADR(o_carrier_id),
          SIZ(int),SQLT_INT);
  OCIDEF(octx->curo1,octx-
>o_ol_cnt_dp[0],errhp,9,ADR(o_ol_cnt),
          SIZ(int),SQLT_INT);

/* 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_INT
);
  OCIDEF(octx->curo2,octx-
>o_entry_d_dp[1],errhp,6, &o_entry_d_base,
          SIZ(OCIDate),SQLT_ODT);
  OCIDEF(octx->curo2, octx-
>o_cr_id_dp[1],errhp,7,ADR(o_carrier_id),
          SIZ(int), SQLT_INT);
  OCIDEF(octx->curo2,octx-
>o_ol_cnt_dp[1],errhp,8,ADR(o_ol_cnt),
          SIZ(int),SQLT_INT);

/* 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_DEFA
ULT);
    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_DEFA
ULT);
    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_DEFA
ULT);
    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_ba
se[i],
         (const
text*)SHORTDATE,(ub1)strlen(SHORTDATE),(tex
t*)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_DEFA
ULT));
  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",oracl
e_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,(dvoid
*)0,0,0,0);
!#endif
!
OCIInitialize(OCI_THREADED|OCI_OBJECT,(dvo
id *)0,0,0,0);
!/* Replaced end */
#endif


OCIInitialize(OCI_DEFAULT|OCI_OBJECT,(dvoid
*)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_dat
e));

  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]={0,1,2,3,4,5,6,7,8,9,10,
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_dat
e));

   if ((str->newout.terror = tkvcn ()) != 0) {
    if (str->newout.terror != RECOVERR)
      str->newout.terror = IRRECERR;
    return (-1);
  }

  /* fill in date for o_entry_d from time in beginning
of txn*/
/*
  cvtdmyhms(cr_date,o_entry_d);
*/
   datelen = sizeof(o_entry_d);
   OCIERROR(errhp,

OCIDateToText(errhp,&cr_date,(text*)FULLDATE,
SIZ(FULLDATE),(text*)0,0,
                &datelen,o_entry_d));

  str->newout.terror = NOERR;
  str->newout.o_id = o_id;
  str->newout.o_ol_cnt = o_ol_cnt;
  strncpy (str->newout.c_last, c_last, 17);
  strncpy (str->newout.c_credit, c_credit, 3);
  str->newout.c_discount = c_discount;
  str->newout.w_tax = (float)(w_tax);
  str->newout.d_tax = (float)(d_tax);
  strncpy (str->newout.o_entry_d,
(char*)o_entry_d, 20);
/* Replaced T.Kato 02.11.13 */
#if 0
!  str->newout.total_amount = total_amount;
#endif
  str->newout.total_amount = 0.0;
/* Replaced end */
  for (i = 0; i < o_ol_cnt; i++) {
    strncpy (str->newout.i_name[i], i_name[i], 25);
    str->newout.brand_generic[i] =
brand_generic[i][0];

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
!    str->newout.s_quantity[i] = s_quantity[i];
!    str->newout.i_price[i] = (float)(i_price[i])/100;
!    str->newout.ol_amount[i] =
(float)(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_dat
e));

   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_date,

(text*)FULLDATE,(ub1)strlen(FULLDATE),(text*)0,
0,&hlen,h_date));

/*
   cvtdmy(c_since,c_since_d);
*/
       sincelen=SIZ(c_since_d);

OCIERROR(errhp,OCIDateToText(errhp,&c_since,

(text*)SHORTDATE,(ub1)strlen(SHORTDATE),(te
xt*)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*)FU
LLDATE,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_dat
e));
#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)s
tr->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


::::::::::::::::::::
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/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)       */
/* ---------------------------- */
payctx *pctx;
/* ---------------------------- */
/*   StockLevel (plsto.cpp)     */
/* ---------------------------- */
stoctx *sctx;
/* ---------------------------- */
/*    (tpccpl.cpp)              */
/* ---------------------------- */
FILE *lfp;
/* Deleted T.Kato 02.10.23 for warning
!FILE *fopen ();
   Deleted end */

/* Added t.Kato 02.10.24 for Delivery logging file
control */
int iflg;   /* Delivery log initialize flag */
/* Added end */
int proc_no;
int logon;
int new_init;
int pay_init;
int ord_init;

#ifdef DEL_ORA8I
int del_init;
#else
int del_init_oci;
int del_init_plsql;
#endif

int sto_init;
int res_init;

int execstatus;
int errcode;

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


/* for stock-level transaction */
int w_id;
int d_id;
int c_id;

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
!   int threshold;
#endif

#ifdef USE_IEEE_NUMBER
float threshold;
#else
int threshold;
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

int low_stock;

/* for delivery transaction */
int del_o_id[10];
int retries;

/* for order-status transaction */
int bylastname;
char c_last[17];
char c_first[17];
char c_middle[3];
double c_balance;
int o_id;
text o_entry_d[20];
ub4  datelen;
int o_carrier_id;
int o_ol_cnt;
int ol_supply_w_id[15];
int ol_i_id[15];

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
!   int ol_quantity[15];
!   int ol_amount[15];
#endif

#ifdef USE_IEEE_NUMBER
float ol_quantity[15];
float ol_amount[15];
#else
int ol_quantity[15];
int ol_amount[15];
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

ub4  ol_del_len[15];
text ol_delivery_d[15][11];
/* xnie - begin */
OCIRowid *o_rowid;
/* xnie - end */

/* for payment transaction */
int c_w_id;
int c_d_id;

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
!   int h_amount;
#endif

#ifdef USE_IEEE_NUMBER
float h_amount;
#else
int h_amount;
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

char w_street_1[21];
char w_street_2[21];
char w_city[21];
char w_state[3];
```

```
char w_zip[10];
char d_street_1[21];
char d_street_2[21];
char d_city[21];
char d_state[3];
char d_zip[10];
char c_street_1[21];
char c_street_2[21];
char c_city[21];
char c_state[3];
char c_zip[10];
char c_phone[17];
ub4  sincelen;
text c_since_d[11];
float c_discount;
char c_credit[3];
int c_credit_lim;
char c_data[201];
ub4 hlen;
text h_date[20];

/* for new order transaction */

int nol_i_id[15];
int nol_supply_w_id[15];

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
!   int nol_quantity[15];
!   int nol_amount[15];
!   int s_quantity[15];
!   int i_price[15];
#endif

#ifdef USE_IEEE_NUMBER
float nol_quantity[15];
float nol_amount[15];
float s_quantity[15];
float i_price[15];
#else
int nol_quantity[15];
int nol_amount[15];
int s_quantity[15];
int i_price[15];
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

int nol_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/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/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  = -DTSL -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)/tpccfmlw
TARGET_NEW_3TIER       =
$(OBJDIR)/tpccfmln
TARGET_PAY_3TIER       =  $(OBJDIR)/tpccfmlp
TARGET_DEL_3TIER       =  $(OBJDIR)/tpccfmld
TARGET_STO_3TIER       =  $(OBJDIR)/tpccfmls
TARGET_ORD_3TIER       =  $(OBJDIR)/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/MakeShell
::::::::::::::::::::
#! /bin/sh
cd /home/tpc/client_apl/svrapl
make > make_result.txt 2>&1

::::::::::::::::::::
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_HT
YPE_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,fty
pe,indp,alen,arcode) \
     DISCARD ocierror(LOG_FILE_LINE,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HT
YPE_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,O
CI_DEFAULT));


/* use with callback data */
#define
OCIBNDRAD(stmp,bndp,errp,sqlvar,progvl,ftype,in
dp,ctxp,\
               cbf_nodata,cbf_data) \
     DISCARD ocierror(LOG_FILE_LINE,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HT
YPE_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_nodata)
,(ctxp),(cbf_data)));



/* bind in/out for plsql without indicator and rcode */
#define
OCIBNDPL(stmp,bndp,errp,sqlvar,progv,progvl,fty
pe,alen) \
   DISCARD ocierror(LOG_FILE_LINE,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HT
YPE_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,ftyp
e,indp,alen,arcode) \
     DISCARD ocierror(LOG_FILE_LINE,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HT
YPE_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,ft
ype,alen,ms,cu) \
     DISCARD ocierror(LOG_FILE_LINE, (errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HT
YPE_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,f
type,indp,alen,arcode,\
             ms,cu) \
     ocierror(LOG_FILE_LINE, (errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HT
YPE_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,ftype
)\

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


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

OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTY
PE_DEFINE,0,\
               (dvoid**)0);\

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(prog
v),(progvl),\

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


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

OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTY
PE_DEFINE,0,\
               (dvoid**)0);\

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

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

OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTY
PE_DEFINE,0,\
               (dvoid**)0));\
     ocierror(LOG_FILE_LINE,(errp), \

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(prog
v), (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/tpccapi.h
::::::::::::::::::::
#ifndef TPCCAPI_H
#define TPCCAPI_H

/*=====================================
============================+
|      Copyright (c) 1997  Oracle Corp, Redwood
Shores, CA      |
|                OPEN SYSTEMS PERFORMANCE
GROUP                |
|                   All Rights Reserved                    |

+=====================================
============================+
| FILENAME
|   tpccapi.h
| DESCRIPTION
|   header file to tpcc.dll

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

#define BOOL    unsigned char

//VERSION RESOURCE DEFINES
#define _APS_NEXT_RESOURCE_VALUE
101
#define _APS_NEXT_COMMAND_VALUE
40001
#define _APS_NEXT_CONTROL_VALUE
1000
#define _APS_NEXT_SYMED_VALUE          101

    //note that the welcome form must be
processed first as terminal ids
    //assigned here, once the terminal id is
assigned then the forms can
    //be processed in any order.

#define WELCOME_FORM    1
        //beginning form no term id assigned, form
id
#define MAIN_MENU_FORM  2
        //term id assigned main menu form id
#define NEW_ORDER_FORM  3
        //new order form id
#define PAYMENT_FORM    4
        //payment form id
#define DELIVERY_FORM   5
        //delivery form id
#define ORDER_STATUS_FORM       6
        //order status id
#define STOCK_LEVEL_FORM        7
        //stock level form id

    //This macro is used to prevent the compiler
error unused formal parameter
#define UNUSEDPARAM(x) (x = x)

    //error message structure used in
ErrorMessage API
typedef struct _SERRORMSG
{
    int     iError;             //error id of message
    char    szMsg[80];          //message to sent
to browser
} SERRORMSG;

   //This structure defines the data necessary to
keep distinct for each
   //terminal or client connection.
typedef struct _CLIENTDATA
{
    int     inUse;          //in use flag allows client
entries to be reused
    int     w_id;           //warehouse id assigned at
welcome form
    int     d_id;           //district id assigned at
welcome form
    int     iProcessRequestCounter;         //request
counter
    int     iSyncId;        //syncronization id
    int     iTickCount;     //time of last access;
    int     iTermId;        //terminal id of http stream
connection
    char    szBuffer[4096]; //form buffer; HTML
form built for client here
    //for TOPEND
    char    type;
    int     retval;
    int     client;
    BOOL    bFailed;
#ifdef TOPEND
    long    *tp_structadr;
#else // assume tuxedo
    struct newstruct        *newOrderDataPtr;
//new order Tuxedo buffer
    struct paystruct        *paymentDataPtr;
//payment Tuxedo buffer
    struct ordstruct        *orderStatusDataPtr;
//order status Tuxedo buffer
    struct delstruct        *deliveryDataPtr;
//delivery Tuxedo buffer
    struct stostruct        *stockLevelDataPtr;
//stock level Tuxedo buffer
#endif
    union {
    struct newstruct        newOrderData;   //new
order form data
    struct paystruct        paymentData;
//payment form data
    struct ordstruct  orderStatusData;      //order
status form data
    struct delstruct        deliveryData;   //delivery
form data
    struct stostruct        stocklevelData; //stock
level form data
    } trans;
} CLIENTDATA;

typedef CLIENTDATA *PCLIENTDATA;
//pointer to client structure
   //This structure is used to define the operational
interface for
   //terminal id support
typedef struct _TERM
{
int     iAvailable;     //total allocated terminal array
entries
int     iNext;          //next available terminal array
element
int     iMasterSyncId;  //syncronization id
BOOL    bInit;          //structure has been initialized
flag
CLIENTDATA      *pClientData;   //pointer to
allocated client data
void    (*Init)(void);  //API to initialize this structure
int     (*Allocate)(void); //API to allocate new
terminal entry;array id returnd
```

```
void    (*Restore)(void);       //API to free terminal
data
int     (*Add)(EXTENSION_CONTROL_BLOCK
*pECB,
    char *pQueryString);       //API to add a terminal
id to array,
                //this context will be passed from the
                //browser to the tpcc.dll in the
                //TERMID= key in the HTTP string.
void    (*Delete)(EXTENSION_CONTROL_BLOCK
*pECB, int id); //API to free
                        //resources used by a
                        //terminal array
                        //entry
} TERM;

typedef TERM *PTERM;    //pointer to terminal
structure type

    //this structure allows the EXTENSION
CONTROL BLOCK to be passed to the
    //msg and error handlers.
typedef struct _ECBINFO
{
    int      iTermId;       //terminal id
    int      iSyncId;       //browser sync id
    BOOL     bDeadlock;     //deadlock condition
flag
    BOOL     bFailed;       //cleared before sql
transaction, set in err
            //handlers if an error occurs
    EXTENSION_CONTROL_BLOCK *pECB;
//inetsrv current connection structure
                //information
} ECBINFO, *PECBINFO;

//function prototypes

#ifdef TOPEND // in tuxedo defined in atmi.h
extern int tpsvrinit (int argc, char * argv[]);
        // argv arguments are  int pid, char *uid, char
*pwd, int txntype

extern int NEWORDER(CLIENTDATA *jobData,
NewOrderData *neword, int deadlock);
extern int PAYMENT(CLIENTDATA *jobData,
PaymentData *paydata, int deadlock);
extern int ORDERSTATUS(CLIENTDATA
*jobData, OrderStatusData *orddata, int deadlock);
extern int STOCKLEVEL(CLIENTDATA *jobData,
StockLevelData *stodata, int deadlock);
extern int DELIVERY(CLIENTDATA *jobData,
DeliveryData *deldata, int deadlock);
#endif

#ifdef TUX
extern void NEWORDER(TPSVCINFO *msg);
extern void PAYMENT(TPSVCINFO *msg);
extern void ORDERSTATUS(TPSVCINFO *msg);
extern void STOCKLEVEL(TPSVCINFO *msg);
extern void DELIVERY(TPSVCINFO *msg);
#endif


BOOL APIENTRY DllMain(HANDLE hModule,
DWORD ul_reason_for_call, LPVOID lpReserved);
static BOOL IsValidTermId(int TermId);
BOOL
ProcessQueryString(EXTENSION_CONTROL_BL
OCK *pECB, int *pCmd, int *pFormId, int *pTermId,
int *pSyncId);
void
NewOrderForm(EXTENSION_CONTROL_BLOCK
*pECB, int iFormId, int iTermId, int iSyncId);
void
PaymentForm(EXTENSION_CONTROL_BLOCK
*pECB, int iFormId, int iTermId, int iSyncId);
void
DeliveryForm(EXTENSION_CONTROL_BLOCK
*pECB, int iFormId, int iTermId, int iSyncId);
void
OrderStatusForm(EXTENSION_CONTROL_BLO
CK *pECB, int iFormId, int iTermId, int iSyncId);
void
StockLevelForm(EXTENSION_CONTROL_BLOC
K *pECB, int iFormId, int iTermId, int iSyncId);
void Exitcmd(EXTENSION_CONTROL_BLOCK
*pECB, int iFormId, int iTermId, int iSyncId);
void
SubmitCmd(EXTENSION_CONTROL_BLOCK
*pECB, int iFormId, int iTermId, int iSyncId);
void BeginCmd(EXTENSION_CONTROL_BLOCK
*pECB, int iFormId, int iTermId, int iSyncId);
void
ProcessCmd(EXTENSION_CONTROL_BLOCK
*pECB, int iFormId, int iTermId, int iSyncId);
void ClearCmd(EXTENSION_CONTROL_BLOCK
*pECB, int iFormId, int iTermId, int iSyncId);
void MenuCmd(EXTENSION_CONTROL_BLOCK
*pECB, int iFormId, int iTermId, int iSyncId);
static void
WriteZString(EXTENSION_CONTROL_BLOCK
*pECB, char *szStr);
static void
h_printf(EXTENSION_CONTROL_BLOCK *pECB,
char *format, ...);
void
ErrorMessage(EXTENSION_CONTROL_BLOCK
*pECB, int iError, int iErrorType, char *szMsg, int
iTermId, int iSyncId);
static BOOL GetKeyValue(char *pQueryString,
char *pKey, char *pValue, int iMax);
static void TermInit(void);
static void TermRestore(void);
static int TermAllocate(void);
static int
TermAdd(EXTENSION_CONTROL_BLOCK
*pECB, char *pQueryString);
static void
TermDelete(EXTENSION_CONTROL_BLOCK
*pECB, int id);
BOOL Init(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, int iSyncId, char *szServer,
char *szUser, char *szPassword, char
*szDatabase);
static BOOL
Close(EXTENSION_CONTROL_BLOCK *pECB,
int iTermId, int iSyncId);
static void FormatString(char *szDest, char *szPic,
char *szSrc);
static char *MakeStockLevelForm(int iTermId, int
iSyncId, BOOL bInput);
static char *MakeMainMenuForm(int iTermId, int
iSyncId);
static char *MakeWelcomeForm(void);
static char *MakeNewOrderForm(int iTermId, int
iSyncId, BOOL bInput, BOOL bValid, char
*execution_status);
static char *MakePaymentForm(int iTermId, int
iSyncId, BOOL bInput);
static char *MakeOrderStatusForm(int iTermId, int
iSyncId, BOOL bInput);
static char *MakeDeliveryForm(int iTermId, int
iSyncId, BOOL bInput, char *execution_status);
void UtilStrCpy(char * pDest, char * pSrc, int n);
static void
ProcessNewOrderForm(EXTENSION_CONTROL
_BLOCK *pECB, int iTermId, int iSyncId);
static void
ProcessPaymentForm(EXTENSION_CONTROL_
BLOCK *pECB, int iTermId, int iSyncId);
static void
ProcessOrderStatusForm(EXTENSION_CONTRO
L_BLOCK *pECB, int iTermId, int iSyncId);
static void
ProcessDeliveryForm(EXTENSION_CONTROL_B
LOCK *pECB, int iTermId, int iSyncId);
static void
ProcessStockLevelForm(EXTENSION_CONTROL
_BLOCK *pECB, int iTermId, int iSyncId);
static int GetNewOrderData(LPSTR
lpszQueryString, NewOrderData
*pNewOrderData);
static int GetPaymentData(LPSTR
lpszQueryString, PaymentData *pPaymentData);
static int GetOrderStatusData(LPSTR
lpszQueryString, OrderStatusData
*pOrderStatusData);
static BOOL ReadRegistrySettings(void);
static BOOL IsNumeric(char *ptr);
static void FormatHTMLString(char *szBuff, char
*szStr, int iLen);
static void Log(char *szType, char *szStr);
int do_neworder(NewOrderData *pNOData);
int do_payment(PaymentData *pPAYData);
int do_orderstatus(OrderStatusData *pOSData);
int do_delivery(pDeliveryData pDelivery);
int do_stocklevel(StockLevelData *pSTOData);
int end_neworder(NewOrderData *pNOData);
int end_payment(PaymentData *pPAYData);
int end_orderstatus(OrderStatusData *pOSData);
int end_stocklevel(StockLevelData *pSTOData);
int do_all_txns(int txn);
int do_disconnect( );


#endif /* TPCCAPI_H */


::::::::::::::::::::
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/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/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;
```

Appendix C: RTE Scripts

```
::::::::::::::::
rte11.conf
::::::::::::::::
#
#       rte11.conf :configuration file for TPC-C
Rev3.0
#       Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
        STARTRTE
                STARTSUT
                        SUTHOST = cl033a
                        SUTPORT = 80
                        SUTTERM = 1346
                        LOGPATH = /w00
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl033a
                        SUTPORT = 80
                        SUTTERM = 1346
                        LOGPATH = /w01
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl033a
                        SUTPORT = 80
                        SUTTERM = 1346
                        LOGPATH = /w02
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl033a
                        SUTPORT = 80
                        SUTTERM = 1346
                        LOGPATH = /w03
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl033a
                        SUTPORT = 80
                        SUTTERM = 1346
                        LOGPATH = /w04
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl033a
                        SUTPORT = 80
                        SUTTERM = 1346
                        LOGPATH = /w05
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl033a
                        SUTPORT = 80
                        SUTTERM = 4
                        LOGPATH = /w06
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl033b
                        SUTPORT = 80
                        SUTTERM = 1342
                        LOGPATH = /w06
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl033b
                        SUTPORT = 80
                        SUTTERM = 1346
                        LOGPATH = /w07
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl033b
                        SUTPORT = 80
                        SUTTERM = 1346
                        LOGPATH = /w08
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl033b
                        SUTPORT = 80
                        SUTTERM = 1346
                        LOGPATH = /w09
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl033b
                        SUTPORT = 80
                        SUTTERM = 1346
                        LOGPATH = /w10
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl033b
                        SUTPORT = 80
                        SUTTERM = 1354
                        LOGPATH = /w11
                        LOGLEVEL = 0
                ENDSUT
        ENDRTE
        STARTVARIABLE
                WAREHOUSE = 63024
                MEASUREMENT = 21600
                PAY-MIX = 4302
                ORD-MIX = 402
                DEL-MIX = 402
                STK-MIX = 402
                NEW-KEYING = 18010
                PAY-KEYING = 3010
                ORD-KEYING = 2010
                DEL-KEYING = 2010
                STK-KEYING = 2010
                NEW-THINK = 12020
                PAY-THINK = 12020
                ORD-THINK = 10020
                DEL-THINK = 5020
                STK-THINK = 5020
                CONST-CLAST = 111
                CONST-CID = 1023
                CONST-IID = 8191
                THR-PER-PROC = 250
                SYNC = 0
        ENDVARIABLE
ENDGROUP

::::::::::::::::
rte12.conf
::::::::::::::::
#
#       rte12.conf :configuration file for TPC-C
Rev3.0
#       Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
        STARTRTE
                STARTSUT
                        SUTHOST = cl035a
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w00
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl035a
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w01
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl035a
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w02
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl035a
                        SUTPORT = 80
                        SUTTERM = 1
                        LOGPATH = /w03
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl035b
                        SUTPORT = 80
                        SUTTERM = 2692
                        LOGPATH = /w03
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl035b
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w04
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl035b
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w05
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl035b
                        SUTPORT = 80
                        SUTTERM = 2
                        LOGPATH = /w06
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl036a
                        SUTPORT = 80
                        SUTTERM = 2691
                        LOGPATH = /w06
                        LOGLEVEL = 0
                ENDSUT
```

```
        STARTSUT
            SUTHOST = cl036a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w07
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl036a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w08
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl036a
            SUTPORT = 80
            SUTTERM = 3
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl036b
            SUTPORT = 80
            SUTTERM = 2690
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl036b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w10
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl036b
            SUTPORT = 80
            SUTTERM = 2697
            LOGPATH = /w11
            LOGLEVEL = 0
        ENDSUT
    ENDRTE
    STARTVARIABLE
        WAREHOUSE = 63024
        MEASUREMENT = 21600
        PAY-MIX = 4302
        ORD-MIX = 402
        DEL-MIX = 402
        STK-MIX = 402
        NEW-KEYING = 18010
        PAY-KEYING = 3010
        ORD-KEYING = 2010
        DEL-KEYING = 2010
        STK-KEYING = 2010
        NEW-THINK = 12020
        PAY-THINK = 12020
        ORD-THINK = 10020
        DEL-THINK = 5020
        STK-THINK = 5020
        CONST-CLAST = 111
        CONST-CID = 1023
        CONST-IID = 8191
        THR-PER-PROC = 250
        SYNC = 0
    ENDVARIABLE
ENDGROUP

::::::::::::::::
rte13.conf
```

```
::::::::::::::::
#
#       rte13.conf :configuration file for TPC-C
Rev3.0
#       Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
    STARTRTE
        STARTSUT
            SUTHOST = cl037a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w00
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl037a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w01
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl037a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w02
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl037a
            SUTPORT = 80
            SUTTERM = 1
            LOGPATH = /w03
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl037b
            SUTPORT = 80
            SUTTERM = 2692
            LOGPATH = /w03
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl037b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w04
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl037b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w05
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl037b
            SUTPORT = 80
            SUTTERM = 2
            LOGPATH = /w06
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl038a
            SUTPORT = 80
            SUTTERM = 2691
            LOGPATH = /w06
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl038a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w07
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl038a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w08
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl038a
            SUTPORT = 80
            SUTTERM = 3
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl038b
            SUTPORT = 80
            SUTTERM = 2690
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl038b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w10
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl038b
            SUTPORT = 80
            SUTTERM = 2697
            LOGPATH = /w11
            LOGLEVEL = 0
        ENDSUT
    ENDRTE
    STARTVARIABLE
        WAREHOUSE = 63024
        MEASUREMENT = 21600
        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 = 2693
                        LOGPATH = /w00
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl039a
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w01
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl039a
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w02
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl039a
                        SUTPORT = 80
                        SUTTERM = 1
                        LOGPATH = /w03
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl039b
                        SUTPORT = 80
                        SUTTERM = 2692
                        LOGPATH = /w03
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl039b
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w04
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl039b
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w05
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl039b
                        SUTPORT = 80
                        SUTTERM = 2
                        LOGPATH = /w06
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl040a
                        SUTPORT = 80
                        SUTTERM = 2691
                        LOGPATH = /w06
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl040a
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w07
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl040a
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w08
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl040a
                        SUTPORT = 80
                        SUTTERM = 3
                        LOGPATH = /w09
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl040b
                        SUTPORT = 80
                        SUTTERM = 2690
                        LOGPATH = /w09
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl040b
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w10
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl040b
                        SUTPORT = 80
                        SUTTERM = 2697
                        LOGPATH = /w11
                        LOGLEVEL = 0
                ENDSUT
        ENDRTE
        STARTVARIABLE
                WAREHOUSE = 63024
                MEASUREMENT = 21600
                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 = cl105a
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w00
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl105a
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w01
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl105a
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w02
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl105a
                        SUTPORT = 80
                        SUTTERM = 1
                        LOGPATH = /w03
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl105b
                        SUTPORT = 80
                        SUTTERM = 2692
                        LOGPATH = /w03
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl105b
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w04
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl105b
                        SUTPORT = 80
                        SUTTERM = 2693
                        LOGPATH = /w05
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl105b
                        SUTPORT = 80
                        SUTTERM = 2
                        LOGPATH = /w06
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
```

```
            SUTHOST = cl106a
            SUTPORT = 80
            SUTTERM = 2691
            LOGPATH = /w06
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl106a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w07
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl106a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w08
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl106a
            SUTPORT = 80
            SUTTERM = 3
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl106b
            SUTPORT = 80
            SUTTERM = 2690
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl106b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w10
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl106b
            SUTPORT = 80
            SUTTERM = 2697
            LOGPATH = /w11
            LOGLEVEL = 0
        ENDSUT
    ENDRTE
    STARTVARIABLE
        WAREHOUSE = 63024
        MEASUREMENT = 21600
        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 = cl107a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w00
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl107a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w01
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl107a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w02
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl107a
                SUTPORT = 80
                SUTTERM = 1
                LOGPATH = /w03
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl107b
                SUTPORT = 80
                SUTTERM = 2692
                LOGPATH = /w03
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl107b
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w04
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl107b
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w05
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl107b
                SUTPORT = 80
                SUTTERM = 2
                LOGPATH = /w06
                LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl108a
            SUTPORT = 80
            SUTTERM = 2691
            LOGPATH = /w06
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl108a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w07
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl108a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w08
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl108a
            SUTPORT = 80
            SUTTERM = 3
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl108b
            SUTPORT = 80
            SUTTERM = 2690
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl108b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w10
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl108b
            SUTPORT = 80
            SUTTERM = 2697
            LOGPATH = /w11
            LOGLEVEL = 0
        ENDSUT
    ENDRTE
    STARTVARIABLE
        WAREHOUSE = 63024
        MEASUREMENT = 21600
        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 = cl109a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w00
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl109a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w01
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl109a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w02
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl109a
                SUTPORT = 80
                SUTTERM = 1
                LOGPATH = /w03
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl109b
                SUTPORT = 80
                SUTTERM = 2692
                LOGPATH = /w03
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl109b
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w04
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl109b
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w05
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl109b
                SUTPORT = 80
                SUTTERM = 2
                LOGPATH = /w06
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl110a
                SUTPORT = 80
                SUTTERM = 2691
                LOGPATH = /w06
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl110a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w07
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl110a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w08
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl110a
                SUTPORT = 80
                SUTTERM = 3
                LOGPATH = /w09
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl110b
                SUTPORT = 80
                SUTTERM = 2690
                LOGPATH = /w09
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl110b
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w10
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl110b
                SUTPORT = 80
                SUTTERM = 2697
                LOGPATH = /w11
                LOGLEVEL = 0
            ENDSUT
        ENDRTE
        STARTVARIABLE
            WAREHOUSE = 63024
            MEASUREMENT = 21600
            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 = cl111a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w00
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl111a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w01
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl111a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w02
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl111a
                SUTPORT = 80
                SUTTERM = 1
                LOGPATH = /w03
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl111b
                SUTPORT = 80
                SUTTERM = 2692
                LOGPATH = /w03
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl111b
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w04
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl111b
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w05
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl111b
```

```
            SUTPORT = 80
            SUTTERM = 2
            LOGPATH = /w06
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl112a
            SUTPORT = 80
            SUTTERM = 2691
            LOGPATH = /w06
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl112a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w07
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl112a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w08
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl112a
            SUTPORT = 80
            SUTTERM = 3
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl112b
            SUTPORT = 80
            SUTTERM = 2690
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl112b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w10
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl112b
            SUTPORT = 80
            SUTTERM = 2697
            LOGPATH = /w11
            LOGLEVEL = 0
        ENDSUT
    ENDRTE
    STARTVARIABLE
        WAREHOUSE = 63024
        MEASUREMENT = 21600
        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 = cl113a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w00
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl113a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w01
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl113a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w02
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl113a
            SUTPORT = 80
            SUTTERM = 1
            LOGPATH = /w03
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl113b
            SUTPORT = 80
            SUTTERM = 2692
            LOGPATH = /w03
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl113b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w04
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl113b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w05
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl113b
            SUTPORT = 80
            SUTTERM = 2
            LOGPATH = /w06
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl114a
            SUTPORT = 80
            SUTTERM = 2691
            LOGPATH = /w06
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl114a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w07
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl114a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w08
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl114a
            SUTPORT = 80
            SUTTERM = 3
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl114b
            SUTPORT = 80
            SUTTERM = 2690
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl114b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w10
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl114b
            SUTPORT = 80
            SUTTERM = 2697
            LOGPATH = /w11
            LOGLEVEL = 0
        ENDSUT
    ENDRTE
    STARTVARIABLE
        WAREHOUSE = 63024
        MEASUREMENT = 21600
        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 = cl115a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w00
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl115a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w01
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl115a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w02
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl115a
                SUTPORT = 80
                SUTTERM = 1
                LOGPATH = /w03
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl115b
                SUTPORT = 80
                SUTTERM = 2692
                LOGPATH = /w03
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl115b
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w04
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl115b
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w05
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl115b
                SUTPORT = 80
                SUTTERM = 2
                LOGPATH = /w06
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl116a
                SUTPORT = 80
                SUTTERM = 2691
                LOGPATH = /w06
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl116a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w07
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl116a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w08
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl116a
                SUTPORT = 80
                SUTTERM = 3
                LOGPATH = /w09
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl116b
                SUTPORT = 80
                SUTTERM = 2690
                LOGPATH = /w09
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl116b
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w10
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl116b
                SUTPORT = 80
                SUTTERM = 2697
                LOGPATH = /w11
                LOGLEVEL = 0
            ENDSUT
        ENDRTE
        STARTVARIABLE
            WAREHOUSE = 63024
            MEASUREMENT = 21600
            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 = cl119a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w00
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl119a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w01
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl119a
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w02
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl119a
                SUTPORT = 80
                SUTTERM = 1
                LOGPATH = /w03
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl119b
                SUTPORT = 80
                SUTTERM = 2692
                LOGPATH = /w03
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl119b
                SUTPORT = 80
                SUTTERM = 2693
                LOGPATH = /w04
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl119b
                SUTPORT = 80
```

```
            SUTTERM = 2693
            LOGPATH = /w05
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl119b
            SUTPORT = 80
            SUTTERM = 2
            LOGPATH = /w06
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl120a
            SUTPORT = 80
            SUTTERM = 2691
            LOGPATH = /w06
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl120a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w07
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl120a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w08
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl120a
            SUTPORT = 80
            SUTTERM = 3
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl120b
            SUTPORT = 80
            SUTTERM = 2690
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl120b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w10
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl120b
            SUTPORT = 80
            SUTTERM = 2697
            LOGPATH = /w11
            LOGLEVEL = 0
        ENDSUT
    ENDRTE
    STARTVARIABLE
        WAREHOUSE = 63024
        MEASUREMENT = 21600
        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 = cl121a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w00
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl121a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w01
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl121a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w02
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl121a
            SUTPORT = 80
            SUTTERM = 1
            LOGPATH = /w03
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl121b
            SUTPORT = 80
            SUTTERM = 2692
            LOGPATH = /w03
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl121b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w04
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl121b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w05
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl121b
            SUTPORT = 80
            SUTTERM = 2
            LOGPATH = /w06
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl122a
            SUTPORT = 80
            SUTTERM = 2691
            LOGPATH = /w06
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl122a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w07
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl122a
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w08
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl122a
            SUTPORT = 80
            SUTTERM = 3
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl122b
            SUTPORT = 80
            SUTTERM = 2690
            LOGPATH = /w09
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl122b
            SUTPORT = 80
            SUTTERM = 2693
            LOGPATH = /w10
            LOGLEVEL = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl122b
            SUTPORT = 80
            SUTTERM = 2697
            LOGPATH = /w11
            LOGLEVEL = 0
        ENDSUT
    ENDRTE
    STARTVARIABLE
        WAREHOUSE = 63024
        MEASUREMENT = 21600
        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 = cl123a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w00
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl123a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w01
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl123b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w02
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl123b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w03
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl124a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w04
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl124a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w05
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl124b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w06
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl124b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w07
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl125a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w08
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl125a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w09
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl125b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w10
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl125b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w11
                        LOGLEVEL = 0
                ENDSUT
        ENDRTE
        STARTVARIABLE
                WAREHOUSE = 63024
                MEASUREMENT = 21600
                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 = cl126a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w00
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl126a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w01
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl126b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w02
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl126b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w03
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl127a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w04
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl127a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w05
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl127b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w06
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl127b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w07
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl128a
                        SUTPORT = 80
                        SUTTERM = 4040
```

```
                LOGPATH = /w08
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl128a
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w09
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl128b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w10
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl128b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w11
                LOGLEVEL = 0
            ENDSUT
        ENDRTE
        STARTVARIABLE
            WAREHOUSE = 63024
            MEASUREMENT = 21600
            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 = cl129a
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w00
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl129a
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w01
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl129b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w02
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl129b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w03
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl130a
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w04
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl130a
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w05
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl130b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w06
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl130b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w07
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl131a
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w08
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl131a
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w09
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl131b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w10
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl131b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w11
                LOGLEVEL = 0
            ENDSUT
        ENDRTE
        STARTVARIABLE
            WAREHOUSE = 63024
            MEASUREMENT = 21600
            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 = cl132a
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w00
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl132a
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w01
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl132b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w02
                LOGLEVEL = 0
            ENDSUT
            STARTSUT
                SUTHOST = cl132b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w03
                LOGLEVEL = 0
            ENDSUT
```

```
        STARTSUT
                SUTHOST = cl133a
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w04
                LOGLEVEL = 0
        ENDSUT
        STARTSUT
                SUTHOST = cl133a
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w05
                LOGLEVEL = 0
        ENDSUT
        STARTSUT
                SUTHOST = cl133b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w06
                LOGLEVEL = 0
        ENDSUT
        STARTSUT
                SUTHOST = cl133b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w07
                LOGLEVEL = 0
        ENDSUT
        STARTSUT
                SUTHOST = cl134a
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w08
                LOGLEVEL = 0
        ENDSUT
        STARTSUT
                SUTHOST = cl134a
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w09
                LOGLEVEL = 0
        ENDSUT
        STARTSUT
                SUTHOST = cl134b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w10
                LOGLEVEL = 0
        ENDSUT
        STARTSUT
                SUTHOST = cl134b
                SUTPORT = 80
                SUTTERM = 4040
                LOGPATH = /w11
                LOGLEVEL = 0
        ENDSUT
ENDRTE
STARTVARIABLE
        WAREHOUSE = 63024
        MEASUREMENT = 21600
        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 = cl135a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w00
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl135a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w01
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl135b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w02
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl135b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w03
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl136a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w04
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl136a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w05
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl136b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w06
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl136b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w07
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl137a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w08
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl137a
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w09
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl137b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w10
                        LOGLEVEL = 0
                ENDSUT
                STARTSUT
                        SUTHOST = cl137b
                        SUTPORT = 80
                        SUTTERM = 4040
                        LOGPATH = /w11
                        LOGLEVEL = 0
                ENDSUT
        ENDRTE
        STARTVARIABLE
                WAREHOUSE = 63024
                MEASUREMENT = 21600
                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 = cl138a
                  SUTPORT = 80
                  SUTTERM = 1346
                  LOGPATH = /w00
                  LOGLEVEL = 0
            ENDSUT
            STARTSUT
                  SUTHOST = cl138a
                  SUTPORT = 80
                  SUTTERM = 1346
                  LOGPATH = /w01
                  LOGLEVEL = 0
            ENDSUT
            STARTSUT
                  SUTHOST = cl138a
                  SUTPORT = 80
                  SUTTERM = 1346
                  LOGPATH = /w02
                  LOGLEVEL = 0
            ENDSUT
            STARTSUT
                  SUTHOST = cl138a
                  SUTPORT = 80
                  SUTTERM = 1346
                  LOGPATH = /w03
                  LOGLEVEL = 0
            ENDSUT
            STARTSUT
                  SUTHOST = cl138a
                  SUTPORT = 80
                  SUTTERM = 1346
                  LOGPATH = /w04
                  LOGLEVEL = 0
            ENDSUT
            STARTSUT
                  SUTHOST = cl138a
                  SUTPORT = 80
                  SUTTERM = 1346
                  LOGPATH = /w05
                  LOGLEVEL = 0
            ENDSUT
            STARTSUT
                  SUTHOST = cl138a
                  SUTPORT = 80
                  SUTTERM = 4
                  LOGPATH = /w06
                  LOGLEVEL = 0
            ENDSUT
            STARTSUT
                  SUTHOST = cl138b
                  SUTPORT = 80
                  SUTTERM = 1342
                  LOGPATH = /w06
                  LOGLEVEL = 0
            ENDSUT
            STARTSUT
                  SUTHOST = cl138b
                  SUTPORT = 80
                  SUTTERM = 1346
                  LOGPATH = /w07
                  LOGLEVEL = 0
            ENDSUT
            STARTSUT
                  SUTHOST = cl138b
                  SUTPORT = 80
                  SUTTERM = 1346
                  LOGPATH = /w08
                  LOGLEVEL = 0
            ENDSUT
            STARTSUT
                  SUTHOST = cl138b
                  SUTPORT = 80
                  SUTTERM = 1346
                  LOGPATH = /w09
                  LOGLEVEL = 0
            ENDSUT
            STARTSUT
                  SUTHOST = cl138b
                  SUTPORT = 80
                  SUTTERM = 1346
                  LOGPATH = /w10
                  LOGLEVEL = 0
            ENDSUT
            STARTSUT
                  SUTHOST = cl138b
                  SUTPORT = 80
                  SUTTERM = 1354
                  LOGPATH = /w11
                  LOGLEVEL = 0
            ENDSUT
      ENDRTE
      STARTVARIABLE
            WAREHOUSE = 63024
            MEASUREMENT = 21600
            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

```
----------------
elilo.conf
----------------
prompt
timeout=20
default=2.6.9-27-New2.EL
relocatable

image=vmlinuz-2.6.9-27.EL
        label=2.6.9-27-New2.EL
        initrd=initrd-2.6.9-27.ELgt3.img
        read-only
        append="rhgb quiet root=/dev/sda2
console=ttyS0,19200n8r noirqdebug
elevator=noop  ide=nodma
ihash_entries=1000000 dhash_entries=1000000
rhash_entries=500000 thash_entries=100000 ro"

--------------------
modprobe.conf
--------------------
alias eth0 e100
alias eth1 e100
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 eth16 tg3
alias eth20 tg3
alias eth24 tg3
alias eth28 tg3
alias eth32 tg3
alias scsi_hostadapter1 mptscsih
options lpfc lpfc_lun_queue_depth=30
alias scsi_hostadapter2 lpfc
alias usb-controller ehci-hcd
alias usb-controller1 uhci-hcd
alias eth35 e1000
alias eth36 e1000

-------------------------
rc.local
-------------------------
#!/bin/sh
#
# This script will be executed *after* all the other
init scripts.
# You can put your own initialization stuff in here if
you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

echo "100 100000 120 512" >
/proc/sys/kernel/sem
echo 0x20000000 > /proc/sys/kernel/shmall
echo 0xc000000000 > /proc/sys/kernel/shmmax
echo 5242880 > /proc/sys/fs/aio-max-nr

# needed for text and RO data in huge pages
mount none /mnt/htlb -t hugetlbfs
chown -R oracle:dba /mnt/*

/sbin/route add -host cl107 dev eth8
/sbin/route add -host cl108 dev eth10
/sbin/route add -host cl109 dev eth12
/sbin/route add -host cl110 dev eth16

/sbin/route add -host cl111 dev eth20
/sbin/route add -host cl112 dev eth24
/sbin/route add -host cl113 dev eth28
/sbin/route add -host cl114 dev eth32

/usr/sbin/ntpdate fjgw

-----------------
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 = 4006
------------------
init.ora
------------------
#########################
# General Database
#########################

trace_enabled = false

control_files =
(/ora_dev/control_001,/ora_dev/control_002)
processes = 850
sessions = 1400
transactions = 1400
db_name                    = tpcc
db_files                   = 1292
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         = 1M

#########################
# Buffer Cache / SGA
#########################

db_cache_size              = 10240M
db_keep_cache_size         = 606000M
db_recycle_cache_size      = 138752M
db_16k_cache_size          = 244032M
db_8k_cache_size           = 2048M
db_4k_cache_size           = 512M
shared_pool_size           = 22272M

java_pool_size             = 0

#########################
# I/O
#########################

db_writer_processes        = 8
disk_asynch_io             = true
db_block_checking          = false
db_block_checksum          = false

#########################
# Undo Management
#########################

undo_management            = auto
undo_retention             = 1
undo_tablespace            = undo_1
transactions_per_rollback_segment = 1

#########################
# Optimizations
#########################

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       = 32

#########################
# Log / Checkpointing
#########################

log_buffer                 = 33554432
log_checkpoint_interval    = 0
log_checkpoint_timeout     = 1600
log_checkpoints_to_alert   = true

#########################
# Statistics
#########################

timed_statistics           = false
statistics_level           = basic

query_rewrite_enabled = false
:::::::::::::::
cl033 configuration
```

```
::::::::::::::::
[OS tunables]
-------------------------------------------------------------------------
------------
::::::::::::::::
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 = 20000
fs.file-max = 30000
# Change Message queue
#kernel.msgmni = 20000
kernel.msgmni = 30000
kernel.msgmnb = 819200

# Change Max process
#kernel.threads-max = 20000
kernel.threads-max = 30000
# Change Semaphore
kernel.sem = 3000  384000 32  128
# Change TCP/IP backlog
net.ipv4.tcp_max_syn_backlog = 4096

::::::::::::::::
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          20000
#tpc             -       nproc           20000
tpc             -       nofile          30000
tpc             -       nproc           30000

# End of file

::::::::::::::::
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


[HTTP server tunables]
--------------------------------------------------------------------------
------------
::::::::::::::::
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>

# 16160 term
StartServers        324
ServerLimit      324
ThreadLimit         50
MaxClients          16200
MinSpareThreads      1
MaxSpareThreads      16200
ThreadsPerChild     50
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#forbid
den
#
#<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.var
#    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.h
tml.var
#    ErrorDocument 414
/error/HTTP_REQUEST_URI_TOO_LARGE.html.
var
#    ErrorDocument 415
/error/HTTP_SERVICE_UNAVAILABLE.html.var
#    ErrorDocument 500
/error/HTTP_INTERNAL_SERVER_ERROR.html.
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>


::::::::::::::::
apache_cl_start.sh
::::::::::::::::
#!/bin/sh
export
LD_LIBRARY_PATH=$ORACLE_HOME/srvm/lib:
$ORACLE_HOME/lib64:$ORACLE_HOME/lib:/usr
/lib:$ORACLE_HOME/rdbms/lib:$ORACLE_HOM
E/network/lib:$TUXDIR/lib

ulimit -u 30000

/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



[Front-end application tunables]
--------------------------------------------------------------------------
------------
::::::::::::::::
tpapl.conf
::::::::::::::::
[TPAPL_INFO]
Term_Base="1"
NumWarehouses="63024"
MaxUsers="630240"
MaxTerm of Client="16160"
CONTROL_Flag="0"
LogPath="/home/tpc/log/userlog.log"

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




[Oracle connection tunables]
--------------------------------------------------------------------------
------------
::::::::::::::::
tnsnames.ora
::::::::::::::::

#
# Installation Generated Net8 Configuration
# Version Date: Oct-27-97
# Filename: Tnsnames.ora
#
extproc_connection_data =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = IPC)(KEY = tpcc))
    (SDU=14600)
    (CONNECT_DATA = (SERVICE_NAME =
tpcc))
  )

tpcc =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
    (SDU=14600)
    (CONNECT_DATA = (SERVICE_NAME =
tpcc))
  )



::::::::::::::::
diff between cl033 and cl035
::::::::::::::::
1108c1108
< Term_Base="1"
---
> Term_Base="16161"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
::::::::::::::::
diff between cl033 and cl036
::::::::::::::::
1108c1108
< Term_Base="1"
---
> Term_Base="32321"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
::::::::::::::::
diff between cl033 and cl037
::::::::::::::::
1108c1108
< Term_Base="1"
---
> Term_Base="48481"
1127d1126
<
1132a1132
> #    (ADDRESS = (PROTOCOL= TCP)(Host=
qfe3)(Port= 1521))
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
1148d1147
<
::::::::::::::::
diff between cl033 and cl038
::::::::::::::::
1108c1108
< Term_Base="1"
---
> Term_Base="64641"
1127d1126
<
1132a1132
> #    (ADDRESS = (PROTOCOL= TCP)(Host=
qfe3)(Port= 1521))
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
1148d1147
<
::::::::::::::::
diff between cl033 and cl039
::::::::::::::::
1108c1108
< Term_Base="1"
---
> Term_Base="80801"
1127d1126
<
1132a1132
> #    (ADDRESS = (PROTOCOL= TCP)(Host=
qfe3)(Port= 1521))
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
1148d1147
<
::::::::::::::::
diff between cl033 and cl040
::::::::::::::::
1108c1108
< Term_Base="1"
---
> Term_Base="96961"
1142c1142
```

```
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
::::::::::::::::
::::::::::::::::
cl105 configuration
::::::::::::::::
::::::::::::::::
[OS tunables]
------------------------------------------------------------------------
------------
::::::::::::::::
::::::::::::::::
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 = 20000
fs.file-max = 30000
# Change Message queue
#kernel.msgmni = 20000
kernel.msgmni = 30000
kernel.msgmnb = 819200

# Change Max process
#kernel.threads-max = 20000
kernel.threads-max = 30000
# Change Semaphore
kernel.sem = 3000  384000 32  128
# Change TCP/IP backlog
net.ipv4.tcp_max_syn_backlog = 4096

::::::::::::::::
::::::::::::::::
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          20000
#tpc             -       nproc           20000
tpc              -       nofile          30000
tpc              -       nproc           30000

# End of file

::::::::::::::::
::::::::::::::::
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


[HTTP server tunables]
```

```
--------------------------------------------------------------------------
------------
::::::::::::::::
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>

# 16160 term
StartServers        324
ServerLimit      324
ThreadLimit         50
MaxClients          16200
MinSpareThreads      1
MaxSpareThreads     16200
ThreadsPerChild     50
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#forbid
den
#
#<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.var
#    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.h
tml.var
#    ErrorDocument 414
/error/HTTP_REQUEST_URI_TOO_LARGE.html.
var
#    ErrorDocument 415
/error/HTTP_SERVICE_UNAVAILABLE.html.var
#    ErrorDocument 500
/error/HTTP_INTERNAL_SERVER_ERROR.html.
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>


::::::::::::::::
apache_cl_start.sh
::::::::::::::::
#!/bin/sh
export
LD_LIBRARY_PATH=$ORACLE_HOME/srvm/lib:
$ORACLE_HOME/lib64:$ORACLE_HOME/lib:/usr
/lib:$ORACLE_HOME/rdbms/lib:$ORACLE_HOM
E/network/lib:$TUXDIR/lib

ulimit -u 30000

/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



[Front-end application tunables]
----------------------------------------------------------------
------------
::::::::::::::::
tpapl.conf
::::::::::::::::
[TPAPL_INFO]
Term_Base="1"
NumWarehouses="63024"
MaxUsers="630240"
MaxTerm of Client="16160"
CONTROL_Flag="0"
LogPath="/home/tpc/log/userlog.log"

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



[Oracle connection tunables]
----------------------------------------------------------------
------------
::::::::::::::::
tnsnames.ora
::::::::::::::::

#
# Installation Generated Net8 Configuration
# Version Date: Oct-27-97
# Filename: Tnsnames.ora
#
extproc_connection_data =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = IPC)(KEY = tpcc))
    (SDU=14600)
    (CONNECT_DATA = (SERVICE_NAME =
tpcc))
  )

tpcc =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
    (SDU=14600)
    (CONNECT_DATA = (SERVICE_NAME =
tpcc))
  )



::::::::::::::::
diff between cl105 and cl106
::::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="129281"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
::::::::::::::::
diff between cl105 and cl107
::::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="145441"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
::::::::::::::::
diff between cl105 and cl108
::::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="161601"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
::::::::::::::::
diff between cl105 and cl109
::::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="177761"
::::::::::::::::
diff between cl105 and cl110
::::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="193921"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
::::::::::::::::
diff between cl105 and cl111
::::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="210081"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
::::::::::::::::
diff between cl105 and cl112
::::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="226241"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
```

```
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
:::::::::::::::
diff between cl105 and cl113
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="242401"
:::::::::::::::
diff between cl105 and cl114
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="258561"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
:::::::::::::::
diff between cl105 and cl115
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="274721"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
:::::::::::::::
diff between cl105 and cl116
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="290881"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
:::::::::::::::
diff between cl105 and cl119
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="307041"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_c)(Port= 1523))
:::::::::::::::
diff between cl105 and cl120
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="323201"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_d)(Port= 1524))
:::::::::::::::
diff between cl105 and cl121
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="339361"
:::::::::::::::
diff between cl105 and cl122
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="355521"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_b)(Port= 1522))
:::::::::::::::
diff between cl105 and cl123
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="371681"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
:::::::::::::::
diff between cl105 and cl124
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="387841"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
:::::::::::::::
diff between cl105 and cl125
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="404001"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
:::::::::::::::
diff between cl105 and cl126
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="420161"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
:::::::::::::::
diff between cl105 and cl127
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="436321"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
:::::::::::::::
diff between cl105 and cl128
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="452481"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
:::::::::::::::
diff between cl105 and cl129
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="468641"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
:::::::::::::::
diff between cl105 and cl130
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="484801"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
:::::::::::::::
diff between cl105 and cl131
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="500961"
1142c1142
<    (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
```

```
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
:::::::::::::::
diff between cl105 and cl132
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="517121"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
:::::::::::::::
diff between cl105 and cl133
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="533281"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
:::::::::::::::
diff between cl105 and cl134
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="549441"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
:::::::::::::::
diff between cl105 and cl135
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="565601"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_e)(Port= 1525))
:::::::::::::::
diff between cl105 and cl136
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="581761"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_f)(Port= 1526))
:::::::::::::::
diff between cl105 and cl137
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="597921"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_g)(Port= 1527))
:::::::::::::::
diff between cl105 and cl138
:::::::::::::::
1108c1108
< Term_Base="113121"
---
> Term_Base="614081"
1142c1142
<     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_a)(Port= 1521))
---
>     (ADDRESS = (PROTOCOL= TCP)(Host=
pqtpc_h)(Port= 1528))
```

Appendix E: Database Creation Code

```
::::::::::::::::::::
Makefile.linux
::::::::::::::::::::
#gcc=/usr/local/bin/gcc
#=================================================
============================+
#       Copyright (c) 1996  Oracle Corp, Redwood
Shores, CA       |
#                OPEN SYSTEMS PERFORMANCE
GROUP                |
#                   All Rights Reserved                    |
#=================================================
============================+
# FILENAME
#      Makefile
# DESCRIPTION
#      Makefile for batch driver, load program and tx
testing.
#=================================================
============================
#
#  Programs:
#
#      tpcc.exe       : OCI TPC-C generator
#      tpccload.exe   : Database loader for TPC-C
#      single_txn.exe : OCI program to test the TPC-
C transactions
#      getrand.exe    :
#      90per.exe      :
#      runtpb.exe     :
#      sleep.exe      :
#      press_return.exe     :
#      runid.exe      :
#

all: compile load

#include
$(ORACLE_HOME)/bench/buildtools/prefix.mk

#I_SYM=-I
#include
$(ORACLE_HOME)/rdbms/lib/env_rdbms.mk

#LINK=/opt/SunProd/SUNWspro6.1/bin/../WS6U1/
bin/cc
#CC=/opt/SunProd/SUNWspro6.1/bin/../WS6U1/bi
n/cc
LINK=/usr/bin/gcc
CC=/usr/bin/gcc

#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 \
#`cat $(ORACLE_HOME)/lib/sysliblist` \
#-o $@

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 \
`cat $(ORACLE_HOME)/lib/sysliblist` \
-L$(ORACLE_HOME)/lib -o $@

#-R$(ORACLE_HOME)/lib -laio -lthread -lposix4 -
lkstat -lm -o $@

I_SYM=-I

TARGS=compile cleanup
REMOVE=rm

DPB_LIB_DIR=../lib
DPB_LIB=$(DPB_LIB_DIR)/dpblibunix.o

TPCBIN=../../bin
INCLUDE=$(I_SYM).
$(I_SYM)$(ORACLE_HOME)/rdbms/demo \
 $(I_SYM)$(ORACLE_HOME)/rdbms/public \
 $(I_SYM)$(ORACLE_HOME)/rdbms/include \
 $(I_SYM)$(ORACLE_HOME)/plsql/public \
 $(I_SYM)$(ORACLE_HOME)/network/public \
 $(I_SYM)$(DPB_LIB_DIR)
ITUX=$(I_SYM)$(ROOTDIR)/include

MEMBS=
OBJS=tpccload.o c_trans.o c_drv_o7.o c_dump.o
tpccpl.o getrand.o 90per.o report.o errrpt.o sleep.o
press_return.o runid.o
CTRAN_OBJS=plnew.o plpay.o plord.o pldel.o
plsto.o
CTRANPOCI_OBJS=plnew.o plpay_oci.o plord.o
pldel.o plsto.o
CTRANTUX_OBJS=plnew_tux.o plpay.o plord.o
pldel_tux.o plsto.o
OTHER_OBJS=c_drv_val.o test_drv.o
test_sample.o test_tran.o single_txn_ran.o
TUX_OBJS=c_drv_tux.o tpccpl_tux.o tpccsvr.o

files:

compile: $(OBJS) $(DPB_LIB)
	@-$(DOTARGS)

load: $(TPCBIN)/tpcc.exe $(TPCBIN)/tpccload.exe
\
	$(TPCBIN)/single_txn.exe
$(TPCBIN)/90per.exe \
	$(TPCBIN)/runtpb.exe
$(TPCBIN)/getrand.exe \
	$(TPCBIN)/sleep.exe $(TPCBIN)/runid.exe \
	$(TPCBIN)/single_txn_ran.exe \
	$(TPCBIN)/press_return.exe \
	$(TUX_OBJS)
	@-$(DOTARGS)

cleanup:
	$(REMOVE) $(OBJS) $(CTRAN_OBJS)
$(CTRANTUX_OBJS) $(OTHER_OBJS) \
	$(TPCBIN)/tpcc.exe $(TPCBIN)/tpccload.exe
\
	$(TPCBIN)/single_txn.exe
$(TPCBIN)/90per.exe \
	$(TPCBIN)/runtpb.exe
$(TPCBIN)/getrand.exe \
	$(TPCBIN)/sleep.exe $(TPCBIN)/runid.exe \
	$(TPCBIN)/single_txn_ran.exe \
	$(TPCBIN)/press_return.exe \
	$(TUX_OBJS)
	@-$(DOTARGS)

$(DPB_LIB):
	( cd $(DPB_LIB_DIR); $(MAKE) -f
Makefile.linux )

report.o: report.c results.h
	$(CC) $(CFLAGS) $(INCLUDE) -c report.c

errrpt.o: errrpt.c results.h
	$(CC) $(CFLAGS) $(INCLUDE) -c errrpt.c

sleep.o: sleep.c
	$(CC) $(CFLAGS) $(INCLUDE) -c sleep.c

press_return.o: press_return.c
	$(CC) $(CFLAGS) $(INCLUDE) -c
press_return.c

runid.o: runid.c
	$(CC) $(CFLAGS) $(INCLUDE) -c runid.c

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

c_drv_o7.o: c_drv_o7.c tpcc.h
	$(CC) $(CFLAGS) $(INCLUDE) -c
c_drv_o7.c

c_drv_val.o: c_drv.c tpcc.h
	$(CP) c_drv.c c_drv_val.c
	-$(CC) $(CFLAGS) -DVALIDATE
$(INCLUDE) -c c_drv_val.c
	$(REMOVE) c_drv_val.c

c_drv_tux.o: c_drv.c tpcc.h
	$(CP) c_drv.c c_drv_tux.c
	-$(CC) $(CFLAGS) -DTUX $(INCLUDE)
$(ITUX) -c c_drv_tux.c
	$(REMOVE) c_drv_tux.c

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

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

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

runtpb.o: runtpb.c
	$(CC) $(CFLAGS) -DORA_AUX $(INCLUDE)
-c runtpb.c

c_trans.o: $(CTRAN_OBJS)
	$(LD) -r -o $@ $(CTRAN_OBJS)

c_trans_tux.o: $(CTRANTUX_OBJS)
	$(LD) -r -o $@ $(CTRANTUX_OBJS)

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

tpccpl_tux.o: tpccpl.c tpcc.h
	$(CP) tpccpl.c tpccpl_tux.c
```

```
	-$(CC) $(CFLAGS) -DTUX $(INCLUDE)
$(ITUX) -c tpccpl_tux.c
	$(REMOVE) tpccpl_tux.c

plnew_tux.o: plnew.c tpcc.h
	$(CP) plnew.c plnew_tux.c
	-$(CC) $(CFLAGS) -DTUX $(INCLUDE)
$(ITUX) -c plnew_tux.c
	$(REMOVE) plnew_tux.c

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

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

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

pldel_tux.o: pldel.c tpcc.h
	$(CP) pldel.c pldel_tux.c
	-$(CC) $(CFLAGS) -DTUX $(INCLUDE)
$(ITUX) -c pldel_tux.c
	$(REMOVE) pldel_tux.c

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

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

tpccsvr.o: tpccsvr.c tpcc.h
	$(CC) $(CFLAGS) $(INCLUDE) $(ITUX) -c
tpccsvr.c

getrand.o: getrand.c
	$(CC) $(CFLAGS) $(INCLUDE) -c getrand.c

90per.o: 90per.c
	$(CC) $(CFLAGS) $(INCLUDE) -c 90per.c

$(TPCBIN)/getrand.exe: getrand.o
	$(LINK) $(LDFLAGS) \
	getrand.o -lc

$(TPCBIN)/sleep.exe: sleep.o
	$(LINK) $(LDFLAGS) \
	sleep.o -lc

$(TPCBIN)/press_return.exe: press_return.o
	$(LINK) $(LDFLAGS) \
	press_return.o -lc

$(TPCBIN)/runid.exe: runid.o
	$(LINK) $(LDFLAGS) \
	runid.o $(DPB_LIB) -lc

$(TPCBIN)/90per.exe: 90per.o
	$(LINK) $(LDFLAGS) \
	90per.o -lc

$(TPCBIN)/tpccload.exe: tpccload.o $(DPB_LIB)
	$(LINK) $(LDFLAGS) \
	tpccload.o $(DPB_LIB) \
	$(SSABED) $(DEF_OPT) $(TTLIBS) -lc

$(TPCBIN)/runtpb.exe: runtpb.o $(DPB_LIB)
	$(LINK) $(LDFLAGS) \
	runtpb.o $(DPB_LIB) \
	$(SSABED) $(DEF_OPT) $(TTLIBS) -lc

$(TPCBIN)/tpcc.exe: c_drv_o7.o c_trans.o
tpccpl.o c_dump.o report.o errrpt.o $(DPB_LIB)
	$(LINK) $(LDFLAGS) \
	c_drv_o7.o c_trans.o tpccpl.o c_dump.o
errrpt.o report.o $(DPB_LIB)\
	$(SSABED) $(DEF_OPT) $(TTLIBS) -lc

$(TPCBIN)/single_txn.exe: single_txn.o
$(DPB_LIB) c_trans.o tpccpl.o c_dump.o
	$(LINK) $(LDFLAGS) \
	single_txn.o c_trans.o tpccpl.o c_dump.o
$(DPB_LIB) \
	$(SSABED) $(DEF_OPT) $(TTLIBS) -lc

$(TPCBIN)/single_txn_ran.exe: single_txn_ran.o
$(DPB_LIB) c_trans.o tpccpl.o c_dump.o
	$(LINK) $(LDFLAGS) \
	single_txn_ran.o c_trans.o tpccpl.o c_dump.o
$(DPB_LIB) \
	$(SSABED) $(DEF_OPT) $(TTLIBS) -lc

::::::::::::::::::
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.sql
::::::::::::::::::
spool analyze.log;
set echo on;

connect tpcc/tpcc

execute dbms_stats.GATHER_TABLE_STATS
(OWNNAME=>'TPCC', -
                        TABNAME=>'STOK', -
                        PARTNAME=>NULL, -

ESTIMATE_PERCENT=>1, -

BLOCK_SAMPLE=>TRUE, -
                        METHOD_OPT=>'FOR
ALL COLUMNS SIZE 1', -
                        DEGREE=>10, -

GRANULARITY=>'DEFAULT', -
                        CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS
(OWNNAME=>'TPCC', -
```

```
                        TABNAME=>'CUST', -
                        PARTNAME=>NULL, -

ESTIMATE_PERCENT=>1, -

BLOCK_SAMPLE=>TRUE, -
                        METHOD_OPT=>'FOR
ALL COLUMNS SIZE 1', -
                        DEGREE=>10, -

GRANULARITY=>'DEFAULT', -
                        CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS
(OWNNAME=>'TPCC', -
                        TABNAME=>'ORDR', -
                        PARTNAME=>NULL, -

ESTIMATE_PERCENT=>1, -

BLOCK_SAMPLE=>TRUE, -
                        METHOD_OPT=>'FOR
ALL COLUMNS SIZE 1', -
                        DEGREE=>10, -

GRANULARITY=>'DEFAULT', -
                        CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS
(OWNNAME=>'TPCC', -
                        TABNAME=>'ORDL', -
                        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(OWNNAM
E=>'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(OWNNAM
E=>'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(OWNNAM
E=>'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(OWNNAM
E=>'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;

::::::::::::::::::::
createdb.sql
::::::::::::::::::::
/* created automatically by  /home/oracle/tpcc-
kit/scripts/buildcreatedb.sh Wed Feb 1 15:37:03
JST 2006 */
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 124511M reuse,
        '/ora_dev/log_1_2' size 124511M 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
::::::::::::::::::::
::::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreateindex.sh Wed Feb 1 15:37:25
JST 2006 */
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 128
  compute statistics
  tablespace icust1_0 ;
    set echo off
    spool off
    exit sql.sqlcode;

::::::::::::::::::::
::::::::::::::::::::
createindex_icust2.sql
::::::::::::::::::::
::::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreateindex.sh Wed Feb 1 15:37:25
JST 2006 */
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 1
  compute statistics
  tablespace icust2_0 ;
    set echo off
    spool off
    exit sql.sqlcode;
```

```
::::::::::::::::::
createindex_idist.sql
::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreateindex.sh Wed Feb 1 15:37:26
JST 2006 */
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
::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreateindex.sh Wed Feb 1 15:37:28
JST 2006 */
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
::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreateindex.sh Wed Feb 1 15:37:31
JST 2006 */
set timing on
  exit 0;

::::::::::::::::::
createindex_iordl.sql
::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreateindex.sh Wed Feb 1 15:37:30
JST 2006 */
set timing on
  exit 0;

::::::::::::::::::
createindex_iordr1.sql
::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreateindex.sh Wed Feb 1 15:37:29
JST 2006 */
set timing on
  exit 0;

::::::::::::::::::
createindex_iordr2.sql
::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreateindex.sh Wed Feb 1 15:37:29
JST 2006 */
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 1
  compute statistics
  tablespace iordr2_0 ;
    set echo off
    spool off
    exit sql.sqlcode;

::::::::::::::::::
createindex_istok.sql
::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreateindex.sh Wed Feb 1 15:37:27
JST 2006 */
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 128
  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 Wed Feb 1 15:37:24
JST 2006 */
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;

::::::::::::::::::
createspacestats.sql
::::::::::::::::::
@space_init
@space_get 12 10
@space_rpt
spool off
exit sql.sqlcode;

::::::::::::::::::
createstoredprocs.sql
::::::::::::::::::
spool createstoreprocs.log
@tkvcinin.sql
spool off
exit sql.sqlcode;

::::::::::::::::::
createtable_cust.sql
::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreatetable.sh Wed Feb 1 15:37:05
JST 2006 */
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 2040000000
  hash is ( (c_id * ( 68000 * 10 ) + c_w_id * 10 +
c_d_id) )
  size 180
  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;

::::::::::::::::::::
createtable_dist.sql
::::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreatetable.sh Wed Feb 1 15:37:08
JST 2006 */
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 680000
  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
::::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreatetable.sh Wed Feb 1 15:37:10
JST 2006 */
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
::::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreatetable.sh Wed Feb 1 15:37:14
JST 2006 */
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
::::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreatetable.sh Wed Feb 1 15:37:18
JST 2006 */
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 680000
    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
::::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreatetable.sh Wed Feb 1 15:37:17
JST 2006 */
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
::::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreatetable.sh Wed Feb 1 15:37:15
JST 2006 */
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 680000
    hash is ( (o_w_id - 1) * 10 + o_d_id - 1 )
    size 1490
    tablespace ordr_0;

  create table ordr (
```

```
    o_id number sort
, o_w_id number
, o_d_id number
, o_c_id number
, o_carrier_id number
, o_ol_cnt number
, o_all_local number
, o_entry_d date
    , constraint ordr_uk primary key ( o_w_id
, o_d_id
, o_id )
  )
  cluster ordrcluster_queue (
    o_w_id
, o_d_id
, o_id
  );
    set echo off
    spool off
    exit sql.sqlcode;

::::::::::::::::::::
createtable_stok.sql
::::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreatetable.sh Wed Feb 1 15:37:11
JST 2006 */
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 6800000000
  hash is ( (s_i_id * 68000 + 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
::::::::::::::::::::
/* created automatically by /home/oracle/tpcc-
kit/scripts/buildcreatetable.sh Wed Feb 1 15:37:04
JST 2006 */
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 68000
  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
::::::::::::::::::::
#!/bin/sh
#NOTE - ANY CHANGES MUST BE MADE TO
CREATETS.KSH AS WELL.
# createts.sh [name] [no. of file] [no. of partition]
[filesize] [ext_size]
#             [unix/nt] [1: temporary ts / 0: others]
[filecount] [no of cpu]
#             [blocksize] [t: bitmapped / f: manual
manage / d: dictionary ]

name=$1
fileno=$2
noofts=$3
filesize=$4
extsize=$5
ver=$6
isTemp=$7
filecount=$8
para=`expr $9 \* 2`
#blocksize=${10}  sh bug workaround
blocksize=`echo $@ | cut -d' ' -f10`
#autospace=${11} sh bug workaround
autospace=`echo $@ | cut -d' ' -f11`
addts=$tpcc_scripts/addts.sh
addfile=$tpcc_scripts/addfile.sh

if expr "x$tpcc_createts_print" = "xt" > /dev/null ;
then

createtsout=${tpcc_genscripts_dir}/createts_node
${tpcc_rac_node}.sh
  fileavg=`expr $fileno / $tpcc_np`

  if test $noofts -gt 1 ; then
    avg_ts_node=`expr $noofts / $tpcc_np`
    if test "x$tpcc_rac_createts_phase" = "x1" ; then
      fileavg=$avg_ts_node
    else
      if test "x$tpcc_rac_createts_phase" = "x2" ;
then
        fileavg=`expr $fileavg - $avg_ts_node`
      fi
    fi
  fi

  fileend=`expr $fileavg \* $tpcc_rac_node`
  filestart=`expr $fileend - $fileavg`
  if expr $tpcc_rac_node = $tpcc_np > /dev/null;
then
    fileend=$fileno
  fi
fi

#if test $ver = unix;
#then
  fileaddr="$tpcc_disks_location/";
#elif test $ver = nt;
#then
#  fileaddr=\\\\.\\
#fi

filecounter=0
i=0
while test $i -lt $noofts; do

  filecount=`expr $filecount + 1`;
  if expr "x$tpcc_createts_print" = "xt" > /dev/null ;
then
    if test "x$tpcc_rac_createts_phase" = "x1" ; then
      if test "x$name" = "xitem" -o "x$name" =
"xiitem" -o "x$name" = "xtemp" -o "x$name" =
"xrestbl" ; then
        if test $tpcc_rac_node = 1 ; then
          echo $addts $name\_$i
$fileaddr$name\_$i\_0 $filesize $extsize
$blocksize $isTemp $autospace \& >>
$createtsout
          rac_count=`expr $rac_count + 1`
          if test "$rac_count" = "$para" ; then
            rac_count=0
            echo wait >> $createtsout
          fi
        fi
      else
        if test $filecounter -ge $filestart -a $filecounter
-lt $fileend ; then
          echo $addts $name\_$i
$fileaddr$name\_$i\_0 $filesize $extsize
$blocksize $isTemp $autospace \& >>
$createtsout
          rac_count=`expr $rac_count + 1`
          if test "$rac_count" = "$para" ; then
            rac_count=0
```

```
          echo wait >> $createtsout
        fi
       fi
      fi
    fi
   else
    $addts $name\_$i $fileaddr$name\_$i\_0
$filesize $extsize $blocksize $isTemp $autospace
\> junk$filecount 2\>\&1 \&;
   fi
   eval "proc$filecount=$!"
   filecounter=`expr $filecounter + 1`

   p=`expr $filecount % $para`;
   if test $p = 0;
   then
     k=`expr $filecount - $para + 1`;
     if test $k -le $8;
     then
       k=`expr $8 + 1`;
     fi
     while test $k -le $filecount ; do
#      wait `eval echo '$proc'$k`
       wait
       eval "proc$k=$?"
       k=`expr $k + 1`;
     done
   fi

   i=`expr $i + 1`;

done

p=`expr $filecount % $para`
if test $p != 0;
then
  k=`expr $filecount - $p + 1`;
  if test $k -le $8;
  then
    k=`expr $8 + 1`;
  fi
  while test $k -le $filecount; do
#    wait `eval echo '$proc'$k`
    wait
    eval "proc$k=$?"
    k=`expr $k + 1`
  done
fi

if test "x$tpcc_createts_print" = "xt" -a
"x$tpcc_rac_createts_phase" = "x1" ; then
  echo $rac_count
  exit 0
fi

if test "x$tpcc_createts_print" = "xt" -a $noofts -gt 1
-a "x$tpcc_rac_createts_phase" = "x2" ; then
  filecounter=0
fi

filecount=0
fileperts=`expr $fileno / $noofts - 1`
if test $fileperts -gt 0 ;
then
  i=0
  while test $i -lt $noofts ; do
    j=0;
    while test $j -lt $fileperts ;do

      filecount=`expr $filecount + 1`;
      if expr "x$tpcc_createts_print" = "xt" >
/dev/null ; then
        if test "x$tpcc_rac_createts_phase" = "x2" ;
then
          if test "x$name" = "xitem" -o "x$name" =
"xtemp" -o "x$name" = "xrestbl" ; then
            if test $tpcc_rac_node = 1 ; then
              echo $addfile $name\_$i
$fileaddr$name\_$i\_`expr $j + 1` $filesize
$isTemp \& >> $createtsout
              rac_count=`expr $rac_count + 1`
              if test "$rac_count" = "$para" ; then
                rac_count=0
                echo wait >> $createtsout
              fi
            fi
          else
            if test $filecounter -ge $filestart -a
$filecounter -lt $fileend ; then
              echo $addfile $name\_$i
$fileaddr$name\_$i\_`expr $j + 1` $filesize
$isTemp \& >> $createtsout
              rac_count=`expr $rac_count + 1`
              if test "$rac_count" = "$para" ; then
                rac_count=0
                echo wait >> $createtsout
              fi
            fi
          fi
        fi
      else
        $addfile $name\_$i
$fileaddr$name\_$i\_`expr $j + 1` $filesize
$isTemp \> junk$filecount 2\>\&1 &
      fi
      eval "proc$filecount=$!"

      filecounter=`expr $filecounter + 1`

      p=`expr $filecount % $para`;
      if test $p = 0;
      then
        k=`expr $filecount - $para + 1`;
        if test $k -le $8;
        then
          k=`expr $8 + 1`;
        fi
        while test $k -le $filecount ; do
#         wait `eval echo '$proc'$k`
          wait
          eval "proc$k=$?"
          k=`expr $k + 1`;
        done
      fi

      j=`expr $j + 1`
    done

    i=`expr $i + 1`
  done

  p=`expr $filecount % $para`
  if test $p != 0;
  then
    k=`expr $filecount - $p + 1`;
    if test $k -le $8;
    then
      k=`expr $8 + 1`;
    fi
    while test $k -le $filecount; do
#      wait `eval echo '$proc'$k`
      wait
      eval "proc$k=$?"
      k=`expr $k + 1`
    done
  fi
fi

if test "x$tpcc_createts_print" = "xt" ; then
  echo $rac_count
fi

i=`expr $8 + 1`
proc=0
while test $i -le $filecount ;do
  eval 'process=$proc"'"$i"
  proc=`expr $proc + $process`
  i=`expr $i + 1`
done

out=`expr $proc % 127`
# Added wait here for all tablespaces to be created
wait
if test $out -ne 0
then
  exit 1;
else
  exit 0;
fi

::::::::::::::::::::
::::::::::::::::::::
driver.sh
::::::::::::::::::::
#!/bin/sh

. ./stepenv.sh

if expr $# \< 1 > /dev/null; then
  echo "$0 <starting stepname> <optional: only>"
  echo OR use:
  echo "$0 buildcreate  - to build the database
creation scripts"
  echo "$0 create       - to create the database
(after buildcreate)"
  echo "$0 steps        - to list individual steps"
  exit 1
fi

if expr x$1 = xsteps > /dev/null; then
  echo stepnames are from creation scripts:
$tpcc_create_steps
  echo
  echo or running steps: $tpcc_steps
  echo "use the 'only' option to only do that step
(otherwise all steps after will also be executed.)"
  echo "   (e.g.  $0 listfiles only)"
  echo "use the 'through' option to do a sequence
of steps (inclusively.)"
  echo "   (e.g.  $0 shutdowndb through startupdb-
p_build)"
  exit 1
fi

startstep=$1
controlcmd=$2
endstep=$3

# Aliases for special steps
if test $startstep = buildcreate; then
  startstep=`echo $tpcc_create_steps | cut -d' ' -f1`
```

```
fi

if test $startstep = create; then
  startstep=`echo $tpcc_steps | cut -d' ' -f1`
fi

if test "x$controlcmd" = x; then
  endstep=
  # Since endstep is null it won't match any other
steps, so we keep going.
elif test "x$controlcmd" = xonly; then
  controlcmd=only
  # this is allowed
elif test "x$controlcmd" = xthrough; then
  actualstep=f
  for step in $tpcc_create_steps $tpcc_steps ; do
    if test "x$step" = "x$endstep"; then
      actualstep=t
    fi
  done
  if test $actualstep = f; then
    echo "Invalid step $endstep.  Use $0 steps  to
show steps."
    exit 1
  fi
else
  echo "Invalid syntax.  Use $0 by itself for help."
  exit 1
fi

echo Starting from step: $startstep

dostep=f
for step in $tpcc_create_steps $tpcc_steps ; do
  if expr $step = $startstep > /dev/null; then
    dostep=t
  fi

  if expr $dostep = t > /dev/null; then
    echo $step
    cd $tpcc_bench
    $tpcc_scripts/`echo $step | cut -d- -f1`.sh `echo
$step | sed -e's/-*$/-/' | cut -d- -f2- | sed -e's/-/ /g`
    lasterror=$?
    cd $tpcc_bench
    if test -n "`find $tpcc_bench/scripts -name
'*.log`"; then
        mv -f *.log `find $tpcc_bench/scripts -name
'*.log` $tpcc_bench/log/
    else
        mv -f *.log $tpcc_bench/log/
    fi

    if expr $lasterror != 0 > /dev/null; then
      if expr $lasterror != 99 > /dev/null; then
        echo Step $step failed.  Stopping driver.
        exit 1
      else
        echo Step $step has completed and
requested stop.  Stopping driver.
        exit 0
      fi
    fi
    if test "x$controlcmd" = xonly; then
      exit 0
    fi
    if test "x$endstep" = "x$step"; then
      echo The driver reached the last desired step.
Stopping driver.
      exit 0
    fi
  fi
done

if expr $dostep = f > /dev/null; then
  echo No such step: $1
fi




::::::::::::::::::::
generated/createts.sh
::::::::::::::::::::
#created automatically by /home/oracle/tpcc-
kit/scripts/buildcreatets.sh Wed Feb 1 15:36:49
JST 2006

# Tablespace ware, ts size 150M (153600K)
# each file 150M (153600K)
# extents 142464K (142464K)
# 1 files

$tpcc_createts ware 1 1       150M 142464K 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 1765000M
(1807360000K)
# each file 3530M (3614720K)
# extents 361054K (361054K)
# 500 files

$tpcc_createts cust 500 1       3530M 361054K
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 1390M (1423360K)
# each file 1390M (1423360K)
# extents 1415424K (1415424K)
# 1 files

$tpcc_createts dist 1 1       1390M 1415424K unix
0       501 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for dist failed.
Exiting.
      exit 0
    fi

# Tablespace hist, ts size 203840M (208732160K)
# each file 7840M (8028160K)
# extents 102768K (102768K)
# 26 files

$tpcc_createts hist 26 1       7840M 102768K unix
0       502 32 4K t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for hist failed.
Exiting.
      exit 0
    fi

# Tablespace stok, ts size 1980000M
(2027520000K)
# each file 3960M (4055040K)
# extents 405138K (405138K)
# 500 files

$tpcc_createts stok 500 1       3960M 405138K
unix 0       528 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
1028 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for item failed.
Exiting.
      exit 0
    fi

# Tablespace ordr, ts size 2905440M
(2975170560K)
# each file 60530M (61982720K)
# extents 103296K (103296K)
# 48 files

$tpcc_createts ordr 48 1       60530M 103296K
unix 0       1029 32 16K t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for ordr failed.
Exiting.
      exit 0
    fi

# Tablespace nord, ts size 23100M (23654400K)
# each file 4620M (4730880K)
# extents 472490K (472490K)
# 5 files

$tpcc_createts nord 5 1       4620M 472490K unix
0       1077 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for nord failed.
Exiting.
      exit 0
    fi

# Tablespace iware, ts size 90M (92160K)
# each file 90M (92160K)
# extents 86024K (86024K)
# 1 files

$tpcc_createts iware 1 1       90M 86024K unix 0
1082 32 auto t
    if expr $? != 0 > /dev/null; then
      echo Creating tablespace for iware failed.
Exiting.
      exit 0
    fi

# Tablespace icust1, ts size 49380M (50565120K)
# each file 8230M (8427520K)
```

```
# extents 32912K (32912K)
# 6 files

$tpcc_createts icust1 6 1       8230M 32912K unix
0       1083 32 16K t
   if expr $? != 0 > /dev/null; then
     echo Creating tablespace for icust1 failed.
Exiting.
     exit 0
   fi

# Tablespace icust2, ts size 108480M
(111083520K)
# each file 6780M (6942720K)
# extents 27104K (27104K)
# 16 files

$tpcc_createts icust2 16 1       6780M 27104K unix
0       1089 32 16K t
   if expr $? != 0 > /dev/null; then
     echo Creating tablespace for icust2 failed.
Exiting.
     exit 0
   fi

# Tablespace idist, ts size 340M (348160K)
# each file 340M (348160K)
# extents 341024K (341024K)
# 1 files

$tpcc_createts idist 1 1       340M 341024K unix 0
1105 32 auto t
   if expr $? != 0 > /dev/null; then
     echo Creating tablespace for idist failed.
Exiting.
     exit 0
   fi

# Tablespace istok, ts size 148080M
(151633920K)
# each file 6170M (6318080K)
# extents 24640K (24640K)
# 24 files

$tpcc_createts istok 24 1       6170M 24640K unix
0       1106 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
1130 32 auto t
   if expr $? != 0 > /dev/null; then
     echo Creating tablespace for iitem failed.
Exiting.
     exit 0
   fi

# Tablespace iordr2, ts size 119200M
(122060800K)
# each file 7450M (7628800K)
# extents 29760K (29760K)
# 16 files

$tpcc_createts iordr2 16 1       7450M 29760K unix
0       1131 32 16K t
   if expr $? != 0 > /dev/null; then
     echo Creating tablespace for iordr2 failed.
Exiting.
     exit 0
   fi

# Tablespace temp, ts size 356400M
(364953600K)
# each file 7920M (8110080K)
# extents 202506K (202506K)
# 45 files

$tpcc_createts temp 45 1       7920M 202506K
unix 1       1147 32 auto t
   if expr $? != 0 > /dev/null; then
     echo Creating tablespace for temp failed.
Exiting.
     exit 0
   fi

::::::::::::::::::::
::::::::::::::::::::
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/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/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/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/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/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/bi
n/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/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);
}


::::::::::::::::::::
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/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/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/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/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/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/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 */



::::::::::::::::::::
loadcust.sh
::::::::::::::::::::
#created automatically by /home/oracle/tpcc-
kit/scripts/evenload.sh Wed Feb 1 15:37:22 JST
2006
rm -f loadcust*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 68000 -C  -l 1 -m 46 >>
loadcust0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 47 -m 92 >>
loadcust1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 93 -m 138 >>
loadcust2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 139 -m 184 >>
loadcust3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 185 -m 230 >>
loadcust4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 231 -m 276 >>
loadcust5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 277 -m 322 >>
loadcust6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 323 -m 368 >>
loadcust7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 369 -m 415 >>
loadcust8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 416 -m 462 >>
loadcust9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 463 -m 509 >>
loadcust10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 510 -m 556 >>
loadcust11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 557 -m 603 >>
loadcust12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 604 -m 650 >>
loadcust13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 651 -m 697 >>
loadcust14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 698 -m 744 >>
loadcust15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 745 -m 791 >>
loadcust16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 792 -m 838 >>
loadcust17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 839 -m 885 >>
loadcust18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 886 -m 932 >>
loadcust19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 933 -m 979 >>
loadcust20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 980 -m 1026 >>
loadcust21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1027 -m 1073 >>
loadcust22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1074 -m 1120 >>
loadcust23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1121 -m 1167 >>
loadcust24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1168 -m 1214 >>
loadcust25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1215 -m 1261 >>
loadcust26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1262 -m 1308 >>
loadcust27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1309 -m 1355 >>
loadcust28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1356 -m 1402 >>
loadcust29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1403 -m 1449 >>
loadcust30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1450 -m 1496 >>
loadcust31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1497 -m 1543 >>
loadcust32.log 2>&1 &
allprocs="$allprocs ${!}"
```

```
$tpcc_load -M 68000 -C  -l 1544 -m 1590 >>
loadcust33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1591 -m 1637 >>
loadcust34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1638 -m 1684 >>
loadcust35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1685 -m 1731 >>
loadcust36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1732 -m 1778 >>
loadcust37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1779 -m 1825 >>
loadcust38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1826 -m 1872 >>
loadcust39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1873 -m 1919 >>
loadcust40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1920 -m 1966 >>
loadcust41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 1967 -m 2013 >>
loadcust42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2014 -m 2060 >>
loadcust43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2061 -m 2107 >>
loadcust44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2108 -m 2154 >>
loadcust45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2155 -m 2201 >>
loadcust46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2202 -m 2248 >>
loadcust47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2249 -m 2295 >>
loadcust48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2296 -m 2342 >>
loadcust49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2343 -m 2389 >>
loadcust50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2390 -m 2436 >>
loadcust51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2437 -m 2483 >>
loadcust52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2484 -m 2530 >>
loadcust53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2531 -m 2577 >>
loadcust54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2578 -m 2624 >>
loadcust55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2625 -m 2671 >>
loadcust56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2672 -m 2718 >>
loadcust57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2719 -m 2765 >>
loadcust58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2766 -m 2812 >>
loadcust59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2813 -m 2859 >>
loadcust60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2860 -m 2906 >>
loadcust61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2907 -m 2953 >>
loadcust62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -C  -l 2954 -m 3000 >>
loadcust63.log 2>&1 &
allprocs="$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 Wed Feb 1 15:37:19 JST
2006
rm -f loadhist*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 68000 -h  -b 1 -e 1062 >>
loadhist0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 1063 -e 2124 >>
loadhist1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 2125 -e 3186 >>
loadhist2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 3187 -e 4248 >>
loadhist3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 4249 -e 5310 >>
loadhist4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 5311 -e 6372 >>
loadhist5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 6373 -e 7434 >>
loadhist6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 7435 -e 8496 >>
loadhist7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 8497 -e 9558 >>
loadhist8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 9559 -e 10620 >>
loadhist9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 10621 -e 11682 >>
loadhist10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 11683 -e 12744 >>
loadhist11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 12745 -e 13806 >>
loadhist12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 13807 -e 14868 >>
loadhist13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 14869 -e 15930 >>
loadhist14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 15931 -e 16992 >>
loadhist15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 16993 -e 18054 >>
loadhist16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 18055 -e 19116 >>
loadhist17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 19117 -e 20178 >>
loadhist18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 20179 -e 21240 >>
loadhist19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 21241 -e 22302 >>
loadhist20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 22303 -e 23364 >>
loadhist21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 23365 -e 24426 >>
loadhist22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 24427 -e 25488 >>
loadhist23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 25489 -e 26550 >>
loadhist24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 26551 -e 27612 >>
loadhist25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 27613 -e 28674 >>
loadhist26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 28675 -e 29736 >>
loadhist27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 29737 -e 30798 >>
loadhist28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 30799 -e 31860 >>
loadhist29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 31861 -e 32922 >>
loadhist30.log 2>&1 &
allprocs="$allprocs ${!}"
```

```
$tpcc_load -M 68000 -h  -b 32923 -e 33984 >>
loadhist31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 33985 -e 35047 >>
loadhist32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 35048 -e 36110 >>
loadhist33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 36111 -e 37173 >>
loadhist34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 37174 -e 38236 >>
loadhist35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 38237 -e 39299 >>
loadhist36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 39300 -e 40362 >>
loadhist37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 40363 -e 41425 >>
loadhist38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 41426 -e 42488 >>
loadhist39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 42489 -e 43551 >>
loadhist40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 43552 -e 44614 >>
loadhist41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 44615 -e 45677 >>
loadhist42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 45678 -e 46740 >>
loadhist43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 46741 -e 47803 >>
loadhist44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 47804 -e 48866 >>
loadhist45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 48867 -e 49929 >>
loadhist46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 49930 -e 50992 >>
loadhist47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 50993 -e 52055 >>
loadhist48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 52056 -e 53118 >>
loadhist49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 53119 -e 54181 >>
loadhist50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 54182 -e 55244 >>
loadhist51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 55245 -e 56307 >>
loadhist52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 56308 -e 57370 >>
loadhist53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 57371 -e 58433 >>
loadhist54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 58434 -e 59496 >>
loadhist55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 59497 -e 60559 >>
loadhist56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 60560 -e 61622 >>
loadhist57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 61623 -e 62685 >>
loadhist58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 62686 -e 63748 >>
loadhist59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 63749 -e 64811 >>
loadhist60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 64812 -e 65874 >>
loadhist61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 65875 -e 66937 >>
loadhist62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -h  -b 66938 -e 68000 >>
loadhist63.log 2>&1 &
allprocs="$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 Wed Feb 1 15:37:20 JST
2006
rm -f loadnord*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 68000 -n  -b 1 -e 1062 >>
loadnord0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 1063 -e 2124 >>
loadnord1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 2125 -e 3186 >>
loadnord2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 3187 -e 4248 >>
loadnord3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 4249 -e 5310 >>
loadnord4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 5311 -e 6372 >>
loadnord5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 6373 -e 7434 >>
loadnord6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 7435 -e 8496 >>
loadnord7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 8497 -e 9558 >>
loadnord8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 9559 -e 10620 >>
loadnord9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 10621 -e 11682 >>
loadnord10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 11683 -e 12744 >>
loadnord11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 12745 -e 13806 >>
loadnord12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 13807 -e 14868 >>
loadnord13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 14869 -e 15930 >>
loadnord14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 15931 -e 16992 >>
loadnord15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 16993 -e 18054 >>
loadnord16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 18055 -e 19116 >>
loadnord17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 19117 -e 20178 >>
loadnord18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 20179 -e 21240 >>
loadnord19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 21241 -e 22302 >>
loadnord20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 22303 -e 23364 >>
loadnord21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 23365 -e 24426 >>
loadnord22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 24427 -e 25488 >>
loadnord23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 25489 -e 26550 >>
loadnord24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 26551 -e 27612 >>
loadnord25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 27613 -e 28674 >>
loadnord26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 28675 -e 29736 >>
loadnord27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 29737 -e 30798 >>
loadnord28.log 2>&1 &
allprocs="$allprocs ${!}"
```

```
$tpcc_load -M 68000 -n  -b 30799 -e 31860 >>
loadnord29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 31861 -e 32922 >>
loadnord30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 32923 -e 33984 >>
loadnord31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 33985 -e 35047 >>
loadnord32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 35048 -e 36110 >>
loadnord33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 36111 -e 37173 >>
loadnord34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 37174 -e 38236 >>
loadnord35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 38237 -e 39299 >>
loadnord36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 39300 -e 40362 >>
loadnord37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 40363 -e 41425 >>
loadnord38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 41426 -e 42488 >>
loadnord39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 42489 -e 43551 >>
loadnord40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 43552 -e 44614 >>
loadnord41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 44615 -e 45677 >>
loadnord42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 45678 -e 46740 >>
loadnord43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 46741 -e 47803 >>
loadnord44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 47804 -e 48866 >>
loadnord45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 48867 -e 49929 >>
loadnord46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 49930 -e 50992 >>
loadnord47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 50993 -e 52055 >>
loadnord48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 52056 -e 53118 >>
loadnord49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 53119 -e 54181 >>
loadnord50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 54182 -e 55244 >>
loadnord51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 55245 -e 56307 >>
loadnord52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 56308 -e 57370 >>
loadnord53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 57371 -e 58433 >>
loadnord54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 58434 -e 59496 >>
loadnord55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 59497 -e 60559 >>
loadnord56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 60560 -e 61622 >>
loadnord57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 61623 -e 62685 >>
loadnord58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 62686 -e 63748 >>
loadnord59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 63749 -e 64811 >>
loadnord60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 64812 -e 65874 >>
loadnord61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 65875 -e 66937 >>
loadnord62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -n  -b 66938 -e 68000 >>
loadnord63.log 2>&1 &
allprocs="$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 Wed Feb 1 15:37:20 JST
2006
rm -f loadordrordl*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy0.dat -b 1 -e 1062
>> loadordrordl0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy1.dat -b 1063 -e
2124 >> loadordrordl1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy2.dat -b 2125 -e
3186 >> loadordrordl2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy3.dat -b 3187 -e
4248 >> loadordrordl3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy4.dat -b 4249 -e
5310 >> loadordrordl4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy5.dat -b 5311 -e
6372 >> loadordrordl5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy6.dat -b 6373 -e
7434 >> loadordrordl6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy7.dat -b 7435 -e
8496 >> loadordrordl7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy8.dat -b 8497 -e
9558 >> loadordrordl8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy9.dat -b 9559 -e
10620 >> loadordrordl9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy10.dat -b 10621 -e
11682 >> loadordrordl10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy11.dat -b 11683 -e
12744 >> loadordrordl11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy12.dat -b 12745 -e
13806 >> loadordrordl12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy13.dat -b 13807 -e
14868 >> loadordrordl13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy14.dat -b 14869 -e
15930 >> loadordrordl14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy15.dat -b 15931 -e
16992 >> loadordrordl15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy16.dat -b 16993 -e
18054 >> loadordrordl16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy17.dat -b 18055 -e
19116 >> loadordrordl17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy18.dat -b 19117 -e
20178 >> loadordrordl18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy19.dat -b 20179 -e
21240 >> loadordrordl19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy20.dat -b 21241 -e
22302 >> loadordrordl20.log 2>&1 &
allprocs="$allprocs ${!}"
```

```
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy21.dat -b 22303 -e
23364 >> loadordrordl21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy22.dat -b 23365 -e
24426 >> loadordrordl22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy23.dat -b 24427 -e
25488 >> loadordrordl23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy24.dat -b 25489 -e
26550 >> loadordrordl24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy25.dat -b 26551 -e
27612 >> loadordrordl25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy26.dat -b 27613 -e
28674 >> loadordrordl26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy27.dat -b 28675 -e
29736 >> loadordrordl27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy28.dat -b 29737 -e
30798 >> loadordrordl28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy29.dat -b 30799 -e
31860 >> loadordrordl29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy30.dat -b 31861 -e
32922 >> loadordrordl30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy31.dat -b 32923 -e
33984 >> loadordrordl31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy32.dat -b 33985 -e
35047 >> loadordrordl32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy33.dat -b 35048 -e
36110 >> loadordrordl33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy34.dat -b 36111 -e
37173 >> loadordrordl34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy35.dat -b 37174 -e
38236 >> loadordrordl35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy36.dat -b 38237 -e
39299 >> loadordrordl36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy37.dat -b 39300 -e
40362 >> loadordrordl37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy38.dat -b 40363 -e
41425 >> loadordrordl38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy39.dat -b 41426 -e
42488 >> loadordrordl39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy40.dat -b 42489 -e
43551 >> loadordrordl40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy41.dat -b 43552 -e
44614 >> loadordrordl41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy42.dat -b 44615 -e
45677 >> loadordrordl42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy43.dat -b 45678 -e
46740 >> loadordrordl43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy44.dat -b 46741 -e
47803 >> loadordrordl44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy45.dat -b 47804 -e
48866 >> loadordrordl45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy46.dat -b 48867 -e
49929 >> loadordrordl46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy47.dat -b 49930 -e
50992 >> loadordrordl47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy48.dat -b 50993 -e
52055 >> loadordrordl48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy49.dat -b 52056 -e
53118 >> loadordrordl49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy50.dat -b 53119 -e
54181 >> loadordrordl50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy51.dat -b 54182 -e
55244 >> loadordrordl51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy52.dat -b 55245 -e
56307 >> loadordrordl52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy53.dat -b 56308 -e
57370 >> loadordrordl53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy54.dat -b 57371 -e
58433 >> loadordrordl54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy55.dat -b 58434 -e
59496 >> loadordrordl55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy56.dat -b 59497 -e
60559 >> loadordrordl56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy57.dat -b 60560 -e
61622 >> loadordrordl57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy58.dat -b 61623 -e
62685 >> loadordrordl58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy59.dat -b 62686 -e
63748 >> loadordrordl59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy60.dat -b 63749 -e
64811 >> loadordrordl60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy61.dat -b 64812 -e
65874 >> loadordrordl61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy62.dat -b 65875 -e
66937 >> loadordrordl62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -o
${tpcc_disks_location}dummy63.dat -b 66938 -e
68000 >> loadordrordl63.log 2>&1 &
allprocs="$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 Wed Feb 1 15:37:22 JST
2006
rm -f loadstok*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 68000 -S  -j 1 -k 1562 >>
loadstok0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 1563 -k 3124 >>
loadstok1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 3125 -k 4686 >>
loadstok2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 4687 -k 6248 >>
loadstok3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 6249 -k 7810 >>
loadstok4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 7811 -k 9372 >>
loadstok5.log 2>&1 &
allprocs="$allprocs ${!}"
```

```
$tpcc_load -M 68000 -S  -j 9373 -k 10934 >>
loadstok6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 10935 -k 12496 >>
loadstok7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 12497 -k 14058 >>
loadstok8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 14059 -k 15620 >>
loadstok9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 15621 -k 17182 >>
loadstok10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 17183 -k 18744 >>
loadstok11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 18745 -k 20306 >>
loadstok12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 20307 -k 21868 >>
loadstok13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 21869 -k 23430 >>
loadstok14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 23431 -k 24992 >>
loadstok15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 24993 -k 26554 >>
loadstok16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 26555 -k 28116 >>
loadstok17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 28117 -k 29678 >>
loadstok18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 29679 -k 31240 >>
loadstok19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 31241 -k 32802 >>
loadstok20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 32803 -k 34364 >>
loadstok21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 34365 -k 35926 >>
loadstok22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 35927 -k 37488 >>
loadstok23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 37489 -k 39050 >>
loadstok24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 39051 -k 40612 >>
loadstok25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 40613 -k 42174 >>
loadstok26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 42175 -k 43736 >>
loadstok27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 43737 -k 45298 >>
loadstok28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 45299 -k 46860 >>
loadstok29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 46861 -k 48422 >>
loadstok30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 48423 -k 49984 >>
loadstok31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 49985 -k 51547 >>
loadstok32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 51548 -k 53110 >>
loadstok33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 53111 -k 54673 >>
loadstok34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 54674 -k 56236 >>
loadstok35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 56237 -k 57799 >>
loadstok36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 57800 -k 59362 >>
loadstok37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 59363 -k 60925 >>
loadstok38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 60926 -k 62488 >>
loadstok39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 62489 -k 64051 >>
loadstok40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 64052 -k 65614 >>
loadstok41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 65615 -k 67177 >>
loadstok42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 67178 -k 68740 >>
loadstok43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 68741 -k 70303 >>
loadstok44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 70304 -k 71866 >>
loadstok45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 71867 -k 73429 >>
loadstok46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 73430 -k 74992 >>
loadstok47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 74993 -k 76555 >>
loadstok48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 76556 -k 78118 >>
loadstok49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 78119 -k 79681 >>
loadstok50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 79682 -k 81244 >>
loadstok51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 81245 -k 82807 >>
loadstok52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 82808 -k 84370 >>
loadstok53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 84371 -k 85933 >>
loadstok54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 85934 -k 87496 >>
loadstok55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 87497 -k 89059 >>
loadstok56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 89060 -k 90622 >>
loadstok57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 90623 -k 92185 >>
loadstok58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 92186 -k 93748 >>
loadstok59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 93749 -k 95311 >>
loadstok60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 95312 -k 96874 >>
loadstok61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 96875 -k 98437 >>
loadstok62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 68000 -S  -j 98438 -k 100000 >>
loadstok63.log 2>&1 &
allprocs="$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

::::::::::::::::::
::::::::::::::::::
localoptions.sh
::::::::::::::::::
#LOCAL OPTION FILE- You must fill these in
# before the driver will work.


#oracle sid to use for the run
ORACLE_SID=tpcc


#folder location of the database files (or links to
raw partitions)
tpcc_disks_location=/ora_dev

#FOR NT
#tpcc_disks_location=\\\\.\\
```

```
#FOR RAC

#node id
#tpcc_rac_id=1


# How many createts_node*.sh will be run in this
node, started from tpcc_rac_id
# eq. if tpcc_rac_id is 3 and
tpcc_rac_createts_count is 2
# createts_node3.sh and createts_node4.sh will
be executed

#tpcc_rac_createts_count=1


#locations of various files used in the generation
scripts.
#(you can usually leave these alone.)
tpcc_sql_dir=${tpcc_bench}/scripts/sql
tpcc_log_dir=${tpcc_bench}/log
tpcc_genscripts_dir=${tpcc_bench}/scripts/generat
ed


#Once you have filled all the options, comment
#out or delete this line.
#tpcc_no_options=t

::::::::::::::::::::
p_build.ora
::::::::::::::::::::
#db_writer_processes=1

compatible = 10.1.0.0.0
db_name = tpcc
control_files =
(/ora_dev/control_001,/ora_dev/control_002)
#parallel_max_servers = 16
#recovery_parallelism = 16
db_files = 1292
db_cache_size   = 8533M
db_4k_cache_size  = 3200M
db_8k_cache_size  = 3200M
db_16k_cache_size  = 8533M
dml_locks = 500
statistics_level = basic
log_buffer = 1048576
processes = 800
sessions = 800
transactions = 800
shared_pool_size = 16000M
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 = 20M

::::::::::::::::::::
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

::::::::::::::::::::
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


::::::::::::::::::::
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_HT
YPE_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,fty
pe,indp,alen,arcode) \
     DISCARD ocierror(__FILE__,__LINE__,(errp),
\

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HT
YPE_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,O
CI_DEFAULT));


/* use with callback data */
#define
OCIBNDRAD(stmp,bndp,errp,sqlvar,progvl,ftype,in
dp,ctxp,\
               cbf_nodata,cbf_data) \
     DISCARD ocierror(__FILE__,__LINE__,(errp),
\

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HT
YPE_BIND,0,(dvoid**)0)); \
     DISCARD ocierror(__FILE__,__LINE__,(errp),
\

OCIBindByName((stmp),&(bndp),(errp),(text
*)(sqlvar), \
                    strlen((sqlvar)),0,(progvl),(ftype), \

indp,0,0,0,0,OCI_DATA_AT_EXEC)); \
     DISCARD ocierror(__FILE__,__LINE__,(errp),
\

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



/* bind in/out for plsql without indicator and rcode */
#define
OCIBNDPL(stmp,bndp,errp,sqlvar,progv,progvl,fty
pe,alen) \
   DISCARD ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HT
YPE_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,ftyp
e,indp,alen,arcode) \
     DISCARD ocierror(__FILE__,__LINE__,(errp),
\

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HT
YPE_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,ft
ype,alen,ms,cu) \
     DISCARD ocierror(__FILE__,__LINE__, (errp),
\

OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HT
YPE_BIND,0,(dvoid**)0));\
     DISCARD
ocierror(__FILE__,__LINE__,(errp),\

OCIBindByName((stmp),&(bndp),(errp),(CONST
text *)(sqlvar), \
          (sb4)strlen((CONST char *)
(sqlvar)),(void *)(progv), \

(progvl),(ftype),NULL,(alen),NULL,(ms),(cu),OCI_
DEFAULT));

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

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

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

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

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


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

OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTY
PE_DEFINE,0,\
                    (dvoid**)0);\

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(prog
v),(progvl),\

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


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

OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTY
PE_DEFINE,0,\
                    (dvoid**)0);\

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(prog
v),\

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

#define
OCIDFNDYN(stmp,dfnp,errp,pos,progv,progvl,ftyp
e,indp,ctxp,cbf_data) \
     ocierror(__FILE__,__LINE__,(errp), \

OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTY
PE_DEFINE,0,\
                    (dvoid**)0));\
     ocierror(__FILE__,__LINE__,(errp), \

OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(prog
v), (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

::::::::::::::::::::
::::::::::::::::::::
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_st
ate, \
  :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_CARR
IER_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_CARR
IER_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,(dvoid
*)0,0,0,0);
    OCIEnvInit(&tpcenv, OCI_DEFAULT, 0, (dvoid
**)0);
    OCIHandleAlloc((dvoid *)tpcenv, (dvoid
**)&tpcsrv, OCI_HTYPE_SERVER, 0 , (dvoid
**)0);
    OCIHandleAlloc((dvoid *)tpcenv, (dvoid
**)&errhp, OCI_HTYPE_ERROR, 0 , (dvoid **)0);
    OCIHandleAlloc((dvoid *)tpcenv, (dvoid
**)&tpcsvc, OCI_HTYPE_SVCCTX, 0 , (dvoid
**)0);
    OCIServerAttach(tpcsrv, errhp, (text
*)0,0,OCI_DEFAULT);
    OCIAttrSet((dvoid *)tpcsvc,
OCI_HTYPE_SVCCTX, (dvoid *)tpcsrv,
            (ub4)0,OCI_ATTR_SERVER, errhp);
    OCIHandleAlloc((dvoid *)tpcenv, (dvoid
**)&tpcusr, OCI_HTYPE_SESSION, 0 , (dvoid
**)0);
    OCIAttrSet((dvoid *)tpcusr,
OCI_HTYPE_SESSION, (dvoid *)uid,

(ub4)strlen(uid),OCI_ATTR_USERNAME, errhp);
    OCIAttrSet((dvoid *)tpcusr,
OCI_HTYPE_SESSION, (dvoid *)pwd,
(ub4)strlen(pwd),
         OCI_ATTR_PASSWORD, errhp);
    OCIERROR(errhp, OCISessionBegin(tpcsvc,
errhp, tpcusr, OCI_CRED_RDBMS,
OCI_DEFAULT));

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

    fprintf (stderr, "\nConnected to Oracle userid
'%s/%s'.\n", uid, pwd);

    /* open cursors and parse statement */
    if (do_A || do_w)  {

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid
**)(&curw), OCI_HTYPE_STMT, 0, (dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curw,
errhp, (text *)SQLTXTW,
              strlen((char *)SQLTXTW), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_A || do_d) {

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid
**)(&curd), OCI_HTYPE_STMT, 0, (dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curd,
errhp, (text *)SQLTXTD,
              strlen((char *)SQLTXTD), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_A || do_c || do_C) {

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid
**)(&curc), OCI_HTYPE_STMT, 0, (dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curc,
errhp, (text *)SQLTXTC,
              strlen((char *)SQLTXTC), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid
**)(&curcs), OCI_HTYPE_STMT, 0, (dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curcs,
errhp, (text *)SQLTXTCQUERY,
              strlen((char *)SQLTXTCQUERY),
(ub4) OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_A || do_h) {

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid
**)(&curh), OCI_HTYPE_STMT, 0, (dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curh,
errhp, (text *)SQLTXTH,
              strlen((char *)SQLTXTH), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_A || do_s || do_S) {

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid
**)(&curs), OCI_HTYPE_STMT, 0, (dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curs,
errhp, (text *)SQLTXTS,
              strlen((char *)SQLTXTS), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid
**)(&curss), OCI_HTYPE_STMT, 0, (dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curss,
errhp, (text *)SQLTXTSQUERY,
              strlen((char *)SQLTXTSQUERY),
(ub4) OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_A || do_i) {

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid
**)(&curi), OCI_HTYPE_STMT, 0, (dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curi,
errhp, (text *)SQLTXTI,
              strlen((char *)SQLTXTI), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_A || do_o) {
      int stat;
      char fname[160];

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid
**)(&curo1), OCI_HTYPE_STMT, 0, (dvoid**)0));
      DISCARD strcpy(fname,basename);
      DISCARD strcat(fname, "/");
      DISCARD strcat(fname,
"benchrun/blocks/load_ordordl.sql");
      stat = sqlfile(fname, stmbuf);
      if (!stat)
      {
        fprintf (stderr, "unable to open %s
\n",fname);
        quit();
        exit(1);
      }
      OCIERROR(errhp,OCIStmtPrepare(curo1,
errhp, stmbuf,
              strlen((char *)stmbuf), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_A || do_n) {

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid
**)(&curno), OCI_HTYPE_STMT, 0, (dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curno,
errhp, (text *)SQLTXTNO,
              strlen((char *)SQLTXTNO), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    /* bind variables */

    /* warehouse */

    if (do_A || do_w) {
      OCIERROR(errhp, OCIBindByName(curw,
&w_id_bp, errhp, (text *)(":w_id"), strlen((":w_id")),
              (ub1 *)&(w_id), sizeof(w_id),
SQLT_INT, (dvoid *) 0, (ub2 *)0, (ub2 *)0,
              (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curw,
&w_name_bp, errhp,(text *)":w_name",
strlen(":w_name"),
              (ub1 *)w_name, 11, SQLT_STR,
(dvoid *) 0, (ub2 *)0, (ub2 *)0,
              (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curw,
&w_street1_bp, errhp, (text *)":w_street_1",
              strlen(":w_street_1"), (ub1
*)w_street_1, 21, SQLT_STR,
              (dvoid *) 0, (ub2 *)0, (ub2 *)0,
              (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curw,
&w_street2_bp, errhp, (text *)":w_street_2",
              strlen(":w_street_2"), (ub1
*)w_street_2, 21, SQLT_STR,
              (dvoid *) 0, (ub2 *)0, (ub2 *)0,
              (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curw,
&w_city_bp, errhp, (text *)":w_city",
              strlen(":w_city"), (ub1 *)w_city, 21,
SQLT_STR,
              (dvoid *) 0, (ub2 *)0, (ub2 *)0,
              (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curw,
&w_state_bp, errhp, (text *)":w_state",
              strlen(":w_state"), (ub1 *)w_state, 2,
SQLT_CHR,
              (dvoid *) 0, (ub2 *)0, (ub2 *)0,
              (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

      OCIERROR(errhp, OCIBindByName(curw,
&w_zip_bp, errhp, (text *)":w_zip",
              strlen(":w_zip"), (ub1 *)w_zip, 9,
SQLT_CHR,
              (dvoid *) 0, (ub2 *)0, (ub2 *)0,
```

```
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curw,
&w_tax_bp, errhp, (text *)":w_tax",
                strlen(":w_tax"), (ub1 *) & w_tax,
sizeof(w_tax), SQLT_FLT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
    }

    /* district */

    if (do_A || do_d) {
      OCIERROR(errhp, OCIBindByName(curd,
&d_id_bp, errhp, (text *)":d_id",
                strlen(":d_id"), (ub1 *)d_id, sizeof(int),
SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curd,
&d_w_id_bp, errhp, (text *)":d_w_id",
                strlen(":d_w_id"), (ub1 *)d_w_id,
sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curd,
&d_name_bp, errhp, (text *)":d_name",
                strlen(":d_name"), (ub1 *)d_name,
11, SQLT_STR,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curd,
&d_street1_bp, errhp, (text *)":d_street_1",
                strlen(":d_street_1"), (ub1
*)d_street_1, 21, SQLT_STR,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curd,
&d_street2_bp, errhp, (text *)":d_street_2",
                strlen(":d_street_2"), (ub1
*)d_street_2, 21, SQLT_STR,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));

        OCIERROR(errhp, OCIBindByName(curd,
&d_city_bp, errhp, (text *)":d_city",
                strlen(":d_city"), (ub1 *)d_city, 21,
SQLT_STR,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (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 %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
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);
}
```

Appendix F: 60 Day Space Calculation

TPM : 792,101,96								
Warehouses : 63024								
SEGMENT	**TYPE**	**TSPACE**	**BLOCKS**	**BLOCK _SIZE**	**KB**	**FIVE_PCT (KB)**	**DAILY_GROW (KB)**	**TOTAL(KB)**
CUSTCLUSTER	CLUSTER	CUST	714,706,393	2,048	1,429,412,786	71,470,639	0	1,500,883,425
DB_STAT	SYS	SYSTEM	1,048,576	2,048	2,097,152	0		2,097,152
DCLUSTER	CLUSTER	DIST	707,712	2,048	1,415,424	70,771	0	1,486,195
HIST	TABLE	HIST	52,103,376	4,096	208,413,504	0	41,910,320	250,323,824
ICUST1	INDEX	ICUST1	2,848,945	16,384	45,583,120	2,279,156	0	47,862,276
ICUST2	INDEX	ICUST2	6,122,116	16,384	97,953,856	4,897,693	0	102,851,549
IDIST	INDEX	IDIST	170,512	2,048	341,024	17,051	0	358,075
IITEM	INDEX	ITEMS	5,632	2,048	11,264	563	0	11,827
IORDR2	INDEX	IORD2	7,618,560	16,384	121,896,960	6,094,848	0	127,991,808
ISTOK	INDEX	ISTK	8,465,380	16,384	135,446,080	6,772,304	0	142,218,384
ITEMCLUSTER	CLUSTER	ITEMS	8,446	2,048	16,892	845	0	17,737
IWARE	INDEX	WARE	43,012	2,048	86,024	4,301	0	90,325
NORDCLUSTER_QUEUE	CLUSTER	NORD	11,576,005	2,048	23,152,010	1,157,601	0	24,309,611
ORDRCLUSTER_QUEUE	CLUSTER	ORDR	185,474,424	16,384	2,967,590,784	0	596,759,200	3,564,349,984
STOCKCLUSTER	CLUSTER	STOK	972,533,769	2,048	1,945,067,538	97,253,377	0	2,042,320,915
SYS_IQ0000009627$$	INDEX	SYSTEM	236,245	2,048	472,490	23,625	0	496,115
SYS_IQ0000009636$$	INDEX	SYSTEM	458,376	16,384	7,334,016	366,701	0	7,700,717
SYSAUX	SYS	SYSTEM	61,440	2,048	122,880	0	0	122,880
SYSTEM	SYS	SYSTEM	4,300,800	2,048	8,601,600	0	0	8,601,600
WCLUSTER	CLUSTER	WARE	71,232	2,048	142,464	7,123	0	149,587
Total			1,968,560,951		6,995,157,868	190,416,597	638,669,520	7,824,243,985
Dynamic space(KB)		3,176,004,288						
Static space(KB)		4,009,570,177						
Free space(KB)		638,669,520						
Daily growth(KB)		638,669,520						
Daily spread		0	Oracle may be configured such that daily spread is 0					
60-day (GB)		40,368.79						
Log KB/tpmC		5.01	Number of log blocks used per New-Order					
8-hour log (GB)		1,816.60						
		Capacity	**NumDisk**				**Total GB**	
Database Disks : ETERNUS6000		33.25	1,536				51,072.00	
							51,072.00	
		Capacity	**NumDisk(RAID0+1)**					
8-Hr Log Disks : ETERNUS3000		66.75	56				3,738.00	
							3,738.00	

Appendix G: Price Quotes

Date: Mon, 10 April 2006 18:04
Subject: Oracle quote

Product	Price	Quantity	Extended Price
Oracle Database 10g Enterprise Edition, Unlimited Users for 3 years	20,000	32	640,000
Oracle Database Server Support Package for 3 years	2,000	3	6,000
Oracle Mandatory E-Business Discount			<129,200>
TOTAL			516,800

Oracle Pricing Contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081

March 24, 2006

Fujitsu Limited

Per your request I am enclosing the pricing information regarding TUXEDO 8.1 that you requested. This pricing applies to Tuxedo 6.4, 6.5, 7.1,8.0, 8.1 and 9.0. Please note that Tuxedo 9.0 is our most recent version of Tuxedo. Core functionality services (CFS)-R pricing is appropriate for your activities. Server systems are classified as either a Tier 1, 2, 3, 4 or 5 systems depending on the performance and CPU capacity of the system. The PRIMERGY RX200 S2 are Tier 1 machines – price is $1,200 per server (License), eligible for a 5% discount = $1,140 per server + $252 per server (7x24) for support – support is non discountable. This quote is valid for 60 days from the date of this letter.

Tuxedo Core Functionality Services (CFS-R) Program Product Pricing and Description

TUX-CFS-R provides a basic level of middleware support for distributed computing, and is best used by organizations with substantial resources and knowledge for advanced distributed computing implementations.

TUX-CFS-R prices are server only and are based on the overall performance characteristics of the server and uses the same five tier computer classification as TUXEDO 6.4, 6.5, 7.1,8.0, 8.1, and 9.0. Prices range from $1,200 for Tier 1 to $100,000 for Tier 5. Under this pricing option EVERY system running TUX-CFS-R at the user site must have a TUXEDO license installed and pay the appropriate per server license fees.

Very Truly Yours,

Robert J. Gieringer

Rob Gieringer,
Worldwide Pricing Director

BEA Tux/CFS-R Unlimited User License Fees Per Server

Unlimited User License fees per server	Number of Users	Dollar Amount	Maintenance (5 x 9) per year	Maintenance (7 x 24) per year
Tier 1 -- PC Servers with 1 or 2 CPUs, entry level RISC Uni-processor workstations and servers	Unlimited	$1,200.00	$216	$252
Tier 2 - PC Servers with 3 or 4 CPUs, Midrange RISC Uni-processor servers and workstations with up to 2 CPUs	Unlimited	$4,800.00	$864	$1,008
Tier 3 - Midrange Multiprocessors, up to 8 CPUs per system capacity	Unlimited	$12,000.00	$2,160	$2,520
Tier 4 - Large (more than 8, less than 32 CPUs)	Unlimited	$40,000.00	$7,200	$8,400
Tier 5 - Massively Parallel Systems, > 32 processors	Unlimited	$100,000.00	$18,000	$21,000


Cisco Catalyst 2950T-24 (WS-C2950T-24) - Hubs & Switches - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.costcentral.com/proddetail/Cisco_Catalyst_2950T_24/WSC2950T24/262620
CostCentral
a product of Erie Computer Company
my account | order tracking | shopping cart | customer care | checkout >>>
search GO | advanced search
Toll-Free 1-866-469-2678 (M-F 9AM-6PM EST)
HOME - Networking - Hubs & Switches - Cisco Systems
FREE SHIPPING ON ORDERS OVER $599
click here for more info
Welcome Guest - March 28, 2006 SIGN IN MY ACCOUNT
Your Cart
0 items $0.00
Add By Item / Mfg Part #
ADD
Live offline SUPPORT leave a message
Hubs & Switches by Brand
3Com (150)
ABL Electronics (2)
Actiontec Electronics (1)
Adaptec (6)
Addonics Technologies (1)
Adesso (2)
ADTRAN (27)
Allied Telesyn International (63)
APC (35)
APEX Computer Technology (10)
Apple Computer (1)
Asante Technologies (22)
ATEN Technology (65)
Avocent (42)
Axis Communications (5)
Belkin Components (96)
BUSlink (3)
Cables to Go (36)
Cisco Systems (125)
Compaq (3)
Comtrol (3)
ConnectPRO (13)
D-Link Systems (69)
Digi International (6)
Fellowes (2)
Hawking Technology (20)
Hewlett-Packard (111)
I-Rocks Technology (1)
IBM (42)
InFocus (2)
Intel (8)
IOGear (46)
Kensington Technology Group (9)
Lantronix (7)
Linksys (37)
Cisco Catalyst 2950T-24
MFG Part #: WS-C2950T-24
Cisco Systems
Cisco Catalyst 2950T-24 is a member of the Catalyst 2950 Series Intelligent Ethernet Switches, and is a fixed-configuration, standalone switch that provides wire-speed Fast Ethernet and Gigabit Ethernet connectivity for midsize networks and the metro access edge. The Catalyst 2950 Series is an affordable product line that brings intelligent services, such as enhanced security, high availability and advanced quality of service (QoS), to the network edge-while maintaining the simplicity of traditional LAN switching. When a Catalyst 2950 Switch is combined with a Catalyst 3550 Series Switch, the solution can enable IP routing from the edge to the core of the network.
Embedded in Catalyst 2950 Series switches is the Cisco Cluster Management Suite (CMS) Software, which allows users to simultaneously configure and troubleshoot multiple Catalyst desktop switches using a standard Web browser. In addition, Cisco CMS Software provides new configuration wizards that greatly simplify the implementation of converged applications and network-wide services.
MSRP $1,295.00
Our Price $719.00 FREE SHIPPING
Availability as of 11:11am 3/27/06 1707 In Stock Click to Verify Price and Stock
Condition New and Factory Sealed
Customer Reviews No reviews for this item. Be the first to rate it.
Quantity 1 ADD
Our company is a Cisco authorized sales center.
General Specs
Extended Specs
Data Sheet
Customer Reviews
General Specifications
Manufacturer Cisco Systems
Manufacturer Part # WS-C2950T-24
Cost Central Item # 262620
Product Description Catalyst 2950 24port Switch 10/100 + 2 10/100/1000 Enh Image
Device Type Switch - stackable
Form Factor Rack-mountable - 1 U
Approximate Dimensions (WxDxH) 17.5 in x 9.5 in x 1.7 in
Done


Cisco Catalyst 2950T-24 (WS-C2950T-24) - Hubs & Switches - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.costcentral.com/proddetail/Cisco_Catalyst_2950T_24/WSC2950T24/26262C
Go
Group (9)
•Lantronix (7)
•Linksys (37)
•Micro Innovations (30)
•NETGEAR (44)
•Nortel Networks (80)
•Quantum (1)
•Raritan Computer (12)
•Sharp (2)
•SIIG (17)
•SMC Networks (42)
•Sony (1)
•Targus (10)
•Toshiba (1)
•Tripp Lite (28)
•USRobotics (4)
•ZyXEL Communications (31)
Networking by Category
•Modems (406)
•Bridges & Routers (509)
•Concentrators & Multiplexers (17)
•Network Devices (665)
•Cables (1896)
•Network Accessories (449)
•Hubs & Switches (1374)
•Adapters (1977)
•Video Conferencing (11)
•Repeater/Transceiver (193)
•Telephones (23)
•Service & Support (344)
Approximate Dimensions (WxDxH) 17.5 in x 9.5 in x 1.7 in
Approximate Weight 6.6 lbs
RAM 16 MB SDRAM
Flash Memory 8 MB
Ports Qty 24 x Ethernet 10Base-T, Ethernet 100Base-TX
Data Link Protocol Ethernet, Fast Ethernet
Auxiliary Network Ports 2x10/100/1000Base-T(uplink)
Remote Management Protocol SNMP, RMON, Telnet
Communication Mode Half-duplex, full-duplex
Features Network monitoring, full duplex capability, uplink, VLAN support, manageable, stackable
Compliant Standards IEEE 802.3, IEEE 802.3U, IEEE 802.1D, IEEE 802.3ab, IEEE 802.1p, IEEE 802.3x, IEEE 802.3ad (LACP), IEEE 802.1w, IEEE 802.1x, IEEE 802.1s
Power AC 120/230 V (50/60 Hz)
Manufacturer Warranty Limited lifetime warranty
Available Accessories & Related Products
UPS & Power Devices
Description Mfg Part # Stock Price
Cisco Redundant Power System 675
Power supply - redundant (rack-mountable) - AC 110/240 V - 675 Watt - 6 output connector(s) - 1 U
PWR675-AC-RPS-N1= Yes $1,303.76
ADD
Network Adapters
Description Mfg Part # Stock Price
Cisco
Network adapter - GBIC - Gigabit EN - 1000Base-T
WS-G5483= Yes $279.90
ADD
Network Service & Support
Description Mfg Part # Stock Price
Cisco Comprehensive
Extended service agreement - 1 year - on-site
CON-OSP-PKG1 Yes $132.08
Cisco SMARTnet Category 1
Extended service agreement - maintenance - 1 year - 4 h
CON-SNTE-PKG1 Yes $94.66
Cisco SMARTnet Category 1
Extended service agreement - replacement - 1 year
CON-SNTP-PKG1 Yes $105.66
Done


Cisco SMARTnet Category 1 (CON-SNTP-PKG1) - Service & Support - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.costcentral.com/proddetail/Cisco_SMARTnet_Category_1/CONSNTPPKG1/
CostCentral
a product of Erie Computer Company
my account | order tracking | shopping cart | customer care | checkout >>>
search GO | advanced search
Toll-Free 1-866-469-2678 (M-F 9AM-6PM EST)
HOME - Networking - Service & Support - Cisco Systems
FREE SHIPPING ON ORDERS OVER $599
click here for more info
Welcome Guest - March 28, 2006 SIGN IN MY ACCOUNT
Your Cart
0 items $0.00
Add By Item / Mfg Part #
ADD
Live offline SUPPORT leave a message
Service & Support by Brand
Cisco Systems (256)
D-Link Systems (5)
SonicWALL (23)
Symantec (54)
ZyXEL Communications (6)
Networking by Category
Modems (406)
Bridges & Routers (509)
Concentrators & Multiplexers (17)
Network Devices (665)
Cables (1896)
Network Accessories (449)
Hubs & Switches (1374)
Adapters (1977)
Video Conferencing (11)
Repeater/Transceiver (193)
Telephones (23)
Service & Support (344)
Cisco SMARTnet Category 1
MFG Part #: CON-SNTP-PKG1
Cisco Systems
No image available
SMARTnet provides enhancement and maintenance support resources during the operational lifetime of your Cisco networking device. SMARTnet augments the resources of your operations staff; it provides them with access to a wealth of expertise, both on line and via telephone, the ability to refresh their system software at will, and a range of hardware advance replacement options.
The value of any networking device can be reduced to two simple criteria: availability and performance. As networks have become the backbone of business, the importance of these two factors has increased dramatically. And you are now responsible not only for keeping that network up and running at a reasonable and predictable cost but also increasing productivity throughout the company, increasing the quality of services delivered to your customers. SMARTnet provides remote support and onsite parts service, augmenting your staff's ability to maintain and operate your device, and ensuring that your device performs at its highest possible levels of performance and longevity.
MSRP $120.00
Our Price $105.66
Availability as of 11:11am 3/27/06 999 In Stock Click to Verify Price and Stock
Condition New and Factory Sealed
Customer Reviews No reviews for this item. Be the first to rate it.
Quantity 1 ADD
Our company is a Cisco authorized sales center.
Please note that this item is an electronic service agreement and not a boxed item. This item is sent via e-mail.
General Specs
Compatible Products
Customer Reviews
General Specifications
Manufacturer Cisco Systems
Manufacturer Part # CON-SNTP-PKG1
Cost Central Item # 671231
Product Description Maintenance 1yr 4hr 24x7 Smartnet Cat 1(esd)
Type Extended service agreement
Service Included Replacement
Full Contract Period 1 year
Done


Cisco SMARTnet Category 1 (CON-SNTP-PKG1) - Service & Support - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.costcentral.com/proddetail/Cisco_SMARTnet_Category_1/CONSNTPPKG1/
Go
Service Included Replacement
Full Contract Period 1 year
Response Time 4 hour(s)
Extended Specifications
General
Type Extended service agreement
Service Included Replacement
Full Contract Period 1 year
Response Time 4 hour(s)
Details
Service & Support Details
Extended service agreement - parts - 1 year - shipment - response time: 4 hours - availability: 24 hours a day / Monday-Sunday
Technical support - phone consulting - 1 year - availability: 24 hours a day / Monday-Sunday
New releases update - 1 year
Technical support - web knowledge base access - 1 year - availability: 24 hours a day / Monday-Sunday
Compatible With
Cisco
Cisco 1701 ADSL Security Access Router
Cisco 1710
Cisco 1711 Security Access Router
Cisco 1712 Security Access Router
Cisco 1721
Cisco 575
Cisco 576 LRE CPE
Cisco 585 LRE Customer Premise Equipment
Cisco 7914
Cisco 801
Cisco 802
Cisco 803
Cisco 804
Cisco 805
Cisco 806
Cisco 813
Cisco 827
Cisco 827-4V
Cisco 827H
Cisco 828
Cisco 831 Ethernet Broadband Router
Cisco 836 ADSL over ISDN Broadband Router
Cisco 837 ADSL Broadband Router
Cisco Aironet 1100
Cisco Aironet 1131AG
Cisco Aironet 1200
Cisco Aironet 1230
Cisco Aironet 1230AG
Cisco Aironet 1231
Cisco Aironet 1310 Outdoor Access Point/Bridge
Cisco Aironet 352
Cisco Aironet 352 Workgroup Bridge
Cisco Aironet 802.11a Radio Module
Cisco Aironet 802.11a/b/g Wireless CardBus Adapter
Cisco Aironet 802.11a/b/g Wireless PCI Adapter
Cisco Aironet Wireless LAN Client Adapter
Cisco ATA 186
Cisco ATA 188
Cisco Catalyst 2940
Cisco Catalyst 2950
Cisco Catalyst 2950-12
Cisco Catalyst 2950-24
Cisco Catalyst 2950SX-24
Cisco Catalyst 2950T-24
Cisco CiscoWorks Wireless LAN Solution Engine Express 2.11
Cisco IP Conference Station 7935
Cisco IP Conference Station 7936
Cisco IP Phone 7902G
Cisco IP Phone 7905G
Cisco IP Phone 7910G+SW
Cisco IP Phone 7912G
Cisco IP Phone 7920
Cisco IP Phone 7940G
Cisco IP Phone 7960G
Cisco IP Phone 7970G
Cisco PIX Firewall 501
Cisco SOHO 71
Cisco SOHO 77
Cisco SOHO 77H
Cisco SOHO 78
Cisco SOHO 91
Cisco SOHO 96
Cisco SOHO 97
Cisco VPN 3002 Hardware Client
Cisco VPN Module
Cisco VT Advantage
Hot Products - Contact Customer Support - SiteMap
Resources 1 2 3
© 2005 Copyright, Erie Computer Company and costcentral.
VERIFIED by VISA
Activate Now
ScanAlert
HACKER SAFE
TESTED 27-MAR
VeriSign Secure Site
Click to verify
BBB
RELIABILITY PROGRAM
BBBOnLine
MasterCard SecureCode.
0 - 0.309715
Done

Appendix H: Auditor's attestation letter

INFO SIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL CERTIFIED AUDITOR

Benchmark Sponsor: Shin'ichi Kurogi
Manager, Platform Solution Center
Fujitsu Limited
Shin-Yokohama Nikko Bldg.
2-15-16 Shin-Yokohama, Kohoku-ku, Yokohama
Kanagawa Pref. 222-0033, Japan

April 24, 2006

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

Platform: Fujitsu PRIMEQUEST 480 c/s
Operating system: Red Hat Enterprise Linux 4 AS
Database Manager: Oracle Database 10g Enterprise Edition
Transaction Manager: BEA Tuxedo 8.1

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: Fujitsu PRIMEQUEST 480				
32 x Itanium2 (1.6GHz)	1024 GB (9 MB L3)	1536 x 36 GB 15Krpm 56 x 73 GB 15Krpm 1 x 73 GB 10Krpm	0.71 Seconds	792,101.96
Thirty Nine Client: Fujitsu PRIMERGY RX200 S2 (each with)				
2 x Xeon DP (3.0 GHz)	3.0 GB (2 MB L2)	1 x 73 GB 10Krpm	n/a	n/a

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

The following verification items were given special attention:

- The transactions were correctly implemented
- The database records were the proper size

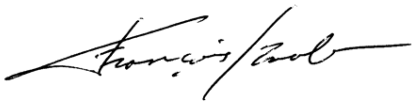
125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com

- 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 (32) priced clients model RX200 S2 and (7) non-priced clients model F250. The priced configuration includes (39) RX200 S2 systems. Based on data analysis done for each type of client, it is my opinion that this substitution has no significant effect on performance.

Respectfully Yours,



François Raab, President

125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com