



TPC Benchmark™C

Full Disclosure Report

*Fujitsu
PRIMEPOWER 2500 c/s
W/ 52 Front-Ends*

running

*Oracle Database 10g
Enterprise Edition*

Jan 6, 2004

The benchmark results contained in this document were submitted for compliance with version 5.1 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 Jan 6,2004.

Fujitsu believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. Fujitsu assumes no responsibility for any errors that may appear in this document.

The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, Fujitsu provides no warranty of the pricing information in this document.

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

All performance data contained in this report were obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. Fujitsu does not warrant or represent that a user can or will achieve similar performance expressed in transactions per minute (tpmC) or normalized price/performance (\$/tpmC). No warranty of system performance or price/performance is expressed or implied in this report.

Copyright (C) 2004 Fujitsu Limited. All rights reserved

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

Second Edition Jan 6, 2004

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

SPARC64 is a trademark of Fujitsu Limited used under license from SPARC International, Inc. of the U.S.A. PRIMERGY TX200 is a trademark of Fujitsu Limited.

Solaris is a trademark of Sun Microsystems, Inc.

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

Intel Xeon, Pentium(R)III, Pentium(R)III XEON are trademarks or registered trademarks of Intel Corporation.

Microsoft, Windows, Windows 2000 are registered trademarks of Microsoft Corporation.

Veritas is a trademark or registered trademark of Veritas Corporation.

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

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 5.1, released February 11th, 2003.

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 PRIMEPOWER 2500 c/s w/ 52 Front-Ends. The operating system used for the benchmark was Solaris 8. The DBMS used was Oracle Database 10g Enterprise Edition .

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:

595,702.31 tpmC

\$12.43 per tpmC


April 30, 2004

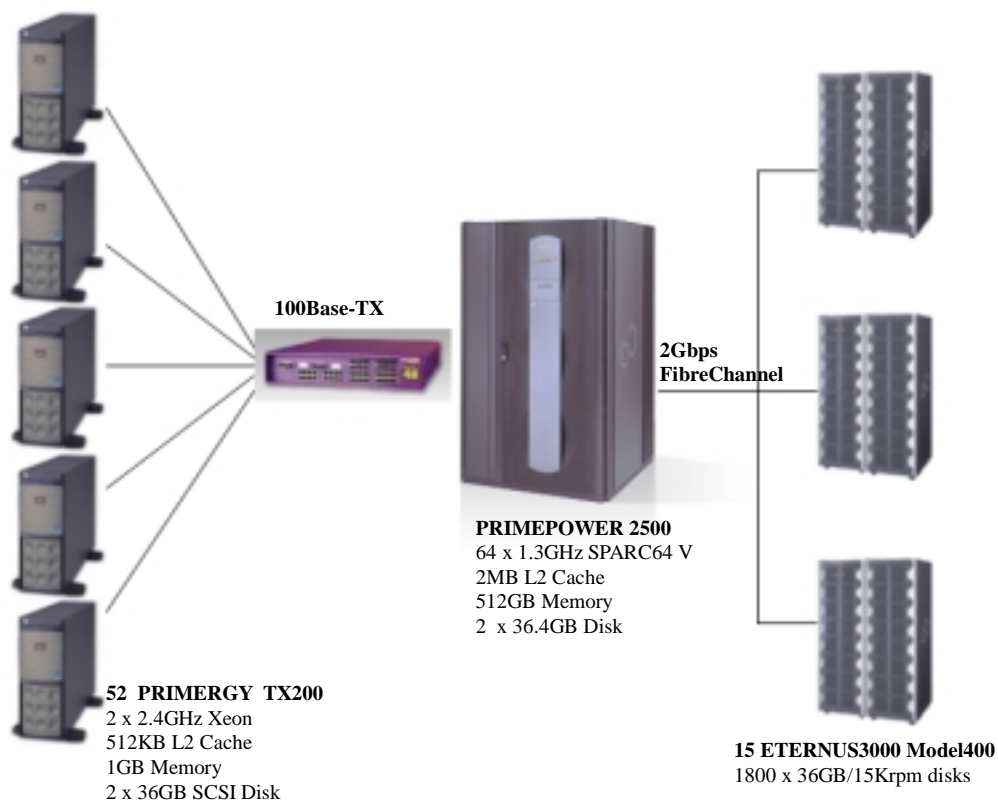
Standard and Executive Summary Statements

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.

		PRIMEPOWER 2500 c/s w/52 Front-Ends		TPC-C Rev 5.1
				Report Date: Jan 6, 2004
Total System Cost		TPC-C Throughput	Price/Performance	Availability Date
\$7,399,502		595,702.31 tpmC	\$12.43/tpmC	April 30,2004
Processors	Database Manager	Operating system	Other Software	Number of users
64 SPARC64 V @ 1.3GHz	Oracle Database 10g Enterprise Edition	Solaris 8	Microsoft IIS 5.0, Veritas Volumegr.3.5.0, Microsoft COM+	480,000



System Components	Server (PRIMEPOWER 2500)		each Client(52 PRIMERGY TX200)	
	Quantity	Description	Quantity	Description
Processor	1	64 x SPARC64 V @ 1.3GHz	52	2 x Intel Xeon 2.4GHz
Cache Memory		2 MB (each processor)		512 KB (each processor)
Memory		512GB		1 GB
Disk Controllers	60	2GBPS Fiber Chennel	1	SCSI controller
Disk Drives	1800 2	36GB 15K rpm Disk 36.4GB 10K rpm Disk	2	36 GB SCSI Disk
Total Storage		57,636.06 GB		72 GB
Terminal	1	Console Terminal	1	15" Monitor



**PRIMEPOWER 2500 c/s
w/52 Front-Ends**

TPC-C Rev 5.1
Report Date: Jan 6, 2004

Server Hardware	Part Number	Qty	Source	Unit Price	Ext. Price	Maint. rate/unit	3 Yr. Maint.
PRIMEPOWER 2500 w/ 8 x SPARC64 V 1.3GHz/2MB, 2 UltraSCSI/D Cards	PW2KBP2S1U	1	1	1,065,000.00	1,065,000.00	/3yr	457,776.00
System Board	PW2K7SB11U	16	1	37,500.00	600,000.00		
CPU Module (SPARC64 V 1.3GHz/2MB x 2)	PW2K1B12U	32	1	42,510.00	1,360,320.00		
Additional Memory (8GB)	PW2K2M41U	64	1	14,800.00	947,200.00		
Additional Disk Unit (36.4GB)	PW0R3D24	2	1	1,000.00	2,000.00		
PCI / Disk Box	PW2K7BD1U	7	1	27,500.00	192,500.00		
I/O Rack Unit	PW2K7RK3U	1	1	24,500.00	24,500.00		
I/O Rack Unit Connection kit (Basic Unit)	PW2K7RK93U	1	1	9,500.00	9,500.00		
Fibre Channel Card(2Gb)	PW008FC2U	60	1	2,795.00	167,700.00		
Fibre Channel cable	DCL-FCA20U	60	1	780.00	46,800.00		
PRIMEPOWER 100 for System Console	PW087MC4AU	1	1	13,500.00	13,500.00		
Server Hardware Subtotals					4,429,020.00		457,776.00
Storage							
ETERNUS3000 Model400 floor stand(rack without door)	E340S20AU	15	1	35,490.00	532,350.00	3,550.00 /yr*2	106,500.00
Additional drive Enclosure	E300CE1U	45	1	7,172.00	322,740.00	715.00 /yr*2	64,350.00
Additional cache memory(4GB)	E340CM4	15	1	9,034.00	135,510.00	900.00 /yr*2	27,000.00
36GB/15Krpm disk drive	E300CA3H	1800	1	662.00	1,191,600.00	65.00 /yr*2	234,000.00
Storage Subtotals					2,182,200.00		431,850.00
Server Software							
Multilingual Solaris(TM) 8 2/02 OE Media for PRIMEPOWER (Rev.4)	B23PT40H0H	1	1	100.00	100.00		
Oracle Database 10g Enterprise Edition for 3 years, Unlimited Users		64	2	20,000.00	1,280,000.00		
Oracle Database Server Support Package for 3 years		1	2	6,000.00		/3yr	6,000.00
Volume Manager, Solaris, V3.5, License, Tier 4C	A08974F-M00000	1	3	39,126.75	39,126.75		
Volume Manager, Solaris, V3.5, Tier 4C Extended Support	W08974F-M00212	3	3			/3yr	35,304.75
Storage Solution, Solaris V3.5 Maintenance Pack 1, English	N09665F	1	3	65.00	65.00		
Volume Manager, Solaris, Administrator's Guide	N08836F	1	3	32.50	32.50		
Server Software Subtotals					1,319,324.25		41,304.75
Client Hardware							
PRIMERGY PY TX200f X/2.4/512 512M	S26361-K880-V102	52	1	1,466.00	76,232.00		
Processor XEON DP 2.4GHz/512K/533MHz	S26361-F2827-E240	52	1	450.00	23,400.00		
Mem 512MB DDR-RAM PC2100 1p ECC	S26361-F2762-E523	52	1	290.00	15,080.00		
CD-ROM, ATAPI	SNP:SY-F2240E1-A	52	1	35.00	1,820.00		
HDD U320 10k 36GB hot plug 1"	S26361-F2764-E136	104	1	269.00	27,976.00		
Fast Ethernet Adapter 10/100TX	S26361-F2643-E1	104	1	74.00	7,696.00		
Cable powercord(USA), 1.8m, grey	T26139-Y1742-E10	52	1	9.00	468.00		
Monitor CRT 15" SCENICVIEW 151E	S26361-K819-V150	52	1	127.00	6,604.00		
Keyboard KBPC USB US	S26361-K280-V110	52	1	42.00	2,184.00		
Client Hardware Subtotals					161,460.00		0.00
Client Software							
Microsoft Windows 2000 Server		52	1	673.00	34,996.00	/3yr	21,800.00
Microsoft Visual C++ Standard		1	1	102.00	102.00	/3yr	480.00
Client Software Subtotals					35,098.00		22,280.00
User Connectivity							
ExtremeNetworks Summit48i(15501)	LFE15501	3	4	8,240.04	24,720.12	/3yr	18,394.65
GBIC Module(10011)	LFE10011	3	4	461.57	1,384.71		
User Connectivity Subtotals					26,104.83		18,394.65
Total					8,153,207.08		971,605.40
Oracle Mandatory E-Business Discount		1	2	(321,500.00)	(321,500.00)		
Large Configuration Discount and Support Prepayment*					(1,328,736.00)		(75,075.26)
Three-Year Cost of Ownership							\$7,399,502
Audited by: Francois Raab, InfoSizing, Inc. (www.sizing.com)							tpmC
Pricing Sources:							595,702.31
1 = Fujitsu , 2 = Oracle , 3 = Veritas , 4 = DreamRetail, Inc							\$ / tpmC
*All discounts are based on US list prices and for similar quantities and configurations							\$12.43
*2: 1 year warranty included							

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							
PRIMEPOWER 2500 c/s w/ 52 Front-Ends							
Oracle Database 10g Enterprise Edition							
MQTH, Computed Maximum Qualified Throughput				595,702.31 tpmC			
Response Times (in seconds)		Average		90%		Max.	
New-Order		0.598		0.870		38.889	
Payment		0.564		0.850		34.921	
Order-Status		0.561		0.850		27.331	
Delivery (interactive portion)		0.131		0.120		22.309	
Delivery (deferred portion)		0.354		0.550		25.093	
Stock-Level		0.544		0.830		29.641	
Menu		0.142		0.120		22.322	
Transaction Mix, in percent of total transaction							
New-Order						44.91	
Payment						43.04	
Order-Status						4.02	
Delivery						4.02	
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		18.01 0.01		18.02 12.02		18.18 120.11	
Payment		3.02 0.01		3.03 12.01		3.32 120.11	
Order-Status		2.00 0.02		2.01 10.02		2.13 100.20	
Delivery (interactive)		2.00 0.01		2.01 5.02		2.15 50.10	
Stock-Level		2.00 0.01		2.01 5.02		2.12 50.09	
Test Duration							
Ramp-up time (seconds)						2,640	
Measurement interval						7,200	
Transactions during measurement interval						71,484,279	
Checkpointing							
Number of checkpoints						4	
Checkpoint interval						1,775	

Numerical Quantities for each client

Client	PRIMERGY F250									
	c1004	c1005	c1007	c1008	c1010	c1011	c1013	c1014	c1016	c1017
tpmC	11485.92	11406.14	11436.04	11428.29	11392.11	11494.76	11409.31	11513.23	11506.42	11480.22
menu-ave	0.138	0.136	0.132	0.130	0.142	0.137	0.138	0.140	0.134	0.141
menu-ninety	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120
new-ave	0.548	0.689	0.659	0.633	0.705	0.524	0.682	0.486	0.520	0.534
new-ninety	0.780	1.020	0.980	0.950	1.020	0.750	0.980	0.690	0.750	0.760
new-think	12.003	12.015	12.017	11.998	12.020	12.017	12.010	11.998	11.999	12.015
new-keying	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.020
new-mix	44.890	44.910	44.940	44.810	44.860	44.920	44.860	44.890	44.940	44.900
pay-ave	0.512	0.665	0.629	0.613	0.666	0.496	0.641	0.453	0.492	0.511
pay-ninety	0.760	1.000	0.960	0.920	1.000	0.720	0.960	0.660	0.730	0.740
pay-think	11.991	12.021	11.994	12.001	12.013	12.011	12.013	12.009	12.009	12.015
pay-keying	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.030
pay-mix	43.070	43.050	43.010	43.130	43.070	43.040	43.050	43.050	43.010	43.060
ord-ave	0.508	0.663	0.627	0.611	0.665	0.491	0.641	0.452	0.488	0.508
ord-ninety	0.750	0.990	0.960	0.920	1.000	0.720	0.960	0.660	0.730	0.730
ord-think	10.015	10.027	10.021	10.013	10.033	9.971	10.000	10.010	10.006	10.012
ord-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010
ord-mix	4.020	4.010	4.020	4.010	4.020	3.990	4.020	4.010	4.020	4.010
del-ave	0.126	0.130	0.123	0.127	0.129	0.129	0.126	0.128	0.127	0.136
del-ninety	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120
del-think	5.015	4.998	5.028	5.001	5.005	5.018	5.010	5.001	5.015	5.012
del-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010
del-mix	4.010	4.020	4.020	4.030	4.030	4.030	4.040	4.010	4.010	4.010
stk-ave	0.493	0.644	0.609	0.594	0.647	0.476	0.620	0.432	0.470	0.493
stk-ninety	0.740	0.970	0.940	0.900	0.980	0.700	0.950	0.640	0.700	0.720
stk-think	5.004	5.005	5.006	5.028	4.995	5.024	5.022	5.021	5.009	5.017
stk-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010
stk-mix	4.010	4.020	4.010	4.030	4.020	4.010	4.040	4.040	4.030	4.020
ntxn	1378311	1368737	1372325	1371395	1367053	1379371	1369117	1381588	1380771	1377626

PRIMERGY C200 (Unpriced clients)										
Client	c1001	c1002	c1019	c1020	c1022	c1023	c1025	c1026	c1028	c1029
tpmC	11422.28	11476.02	11348.78	11451.83	11377.91	11443.72	11428.99	11467.00	11410.33	11469.55
menu-ave	0.141	0.142	0.145	0.141	0.146	0.136	0.143	0.146	0.143	0.144
menu-ninety	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120
new-ave	0.653	0.579	0.825	0.603	0.743	0.621	0.644	0.571	0.680	0.558
new-ninety	0.950	0.820	1.200	0.870	1.080	0.920	0.930	0.800	0.990	0.800
new-think	12.009	12.012	12.006	12.026	12.027	11.992	12.022	12.000	11.988	12.016
new-keying	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.020
new-mix	44.880	44.950	44.930	44.920	44.910	44.910	44.900	44.900	44.890	44.900
pay-ave	0.615	0.535	0.791	0.572	0.705	0.603	0.605	0.541	0.652	0.527
pay-ninety	0.930	0.800	1.180	0.850	1.060	0.900	0.900	0.770	0.970	0.770
pay-think	12.011	11.999	11.999	12.016	12.007	12.025	12.008	12.012	12.014	12.014
pay-keying	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.030
pay-mix	43.060	43.030	43.020	43.020	43.030	43.000	43.040	43.010	43.050	43.060
ord-ave	0.615	0.530	0.789	0.569	0.706	0.601	0.598	0.539	0.650	0.524
ord-ninety	0.930	0.790	1.180	0.840	1.060	0.890	0.900	0.770	0.970	0.770
ord-think	10.029	9.992	9.938	9.962	10.008	10.037	9.986	10.028	10.024	10.033
ord-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010
ord-mix	4.030	4.010	4.000	4.000	4.030	4.030	4.020	4.030	4.040	4.030
del-ave	0.129	0.124	0.132	0.129	0.135	0.134	0.127	0.136	0.135	0.134
del-ninety	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120
del-think	5.039	5.016	5.004	4.995	5.007	5.016	5.013	5.017	5.026	4.991
del-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010
del-mix	4.000	4.000	4.040	4.030	4.000	4.020	4.020	4.040	4.010	4.000
stk-ave	0.596	0.510	0.773	0.553	0.689	0.588	0.585	0.521	0.634	0.506
stk-ninety	0.910	0.780	1.160	0.830	1.050	0.880	0.890	0.750	0.950	0.750
stk-think	5.001	5.016	5.010	5.009	5.024	4.997	5.013	5.019	5.012	5.005
stk-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010
stk-mix	4.020	4.010	4.020	4.030	4.020	4.030	4.020	4.020	4.010	4.000
ntxn	1370673	1377122	1361853	1374219	1365349	1373247	1371479	1376040	1369240	1376346

PRIMERGY C200 (Unpriced clients)										
Client	c1031	c1032	c1034	c1035	c1037	c1038	c1040	c1041	c1043	c1044
tpmC	11357.98	11470.82	11517.05	11498.98	11477.07	11523.14	11477.39	11530.58	11484.74	11467.88
menu-ave	0.151	0.140	0.131	0.147	0.148	0.146	0.144	0.136	0.138	0.152
menu-ninety	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120
new-ave	0.778	0.569	0.493	0.524	0.545	0.460	0.539	0.450	0.538	0.572
new-ninety	1.110	0.810	0.710	0.730	0.770	0.630	0.740	0.660	0.770	0.800
new-think	12.019	12.011	12.006	12.001	12.015	12.002	12.010	12.034	12.001	12.002
new-keying	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.020
new-mix	44.900	44.910	44.950	44.960	44.910	44.880	44.860	44.930	44.870	44.940
pay-ave	0.740	0.534	0.468	0.488	0.515	0.432	0.493	0.431	0.501	0.536
pay-ninety	1.090	0.780	0.680	0.700	0.740	0.610	0.720	0.640	0.740	0.770
pay-think	12.010	12.009	12.020	12.013	12.012	11.993	12.018	11.997	12.004	12.028
pay-keying	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.030
pay-mix	43.050	43.030	43.010	43.000	43.020	43.050	43.070	43.030	43.080	42.980
ord-ave	0.737	0.530	0.460	0.485	0.511	0.423	0.489	0.431	0.497	0.530
ord-ninety	1.090	0.780	0.680	0.700	0.740	0.600	0.720	0.630	0.740	0.770
ord-think	9.974	10.081	9.993	10.018	10.036	10.022	10.034	10.016	10.063	10.014
ord-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010
ord-mix	4.000	4.000	4.020	4.040	4.030	4.020	4.020	4.020	4.030	4.030
del-ave	0.137	0.128	0.125	0.135	0.140	0.136	0.128	0.132	0.126	0.138
del-ninety	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120
del-think	5.028	5.016	5.023	5.029	4.995	5.004	5.002	5.023	5.018	5.004
del-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010
del-mix	4.020	4.020	4.010	4.010	4.000	4.030	4.040	4.010	4.020	4.040
stk-ave	0.717	0.511	0.447	0.469	0.493	0.410	0.472	0.410	0.478	0.515
stk-ninety	1.070	0.770	0.660	0.680	0.720	0.580	0.700	0.620	0.720	0.760
stk-think	4.994	5.017	5.011	5.013	5.007	5.029	5.017	4.986	4.993	5.003
stk-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010
stk-mix	4.030	4.040	4.010	4.000	4.040	4.020	4.020	4.000	4.000	4.010
ntxn	1362958	1376498	1382046	1379878	1377248	1382777	1377287	1383670	1378169	1376145

PRIMERGY C200 (Unpriced clients)										
Client	c1046	c1047	c1049	c1050	c1052	c1053	c1055	c1056	c1058	c1059
tpmC	11481.31	11469.03	11459.36	11475.82	11454.84	11508.53	11452.05	11463.82	11485.90	11434.41
menu-ave	0.145	0.143	0.148	0.147	0.144	0.116	0.149	0.135	0.135	0.152
menu-ninety	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120
new-ave	0.539	0.558	0.599	0.564	0.590	0.518	0.601	0.587	0.530	0.622
new-ninety	0.770	0.790	0.850	0.770	0.810	0.800	0.850	0.860	0.770	0.880
new-think	12.018	12.030	12.018	12.016	12.020	12.003	12.022	12.000	12.017	12.004
new-keying	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.020
new-mix	44.920	44.900	44.950	44.920	44.900	44.910	44.930	44.900	44.880	44.880
pay-ave	0.515	0.526	0.565	0.525	0.539	0.502	0.558	0.552	0.502	0.586
pay-ninety	0.750	0.760	0.830	0.750	0.790	0.780	0.820	0.840	0.740	0.860
pay-think	12.005	12.008	11.992	11.991	12.016	12.008	12.014	12.019	12.009	12.013
pay-keying	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.030
pay-mix	43.030	43.050	43.010	43.030	43.080	43.020	43.000	43.030	43.050	43.060
ord-ave	0.513	0.525	0.558	0.519	0.532	0.502	0.556	0.549	0.500	0.584
ord-ninety	0.750	0.760	0.830	0.750	0.790	0.770	0.820	0.830	0.740	0.850
ord-think	10.034	9.985	10.030	10.048	10.041	10.012	10.003	10.032	9.991	10.028
ord-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010
ord-mix	4.020	4.010	4.010	4.010	4.010	4.030	4.020	4.010	4.040	4.010
del-ave	0.139	0.133	0.136	0.134	0.126	0.116	0.133	0.124	0.128	0.138
del-ninety	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120
del-think	5.028	4.990	5.015	5.004	5.023	4.982	5.006	5.005	5.027	5.000
del-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010
del-mix	4.010	4.030	4.020	4.020	4.010	4.000	4.030	4.030	4.010	4.010
stk-ave	0.487	0.503	0.545	0.504	0.518	0.487	0.536	0.530	0.484	0.560
stk-ninety	0.730	0.740	0.810	0.730	0.770	0.760	0.800	0.810	0.720	0.830
stk-think	5.017	5.022	5.055	4.984	5.025	4.991	5.030	5.014	5.011	5.009
stk-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010
stk-mix	4.020	4.020	4.010	4.030	4.010	4.030	4.020	4.020	4.020	4.040
ntxn	1377757	1376283	1375123	1377098	1374581	1381023	1374246	1375658	1378308	1372129

PRIMERGY C200 (Unpriced clients)										
Client	c1061	c1062	c1064	c1065	c1067	c1068	c1070	c1071	c1073	c1075
tpmC	11397.21	11421.41	11434.79	11404.22	11425.39	11468.28	11449.06	11458.62	11436.13	11473.78
menu-ave	0.139	0.142	0.146	0.153	0.149	0.135	0.138	0.141	0.145	0.151
menu-ninety	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120
new-ave	0.706	0.655	0.644	0.691	0.664	0.579	0.628	0.595	0.629	0.529
new-ninety	1.030	0.930	0.920	0.980	0.940	0.840	0.900	0.850	0.910	0.740
new-think	12.016	12.018	12.008	12.020	12.015	12.021	12.001	11.994	12.017	12.015
new-keying	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.020	18.021	18.021
new-mix	44.870	44.850	44.930	44.910	44.940	44.920	44.940	44.880	44.890	44.880
pay-ave	0.670	0.614	0.604	0.653	0.625	0.552	0.594	0.554	0.592	0.491
pay-ninety	1.010	0.910	0.900	0.960	0.920	0.820	0.880	0.830	0.890	0.710
pay-think	12.001	11.989	12.018	11.999	11.997	12.010	12.015	12.009	11.994	12.030
pay-keying	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.030	3.031	3.031
pay-mix	43.040	43.080	43.000	43.020	43.000	42.990	43.000	43.050	43.080	43.060
ord-ave	0.669	0.612	0.603	0.655	0.623	0.550	0.588	0.551	0.588	0.491
ord-ninety	1.010	0.910	0.890	0.960	0.920	0.810	0.880	0.830	0.890	0.710
ord-think	10.036	10.012	10.015	10.027	10.049	9.964	10.017	10.009	9.971	10.051
ord-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.011	2.011
ord-mix	4.030	4.000	4.030	4.020	4.020	4.030	4.040	4.040	4.030	4.010
del-ave	0.128	0.130	0.130	0.140	0.135	0.128	0.123	0.126	0.133	0.138
del-ninety	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120
del-think	5.009	5.006	5.014	5.013	5.009	5.000	5.001	5.016	5.010	4.988
del-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.011	2.011
del-mix	4.030	4.030	4.020	4.010	4.020	4.030	4.020	4.010	4.010	4.020
stk-ave	0.651	0.595	0.584	0.631	0.599	0.534	0.571	0.532	0.569	0.470
stk-ninety	0.990	0.890	0.880	0.940	0.900	0.800	0.860	0.810	0.870	0.690
stk-think	5.022	5.032	5.036	5.012	5.023	5.019	5.025	5.019	5.001	5.041
stk-keying	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.010	2.011	2.011
stk-mix	4.030	4.050	4.020	4.040	4.030	4.030	4.000	4.020	4.000	4.030
ntxn	1367665	1370569	1372175	1368507	1371047	1376193	1373887	1375034	1372336	1376853

	PRIMERGY C200	
Client	c1077	c1079
tpmC	11503.95	11489.96
menu-ave	0.141	0.145
menu-ninety	0.120	0.120
new-ave	0.508	0.629
new-ninety	0.710	0.910
new-think	12.015	12.003
new-keying	18.021	18.021
new-mix	44.940	44.920
pay-ave	0.476	0.600
pay-ninety	0.690	0.890
pay-think	12.012	12.010
pay-keying	3.031	3.031
pay-mix	42.990	43.030
ord-ave	0.471	0.600
ord-ninety	0.680	0.890
ord-think	10.035	9.990
ord-keying	2.011	2.011
ord-mix	4.030	4.010
del-ave	0.130	0.135
del-ninety	0.120	0.120
del-think	5.031	5.009
del-keying	2.011	2.011
del-mix	4.030	4.010
stk-ave	0.453	0.582
stk-ninety	0.660	0.870
stk-think	4.999	5.021
stk-keying	2.011	2.011
stk-mix	4.010	4.030
ntxn	1380474	1378795

Table Of Contents

PREFACE	I
TPC BENCHMARK C OVERVIEW	I
ABSTRACT	III
OVERVIEW	III
TPC BENCHMARK C METRICS	III
STANDARD AND EXECUTIVE SUMMARY STATEMENTS	III
AUDITOR	III
NUMERICAL QUANTITIES SUMMARY	VI
TABLE OF CONTENTS	XIII
GENERAL ITEMS	17
APPLICATION CODE AND DEFINITION STATEMENTS.....	17
TEST SPONSOR.....	17
PARAMETER SETTINGS	17
CONFIGURATION ITEMS.....	18
1.1 TABLE DEFINITIONS	20
1.2 PHYSICAL ORGANIZATION OF DATABASE	20
1.3 INSERT AND DELETE OPERATIONS	20
1.4 PARTITIONING.....	20
1.5 REPLICATION, DUPLICATION OR ADDITIONS	20
CLAUSE 2 RELATED ITEMS.....	21
2.1 RANDOM NUMBER GENERATION	21
2.2 INPUT/OUTPUT SCREEN LAYOUT	21
2.3 PRICED TERMINAL FEATURE VERIFICATION	21
2.4 PRESENTATION MANAGER OR INTELLIGENT TERMINAL	21
2.5 TRANSACTION STATISTICS.....	22
2.6 QUEUEING MECHANISM.....	22

CLAUSE 3 RELATED ITEMS	23
3.1 TRANSACTION SYSTEM PROPERTIES (ACID)	23
3.2 ATOMICITY	23
3.3 CONSISTENCY	24
3.4 ISOLATION	24
3.5 DURABILITY	25
CLAUSE 4 RELATED ITEMS	27
4.1 INITIAL CARDINALITY OF TABLES	27
4.2 DATABASE LAYOUT	28
4.3 TYPE OF DATABASE	33
4.4 DATABASE MAPPING	33
4.5 60 DAY SPACE	33
CLAUSE 5 RELATED ITEMS	35
5.1 THROUGHPUT	35
5.2 RESPONSE TIMES	35
5.3 KEYING AND THINK TIMES	36
5.4 RESPONSE TIME FREQUENCY DISTRIBUTION CURVES AND OTHER GRAPHS	36
5.5 STEADY STATE DETERMINATION	41
5.6 WORK PERFORMED DURING STEADY STATE	41
5.7 REPRODUCIBILITY	42
5.8 MEASUREMENT PERIOD DURATION	42
5.9 REGULATION OF TRANSACTION MIX	42
5.10 TRANSACTION STATISTICS	43
5.11 CHECKPOINT COUNT AND LOCATION	43
CLAUSE 6 RELATED ITEMS	45
6.1 RTE DESCRIPTIONS	45
6.2 EMULATED COMPONENTS	45
6.3 FUNCTIONAL DIAGRAMS	45
6.4 NETWORKS	46
6.5 OPERATOR INTERVENTION	46
CLAUSE 7 RELATED ITEMS	47
7.1 SYSTEM PRICING	47
7.2 AVAILABILITY	47
7.4 COUNTRY SPECIFIC PRICING	48
7.5 USAGE PRICING	48

CLAUSE 9 RELATED ITEMS.....49

9.1 AUDITOR’S REPORT.....49

9.2 AVAILABILITY OF THE FULL DISCLOSURE REPORT.....49

APPENDIX A: CLIENT SOURCE CODE.....51

APPENDIX B: SERVER SOURCE CODE.....95

APPENDIX C: RTE SCRIPTS155

APPENDIX D: SYSTEM TUNABLES171

APPENDIX E: DATABASE CREATION CODE191

APPENDIX F: 60 DAY SPACE CALCULATION.....259

APPENDIX G: PRICE QUOTES.....260

APPENDIX H: AUDITOR’S ATTESTATION LETTER.....265

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.*
- This requirement can be satisfied by providing a full list of all parameters.*

Appendix D contains the parameters for the database, the operating system, and the configuration for the transaction monitor.

Configuration Items

Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

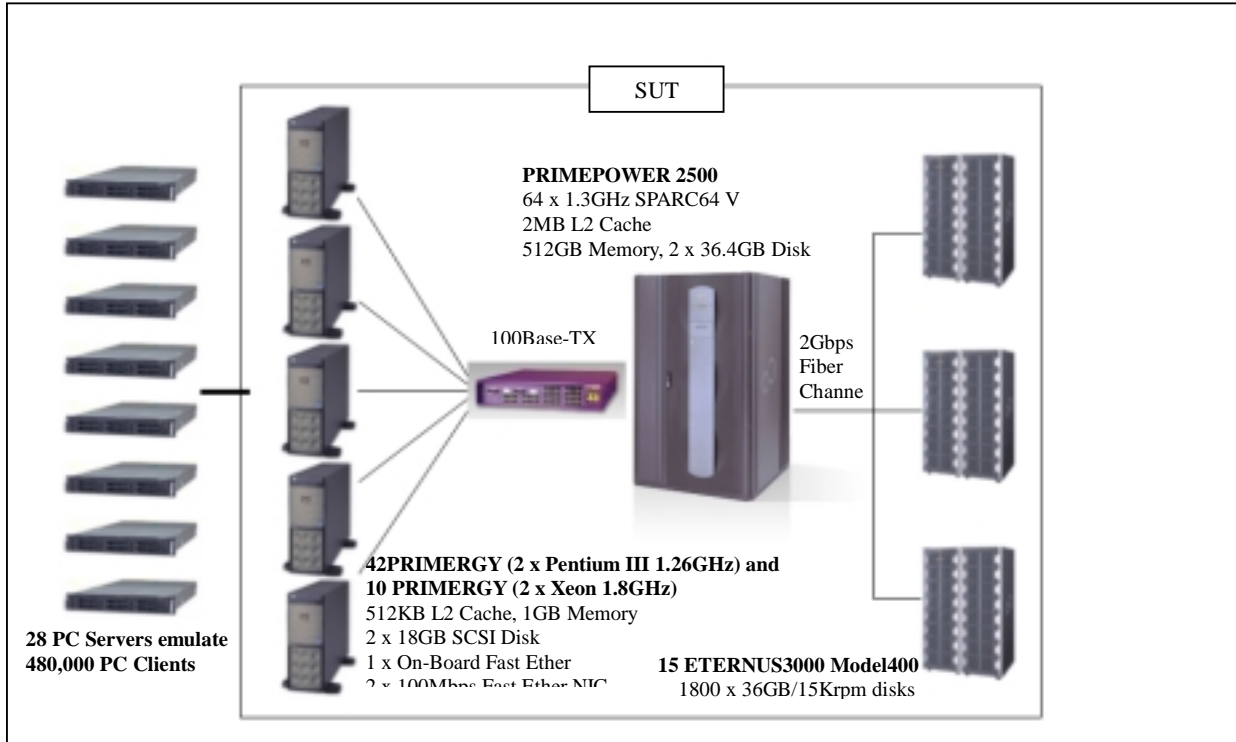
The System Under Test (SUT), a PRIMEPOWER 2500 c/s w/ 52 Front-Ends, is depicted in the following diagrams.

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

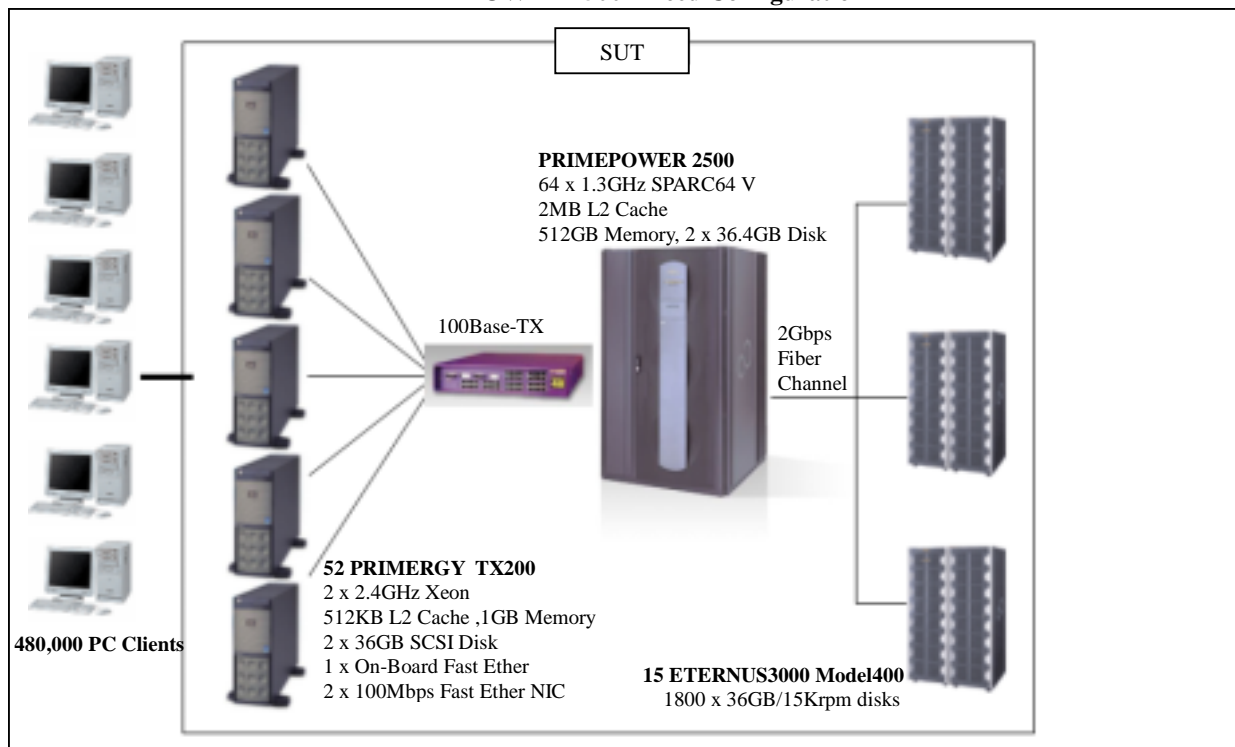
There were differences between the priced and measured configurations. The differences are:

- A RTE was used in the tested configuration.
- The clients that use Pentium III@1.26GHz and Xeon@1.8GHz in the measurement were replaced by those that use Xeon@2.4BGHz in the priced configuration.

PRIMEPOWER 2500 Tested Configuration



PRIMEPOWER 2500 Priced Configuration



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

No replications, duplications or additional attributes were used in this 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 process id. Each RTE machine was given a number incremented by 30,000. The process id was appended to this number to ensure uniqueness across all RTE machines. These seeds were printed to a file and verified by the auditor to be unique.

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 Statistics

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

Table 2.1 Transaction Statistics

Statistic		Value
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse	85.00%
	Remote warehouse	15.00%
	Accessed by last name	59.99%
Order Status	Accessed by last name	60.00%
Delivery	Skipped transactions	None
Transaction Mix	New Order	44.91%
	Payment	43.04%
	Order status	4.02%
	Delivery	4.02%
	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 are queued by the list in the application process (tpapl.dll) running on the client system. And they were asynchronously processed by the background threads of the application.

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 as specified in Clause 2.5.1.2) 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 as specified in Clause 2.5.1.2) and substitute a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the records in 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

Isolation can be defined in terms of phenomena that can occur during the execution of concurrent transactions. These phenomena are P0 (“Dirty Write”), P1 (“Dirty Read”), P2 (“non-repeatable Read”), and P3 (“Phantom”). The table in Clause 3.4.1 of the TPC-C specifications defines the isolation requirements which must be met by the TPC-C transactions. 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 Durable Media Failure

3.5.1.1 Loss of Log And Data

To demonstrate recovery from a permanent failure of durable medial containing the Oracle recovery log data and TPC-C tables, the following steps were executed using 14000 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 140,000 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 RAID5, 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.
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 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 48,000 warehouses under a full load of 480,000 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 480,000 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, must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table were deleted, 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 51,000 warehouses.

Table 4.1 Number of Rows for Server

Table	Occurrences
Warehouse	51,000
District	510,000
Customer	1,530,000,000
History	1,530,000,000
Order	1,530,000,000
New Order	459,000,000
Order Line	15,300,804,510
Stock	5,100,000,000
Item	100,000

4.2 Database Layout

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

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

datafile name	device name	RAID Level	number of drives
stok_0_0	/dev/rdisk/c210t11d8s3	0	5
stok_0_1	/dev/rdisk/c210t11d9s3	0	5
stok_0_2	/dev/rdisk/c210t11d10s3	0	5
stok_0_3	/dev/rdisk/c210t11d11s3	0	5
stok_0_4	/dev/rdisk/c205t11d6s3	0	5
stok_0_5	/dev/rdisk/c205t11d7s3	0	5
stok_0_6	/dev/rdisk/c205t11d8s3	0	5
stok_0_7	/dev/rdisk/c205t11d9s3	0	5
stok_0_8	/dev/rdisk/c205t11d10s3	0	5
stok_0_9	/dev/rdisk/c205t11d11s3	0	5
stok_0_10	/dev/rdisk/c166t12d12s3	0	5
stok_0_11	/dev/rdisk/c166t12d13s3	0	5
stok_0_12	/dev/rdisk/c166t12d14s3	0	5
stok_0_13	/dev/rdisk/c166t12d15s3	0	5
stok_0_14	/dev/rdisk/c166t12d16s3	0	5
stok_0_15	/dev/rdisk/c166t12d17s3	0	5
stok_0_16	/dev/rdisk/c162t12d12s3	0	5
stok_0_17	/dev/rdisk/c162t12d13s3	0	5
stok_0_18	/dev/rdisk/c162t12d14s3	0	5
stok_0_19	/dev/rdisk/c162t12d15s3	0	5
stok_0_20	/dev/rdisk/c162t12d16s3	0	5
stok_0_21	/dev/rdisk/c162t12d17s3	0	5
stok_0_22	/dev/rdisk/c140t12d12s3	0	5
stok_0_23	/dev/rdisk/c140t12d13s3	0	5
stok_0_24	/dev/rdisk/c140t12d14s3	0	5
stok_0_25	/dev/rdisk/c140t12d15s3	0	5
stok_0_26	/dev/rdisk/c140t12d16s3	0	5
stok_0_27	/dev/rdisk/c140t12d17s3	0	5
stok_0_28	/dev/rdisk/c159t12d12s3	0	5
stok_0_29	/dev/rdisk/c159t12d13s3	0	5
stok_0_30	/dev/rdisk/c159t12d14s3	0	5
stok_0_31	/dev/rdisk/c159t12d15s3	0	5
stok_0_32	/dev/rdisk/c159t12d16s3	0	5
stok_0_33	/dev/rdisk/c159t12d17s3	0	5

datafile name	device name	RAID Level	number of drives
cust_0_0	/dev/rdisk/c163t10d0s3	0	5
temp_0_0	/dev/rdisk/c163t10d0s4		
cust_0_1	/dev/rdisk/c163t10d1s3	0	5
temp_0_1	/dev/rdisk/c163t10d1s4		
cust_0_2	/dev/rdisk/c163t10d2s3	0	5
temp_0_2	/dev/rdisk/c163t10d2s4		
cust_0_3	/dev/rdisk/c163t10d3s3	0	5
temp_0_3	/dev/rdisk/c163t10d3s4		
cust_0_4	/dev/rdisk/c163t10d4s3	0	5
temp_0_4	/dev/rdisk/c163t10d4s4		
cust_0_5	/dev/rdisk/c163t10d5s3	0	5
temp_0_5	/dev/rdisk/c163t10d5s4		
cust_0_6	/dev/rdisk/c161t10d0s3	0	5
temp_0_6	/dev/rdisk/c161t10d0s4		
cust_0_7	/dev/rdisk/c161t10d1s3	0	5
temp_0_7	/dev/rdisk/c161t10d1s4		
cust_0_8	/dev/rdisk/c161t10d2s3	0	5
temp_0_8	/dev/rdisk/c161t10d2s4		
cust_0_9	/dev/rdisk/c161t10d3s3	0	5
temp_0_9	/dev/rdisk/c161t10d3s4		
cust_0_10	/dev/rdisk/c161t10d4s3	0	5
temp_0_10	/dev/rdisk/c161t10d4s4		
cust_0_11	/dev/rdisk/c161t10d5s3	0	5
temp_0_11	/dev/rdisk/c161t10d5s4		
cust_0_12	/dev/rdisk/c142t10d0s3	0	5
temp_0_12	/dev/rdisk/c142t10d0s4		
cust_0_13	/dev/rdisk/c142t10d1s3	0	5
temp_0_13	/dev/rdisk/c142t10d1s4		
cust_0_14	/dev/rdisk/c142t10d2s3	0	5
temp_0_14	/dev/rdisk/c142t10d2s4		
cust_0_15	/dev/rdisk/c142t10d3s3	0	5
temp_0_15	/dev/rdisk/c142t10d3s4		
cust_0_16	/dev/rdisk/c142t10d4s3	0	5
temp_0_16	/dev/rdisk/c142t10d4s4		

datafile name	device name	RAID Level	number of drives
stok_0_34	/dev/rdisk/c167t12d12s3	0	5
stok_0_35	/dev/rdisk/c167t12d13s3	0	5
stok_0_36	/dev/rdisk/c167t12d14s3	0	5
stok_0_37	/dev/rdisk/c167t12d15s3	0	5
stok_0_38	/dev/rdisk/c167t12d16s3	0	5
stok_0_39	/dev/rdisk/c167t12d17s3	0	5
stok_0_40	/dev/rdisk/c155t12d12s3	0	5
stok_0_41	/dev/rdisk/c155t12d13s3	0	5
stok_0_42	/dev/rdisk/c155t12d14s3	0	5
stok_0_43	/dev/rdisk/c155t12d15s3	0	5
stok_0_44	/dev/rdisk/c155t12d16s3	0	5
stok_0_45	/dev/rdisk/c155t12d17s3	0	5
stok_0_46	/dev/rdisk/c147t12d12s3	0	5
stok_0_47	/dev/rdisk/c147t12d13s3	0	5
stok_0_48	/dev/rdisk/c147t12d14s3	0	5
stok_0_49	/dev/rdisk/c147t12d15s3	0	5
stok_0_50	/dev/rdisk/c147t12d16s3	0	5
stok_0_51	/dev/rdisk/c147t12d17s3	0	5
stok_0_52	/dev/rdisk/c157t12d12s3	0	5
stok_0_53	/dev/rdisk/c157t12d13s3	0	5
stok_0_54	/dev/rdisk/c157t12d14s3	0	5
stok_0_55	/dev/rdisk/c157t12d15s3	0	5
stok_0_56	/dev/rdisk/c157t12d16s3	0	5
stok_0_57	/dev/rdisk/c157t12d17s3	0	5
stok_0_58	/dev/rdisk/c151t12d12s3	0	5
stok_0_59	/dev/rdisk/c151t12d14s3	0	5
stok_0_60	/dev/rdisk/c151t12d15s3	0	5
stok_0_61	/dev/rdisk/c151t12d16s3	0	5
stok_0_62	/dev/rdisk/c151t12d17s3	0	5
stok_0_63	/dev/rdisk/c160t12d12s3	0	5
stok_0_64	/dev/rdisk/c160t12d13s3	0	5
stok_0_65	/dev/rdisk/c160t12d14s3	0	5
stok_0_66	/dev/rdisk/c160t12d15s3	0	5
stok_0_67	/dev/rdisk/c160t12d16s3	0	5
stok_0_68	/dev/rdisk/c160t12d17s3	0	5
stok_0_69	/dev/rdisk/c148t12d12s3	0	5
stok_0_70	/dev/rdisk/c148t12d13s3	0	5
stok_0_71	/dev/rdisk/c148t12d14s3	0	5
stok_0_72	/dev/rdisk/c148t12d15s3	0	5
stok_0_73	/dev/rdisk/c148t12d16s3	0	5

datafile name	device name	RAID Level	number of drives
cust_0_17	/dev/rdisk/c142t10d5s3	0	5
temp_0_17	/dev/rdisk/c142t10d5s4		
cust_0_18	/dev/rdisk/c158t10d0s3	0	5
temp_0_18	/dev/rdisk/c158t10d0s4		
cust_0_19	/dev/rdisk/c158t10d1s3	0	5
temp_0_19	/dev/rdisk/c158t10d1s4		
cust_0_20	/dev/rdisk/c158t10d2s3	0	5
temp_0_20	/dev/rdisk/c158t10d2s4		
cust_0_21	/dev/rdisk/c158t10d4s3	0	5
temp_0_21	/dev/rdisk/c158t10d4s4		
cust_0_22	/dev/rdisk/c158t10d5s3	0	5
temp_0_22	/dev/rdisk/c158t10d5s4		
cust_0_23	/dev/rdisk/c168t10d0s3	0	5
temp_0_23	/dev/rdisk/c168t10d0s4		
cust_0_24	/dev/rdisk/c168t10d2s3	0	5
temp_0_24	/dev/rdisk/c168t10d2s4		
cust_0_25	/dev/rdisk/c168t10d3s3	0	5
temp_0_25	/dev/rdisk/c168t10d3s4		
cust_0_26	/dev/rdisk/c168t10d4s3	0	5
temp_0_26	/dev/rdisk/c168t10d4s4		
cust_0_27	/dev/rdisk/c168t10d5s3	0	5
temp_0_27	/dev/rdisk/c168t10d5s4		
cust_0_28	/dev/rdisk/c154t10d0s3	0	5
temp_0_28	/dev/rdisk/c154t10d0s4		
cust_0_29	/dev/rdisk/c154t10d1s3	0	5
temp_0_29	/dev/rdisk/c154t10d1s4		
cust_0_30	/dev/rdisk/c154t10d2s3	0	5
temp_0_30	/dev/rdisk/c154t10d2s4		
cust_0_31	/dev/rdisk/c154t10d3s3	0	5
temp_0_31	/dev/rdisk/c154t10d3s4		
cust_0_32	/dev/rdisk/c154t10d4s3	0	5
temp_0_32	/dev/rdisk/c154t10d4s4		
cust_0_33	/dev/rdisk/c154t10d5s3	0	5
temp_0_33	/dev/rdisk/c154t10d5s4		
cust_0_34	/dev/rdisk/c146t10d0s3	0	5
temp_0_34	/dev/rdisk/c146t10d0s4		
cust_0_35	/dev/rdisk/c146t10d1s3	0	5
temp_0_35	/dev/rdisk/c146t10d1s4		
cust_0_36	/dev/rdisk/c146t10d2s3	0	5
temp_0_36	/dev/rdisk/c146t10d2s4		

datafile name	device name	RAID Level	number of drives
stok_0_74	/dev/rdisk/c148t12d17s3	0	5
stok_0_75	/dev/rdisk/c145t12d12s3	0	5
stok_0_76	/dev/rdisk/c145t12d13s3	0	5
stok_0_77	/dev/rdisk/c145t12d14s3	0	5
stok_0_78	/dev/rdisk/c145t12d15s3	0	5
stok_0_79	/dev/rdisk/c145t12d16s3	0	5
stok_0_80	/dev/rdisk/c145t12d17s3	0	5
stok_0_81	/dev/rdisk/c150t12d12s3	0	5
stok_0_82	/dev/rdisk/c150t12d13s3	0	5
stok_0_83	/dev/rdisk/c150t12d14s3	0	5
stok_0_84	/dev/rdisk/c150t12d15s3	0	5
stok_0_85	/dev/rdisk/c150t12d16s3	0	5
stok_0_86	/dev/rdisk/c150t12d17s3	0	5
stok_0_87	/dev/rdisk/c198t13d18s3	0	5
stok_0_88	/dev/rdisk/c198t13d19s3	0	5
stok_0_89	/dev/rdisk/c198t13d21s3	0	5
stok_0_90	/dev/rdisk/c198t13d22s3	0	5
stok_0_91	/dev/rdisk/c198t13d23s3	0	5
stok_0_92	/dev/rdisk/c196t13d18s3	0	5
stok_0_93	/dev/rdisk/c196t13d19s3	0	5
stok_0_94	/dev/rdisk/c196t13d20s3	0	5
stok_0_95	/dev/rdisk/c196t13d21s3	0	5
stok_0_96	/dev/rdisk/c196t13d22s3	0	5
stok_0_97	/dev/rdisk/c196t13d23s3	0	5
stok_0_98	/dev/rdisk/c200t13d18s3	0	5
stok_0_99	/dev/rdisk/c200t13d19s3	0	5
stok_0_100	/dev/rdisk/c200t13d20s3	0	5
stok_0_101	/dev/rdisk/c200t13d21s3	0	5
stok_0_102	/dev/rdisk/c200t13d22s3	0	5
stok_0_103	/dev/rdisk/c200t13d23s3	0	5
stok_0_104	/dev/rdisk/c217t13d18s3	0	5
stok_0_105	/dev/rdisk/c217t13d19s3	0	5
stok_0_106	/dev/rdisk/c217t13d20s3	0	5
stok_0_107	/dev/rdisk/c217t13d21s3	0	5
stok_0_108	/dev/rdisk/c217t13d22s3	0	5
ordr_0_0	/dev/rdisk/c191t11d9s3	0	5
ordr_0_1	/dev/rdisk/c191t11d10s3	0	5
ordr_0_2	/dev/rdisk/c191t11d11s3	0	5

datafile name	device name	RAID Level	number of drives
cust_0_37	/dev/rdisk/c146t10d3s3	0	5
temp_0_37	/dev/rdisk/c146t10d3s4		
cust_0_38	/dev/rdisk/c146t10d4s3	0	5
temp_0_38	/dev/rdisk/c146t10d4s4		
cust_0_39	/dev/rdisk/c146t10d5s3	0	5
temp_0_39	/dev/rdisk/c146t10d5s4		
cust_0_40	/dev/rdisk/c152t10d0s3	0	5
temp_0_40	/dev/rdisk/c152t10d0s4		
cust_0_41	/dev/rdisk/c152t10d1s3	0	5
temp_0_41	/dev/rdisk/c152t10d1s4		
cust_0_42	/dev/rdisk/c152t10d2s3	0	5
temp_0_42	/dev/rdisk/c152t10d2s4		
cust_0_43	/dev/rdisk/c152t10d3s3	0	5
temp_0_43	/dev/rdisk/c152t10d3s4		
cust_0_44	/dev/rdisk/c152t10d4s3	0	5
temp_0_44	/dev/rdisk/c152t10d4s4		
cust_0_45	/dev/rdisk/c152t10d5s3	0	5
temp_0_45	/dev/rdisk/c152t10d5s4		
cust_0_46	/dev/rdisk/c153t10d0s3	0	5
temp_0_46	/dev/rdisk/c153t10d0s4		
cust_0_47	/dev/rdisk/c153t10d1s3	0	5
temp_0_47	/dev/rdisk/c153t10d1s4		
cust_0_48	/dev/rdisk/c153t10d2s3	0	5
temp_0_48	/dev/rdisk/c153t10d2s4		
cust_0_49	/dev/rdisk/c153t10d3s3	0	5
temp_0_49	/dev/rdisk/c153t10d3s4		
cust_0_50	/dev/rdisk/c153t10d4s3	0	5
temp_0_50	/dev/rdisk/c153t10d4s4		
cust_0_51	/dev/rdisk/c153t10d5s3	0	5
temp_0_51	/dev/rdisk/c153t10d5s4		
cust_0_52	/dev/rdisk/c156t10d0s3	0	5
temp_0_52	/dev/rdisk/c156t10d0s4		
cust_0_53	/dev/rdisk/c156t10d1s3	0	5
temp_0_53	/dev/rdisk/c156t10d1s4		
cust_0_54	/dev/rdisk/c156t10d2s3	0	5
temp_0_54	/dev/rdisk/c156t10d2s4		
cust_0_55	/dev/rdisk/c156t10d3s3	0	5
temp_0_55	/dev/rdisk/c156t10d3s4		

datafile name	device name	RAID Level	number of drives
ordr_0_3	/dev/rdisk/c190t11d6s3	0	5
ordr_0_4	/dev/rdisk/c190t11d7s3	0	5
ordr_0_5	/dev/rdisk/c190t11d8s3	0	5
ordr_0_6	/dev/rdisk/c190t11d9s3	0	5
ordr_0_7	/dev/rdisk/c190t11d10s3	0	5
ordr_0_8	/dev/rdisk/c190t11d11s3	0	5
ordr_0_9	/dev/rdisk/c216t11d6s3	0	5
ordr_0_10	/dev/rdisk/c216t11d7s3	0	5
ordr_0_11	/dev/rdisk/c216t11d9s3	0	5
ordr_0_12	/dev/rdisk/c216t11d10s3	0	5
ordr_0_13	/dev/rdisk/c216t11d11s3	0	5
ordr_0_14	/dev/rdisk/c204t11d6s3	0	5
ordr_0_15	/dev/rdisk/c204t11d7s3	0	5
ordr_0_16	/dev/rdisk/c204t11d8s3	0	5
ordr_0_17	/dev/rdisk/c204t11d9s3	0	5
ordr_0_18	/dev/rdisk/c204t11d10s3	0	5
ordr_0_19	/dev/rdisk/c204t11d11s3	0	5
ordr_0_20	/dev/rdisk/c206t11d6s3	0	5
ordr_0_21	/dev/rdisk/c206t11d7s3	0	5
ordr_0_22	/dev/rdisk/c206t11d8s3	0	5
ordr_0_23	/dev/rdisk/c206t11d9s3	0	5
ordr_0_24	/dev/rdisk/c206t11d10s3	0	5
ordr_0_25	/dev/rdisk/c206t11d11s3	0	5
ordr_0_26	/dev/rdisk/c208t11d6s3	0	5
ordr_0_27	/dev/rdisk/c208t11d7s3	0	5
ordr_0_28	/dev/rdisk/c208t11d8s3	0	5
ordr_0_29	/dev/rdisk/c208t11d9s3	0	5
nord_0_0	/dev/rdisk/c191t11d6s3	0	5
nord_0_1	/dev/rdisk/c191t11d7s3	0	5
nord_0_2	/dev/rdisk/c191t11d8s3	0	5
icust2_0_5	/dev/rdisk/c192t13d18s3	0	5
istok_0_5	/dev/rdisk/c192t13d18s4	0	5
ware_0_0	/dev/rdisk/c208t11d10s3	0	5
iware_0_0	/dev/rdisk/c208t11d10s4	0	5
aux.df	/dev/rdisk/c208t11d10s5	0	5
SYS	/dev/rdisk/c208t11d10s6	0	5
dist_0_0	/dev/rdisk/c208t11d11s3	0	5
idist_0_0	/dev/rdisk/c208t11d11s4	0	5
sp_0	/dev/rdisk/c208t11d11s5	0	5
item_0_0	/dev/rdisk/c210t11d7s3	0	5
iitem_0_0	/dev/rdisk/c210t11d7s4	0	5
roll_0_0	/dev/rdisk/c212t13d22s5	0	5

datafile name	device name	RAID Level	number of drives
cust_0_56	/dev/rdisk/c156t10d4s3	0	5
temp_0_56	/dev/rdisk/c156t10d4s4	0	5
cust_0_57	/dev/rdisk/c156t10d5s3	0	5
temp_0_57	/dev/rdisk/c156t10d5s4	0	5
cust_0_58	/dev/rdisk/c144t10d0s3	0	5
temp_0_58	/dev/rdisk/c144t10d0s4	0	5
cust_0_59	/dev/rdisk/c144t10d1s3	0	5
temp_0_59	/dev/rdisk/c144t10d1s4	0	5
cust_0_60	/dev/rdisk/c144t10d2s3	0	5
temp_0_60	/dev/rdisk/c144t10d2s4	0	5
cust_0_61	/dev/rdisk/c144t10d3s3	0	5
temp_0_61	/dev/rdisk/c144t10d3s4	0	5
cust_0_62	/dev/rdisk/c144t10d4s3	0	5
temp_0_62	/dev/rdisk/c144t10d4s4	0	5
cust_0_63	/dev/rdisk/c144t10d5s3	0	5
temp_0_63	/dev/rdisk/c144t10d5s4	0	5
cust_0_64	/dev/rdisk/c141t10d0s3	0	5
temp_0_64	/dev/rdisk/c141t10d0s4	0	5
cust_0_65	/dev/rdisk/c141t10d1s3	0	5
temp_0_65	/dev/rdisk/c141t10d1s4	0	5
cust_0_66	/dev/rdisk/c141t10d2s3	0	5
temp_0_66	/dev/rdisk/c141t10d2s4	0	5
cust_0_67	/dev/rdisk/c141t10d3s3	0	5
temp_0_67	/dev/rdisk/c141t10d3s4	0	5
cust_0_68	/dev/rdisk/c141t10d4s3	0	5
temp_0_68	/dev/rdisk/c141t10d4s4	0	5
cust_0_69	/dev/rdisk/c141t10d5s3	0	5
temp_0_69	/dev/rdisk/c141t10d5s4	0	5
cust_0_70	/dev/rdisk/c149t10d0s3	0	5
temp_0_70	/dev/rdisk/c149t10d0s4	0	5
cust_0_71	/dev/rdisk/c149t10d1s3	0	5
temp_0_71	/dev/rdisk/c149t10d1s4	0	5
cust_0_72	/dev/rdisk/c149t10d2s3	0	5
temp_0_72	/dev/rdisk/c149t10d2s4	0	5
cust_0_73	/dev/rdisk/c149t10d3s3	0	5
temp_0_73	/dev/rdisk/c149t10d3s4	0	5
cust_0_74	/dev/rdisk/c149t10d4s3	0	5
temp_0_74	/dev/rdisk/c149t10d4s4	0	5
cust_0_75	/dev/rdisk/c149t10d5s3	0	5
temp_0_75	/dev/rdisk/c149t10d5s4	0	5
cust_0_76	/dev/rdisk/c199t11d6s3	0	5
temp_0_76	/dev/rdisk/c199t11d6s4	0	5

datafile name	device name	RAID Level	number of drives
hist_0_0	/dev/rdisk/c213t11d10s3	0	5
hist_0_1	/dev/rdisk/c213t11d11s3	0	5
hist_0_2	/dev/rdisk/c214t11d6s3	0	5
hist_0_3	/dev/rdisk/c214t11d7s3	0	5
hist_0_4	/dev/rdisk/c214t11d8s3	0	5
hist_0_5	/dev/rdisk/c214t11d9s3	0	5
hist_0_6	/dev/rdisk/c214t11d10s3	0	5
hist_0_7	/dev/rdisk/c214t11d11s3	0	5
icust1_0_0	/dev/rdisk/c217t13d23s3	0	5
icust1_0_1	/dev/rdisk/c215t13d18s3	0	5
icust2_0_0	/dev/rdisk/c215t13d19s3	0	5
istok_0_0	/dev/rdisk/c215t13d19s4		
icust2_0_1	/dev/rdisk/c215t13d20s3	0	5
istok_0_1	/dev/rdisk/c215t13d20s4		
icust2_0_2	/dev/rdisk/c215t13d21s3	0	5
istok_0_2	/dev/rdisk/c215t13d21s4		
icust2_0_3	/dev/rdisk/c215t13d22s3	0	5
istok_0_3	/dev/rdisk/c215t13d22s4		
icust2_0_4	/dev/rdisk/c215t13d23s3	0	5
istok_0_4	/dev/rdisk/c215t13d23s4		
LOG_3	/dev/vx/rdisk/E/log3*	5	Stripe on 48LUN(each 5Drives)
LOG_4	/dev/vx/rdisk/E/log4*		
LOG_5	/dev/vx/rdisk/E/log5*		
LOG_6	/dev/vx/rdisk/E/log6*		
LOG_7	/dev/vx/rdisk/E/log7*		
LOG_8	/dev/vx/rdisk/E/log8*		
LOG_9	/dev/vx/rdisk/E/log9*		
iordr2_0_0	/dev/vx/rdisk/iO/iordr2-0*	0	Stripe on 12LUN (each 5Drives)
iordr2_0_1	/dev/vx/rdisk/iO/iordr2-1*	0	
iordr2_0_2	/dev/vx/rdisk/iO/iordr2-2*	0	

datafile name	device name	RAID Level	number of drives
cust_0_77	/dev/rdisk/c199t11d7s3	0	5
temp_0_77	/dev/rdisk/c199t11d7s4		
cust_0_78	/dev/rdisk/c199t11d8s3	0	5
temp_0_78	/dev/rdisk/c199t11d8s4		
cust_0_79	/dev/rdisk/c199t11d9s3	0	5
temp_0_79	/dev/rdisk/c199t11d9s4		
cust_0_80	/dev/rdisk/c199t11d10s3	0	5
temp_0_80	/dev/rdisk/c199t11d10s4		
cust_0_81	/dev/rdisk/c199t11d11s3	0	5
temp_0_81	/dev/rdisk/c199t11d11s4		
cust_0_82	/dev/rdisk/c197t11d6s3	0	5
temp_0_82	/dev/rdisk/c197t11d6s4		
cust_0_83	/dev/rdisk/c197t11d7s3	0	5
temp_0_83	/dev/rdisk/c197t11d7s4		
cust_0_84	/dev/rdisk/c197t11d8s3	0	5
temp_0_84	/dev/rdisk/c197t11d8s4		
cust_0_85	/dev/rdisk/c197t11d9s3	0	5
temp_0_85	/dev/rdisk/c197t11d9s4		
cust_0_86	/dev/rdisk/c197t11d10s3	0	5
temp_0_86	/dev/rdisk/c197t11d10s4		
cust_0_87	/dev/rdisk/c201t11d7s3	0	5
temp_0_87	/dev/rdisk/c201t11d7s4		
cust_0_88	/dev/rdisk/c201t11d8s3	0	5
temp_0_88	/dev/rdisk/c201t11d8s4		
cust_0_89	/dev/rdisk/c201t11d9s3	0	5
temp_0_89	/dev/rdisk/c201t11d9s4		
cust_0_90	/dev/rdisk/c201t11d10s3	0	5
temp_0_90	/dev/rdisk/c201t11d10s4		
cust_0_91	/dev/rdisk/c201t11d11s3	0	5
temp_0_91	/dev/rdisk/c201t11d11s4		
cust_0_92	/dev/rdisk/c213t11d6s3	0	5
temp_0_92	/dev/rdisk/c213t11d6s4		
cust_0_93	/dev/rdisk/c213t11d7s3	0	5
temp_0_93	/dev/rdisk/c213t11d7s4		
cust_0_94	/dev/rdisk/c213t11d8s3	0	5
temp_0_94	/dev/rdisk/c213t11d8s4		
cust_0_95	/dev/rdisk/c213t11d9s3	0	5
temp_0_95	/dev/rdisk/c213t11d9s4		

Table 4.2 Database Layout

4.3 Type of Database

A statement must be provided that describes:

- 1. The data model implemented by DBMS used (e.g. relational, network, hierarchical).*
- 2. The database interface (e.g. embedded, call level) and access language (e.g. SQL, DL/I, COBOL read/write used to implement the TPC-C transaction. If more than one interface/access language is used to implement TPC-C, each interface/access language must be described and a list of which interface/access language is used with which transaction type must be disclosed.*

Oracle is a relational DBMS.

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

4.4 Database Mapping

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

The database was neither partitioned nor replicated.

4.5 60 Day Space

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

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	595,702.31
Price per tpmC	\$12.43

5.2 Response Times

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

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

Type	Average	90th %	Maximum
New-Order	0.598	0.870	38.889
Payment	0.564	0.850	34.921
Order-Status	0.561	0.850	27.331
Interactive Delivery	0.131	0.120	22.309
Deferred Delivery	0.354	0.550	25.093
Stock-Level	0.544	0.830	29.641
Menu	0.142	0.120	22.322

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	18.01	18.02	18.18
Payment	3.02	3.03	3.32
Order-Status	2.00	2.01	2.13
Interactive Delivery	2.00	2.01	2.15
Stock-Level	2.00	2.01	2.12

Table 5.3 Think Times

Type	Minimum	Average	Maximum
New-Order	0.01	12.02	120.11
Payment	0.01	12.01	120.11
Order-Status	0.02	10.02	100.20
Interactive Delivery	0.01	5.02	50.10
Stock-Level	0.01	5.02	50.09

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

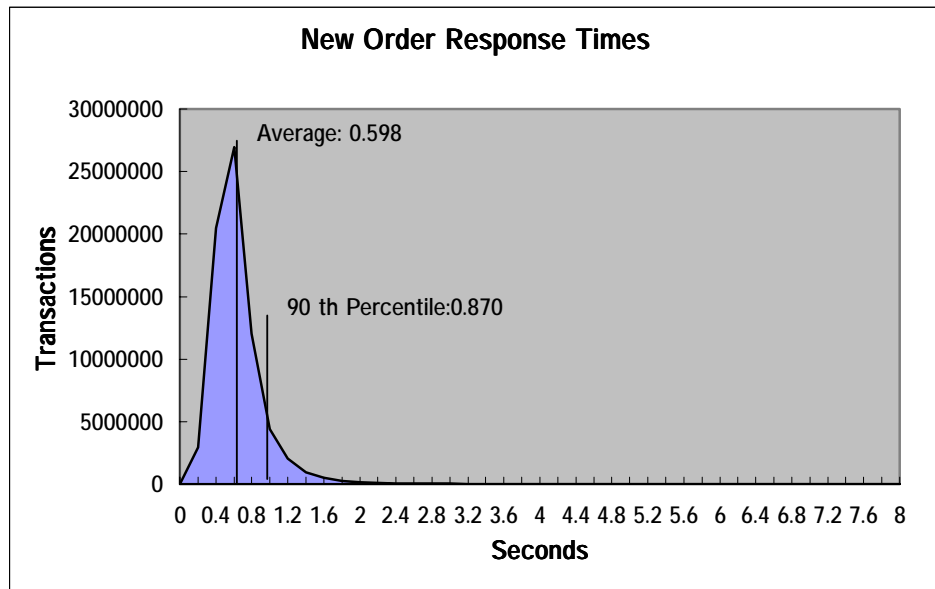


Figure 5.2: Payment Response Time Distribution

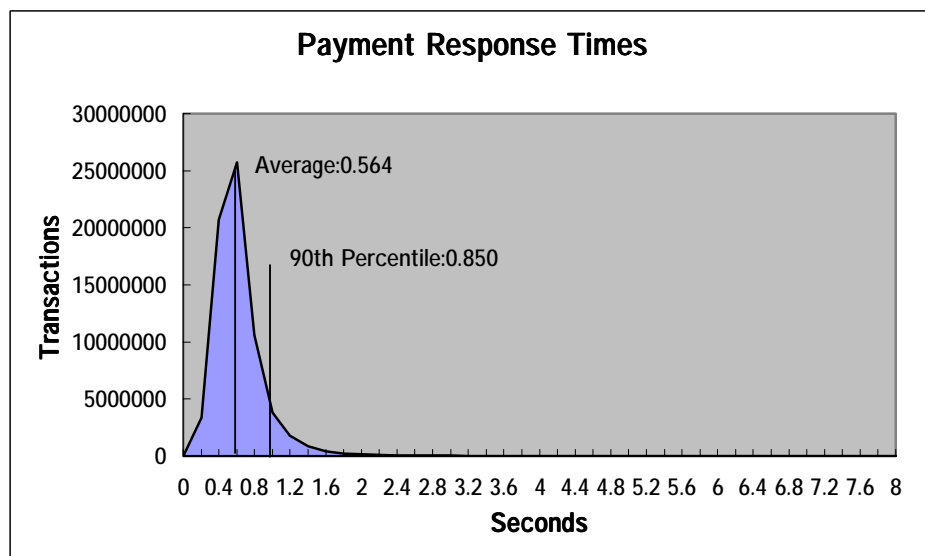


Figure 5.3: Order Status Response Time Distribution

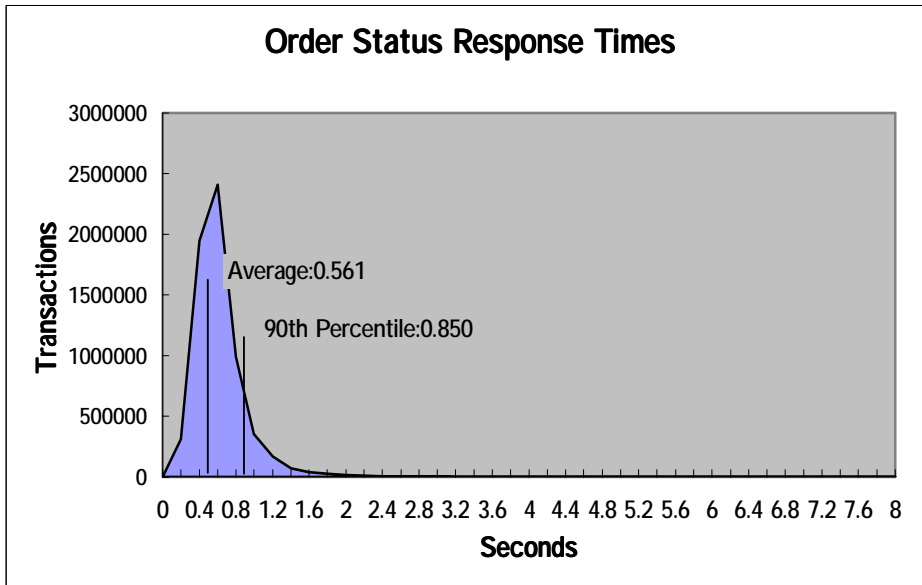


Figure 5.4: Delivery Response Time Distribution

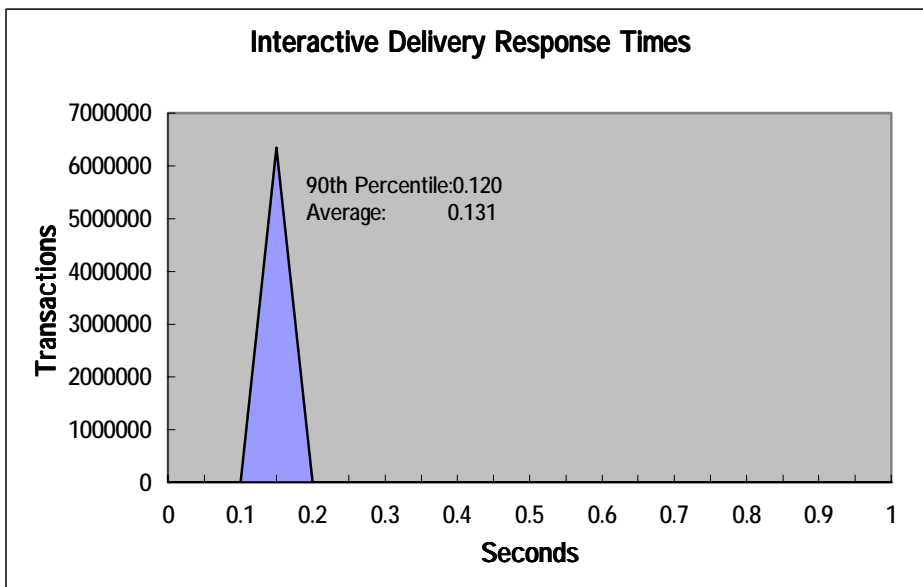


Figure 5.5: Stock Level Response Time Distribution

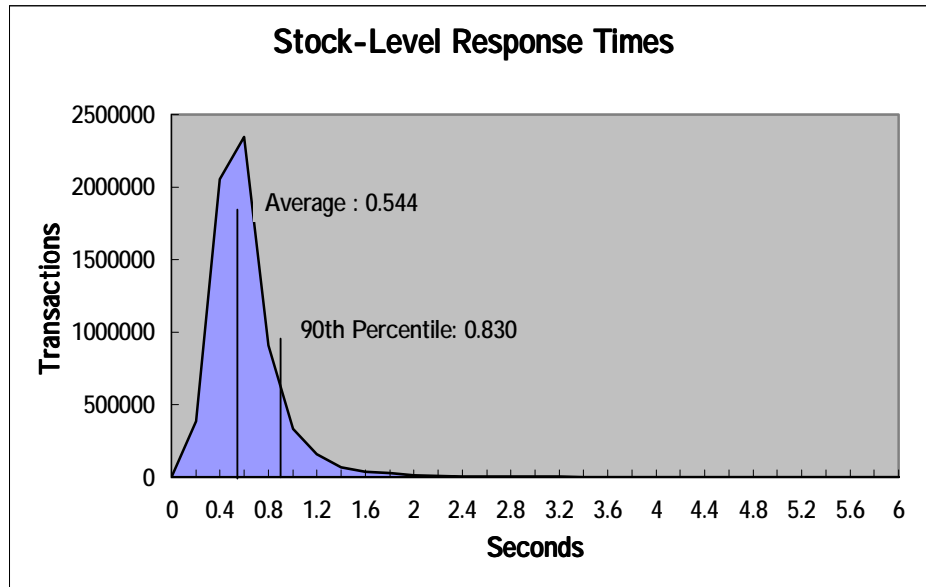


Figure 5.6: New Order Think Time Frequency Distribution

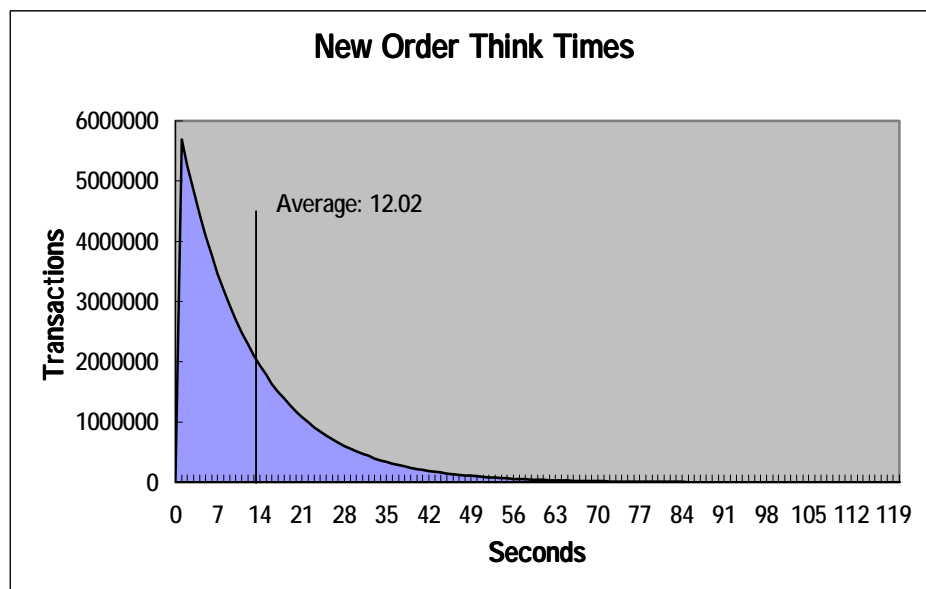


Figure 5.7: Response time versus Throughput

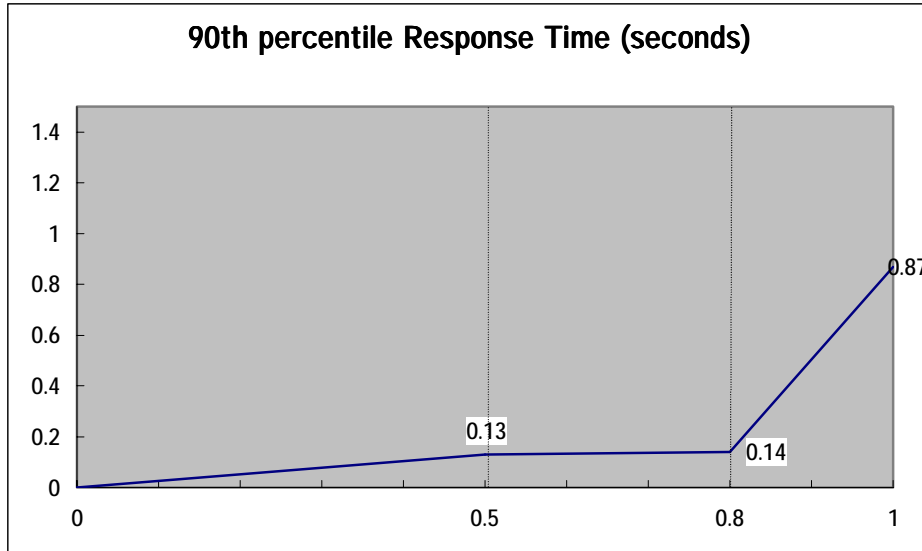
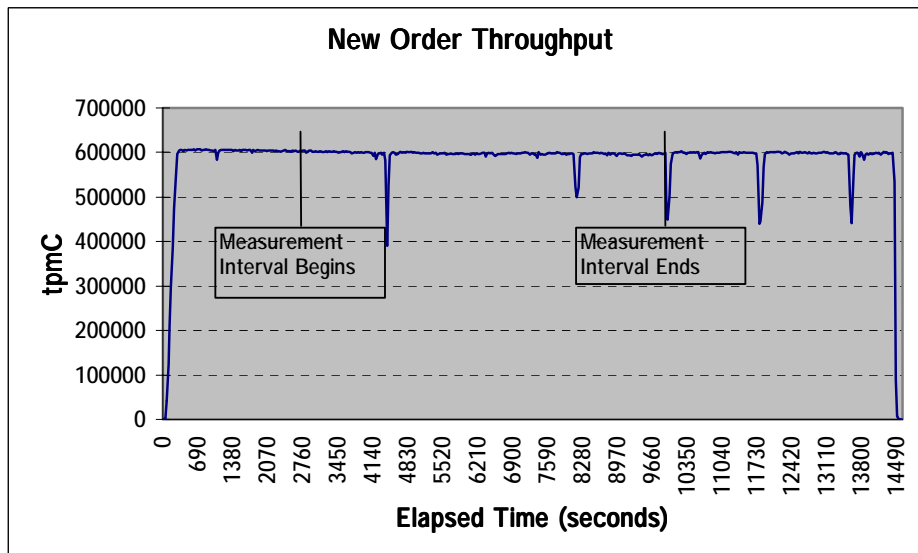


Figure 5.8: New Order Sustained Throughput



5.5 Steady State Determination

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval 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 RAID5 array. In Oracle Database 10g, when one Logfile becomes full, a checkpoint occurs. Oracle automatically logs all checkpoints to an alert file on the server. We created the Log file so that it might become full in about 30 minutes. Oracle Database 10g performed 4 times of Log file Switches during MI. At each checkpoint, Oracle wrote to disk all buffer pages that had been updated but not yet physically written to disk.

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

Serializable Transactions:

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

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

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

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

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

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

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

When a serializable transaction fails with this error, the application may either commit the work executed to that point, execute additional statements, or rollback the entire transaction. Repeated attempts to execute the same statement will always fail with the error "can't serialize access" unless the other transaction has rolled back and released 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 used the UNIX function `lrand48()` to control the transaction mix, and 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	44.91%
	Payment	43.03%
	Order status	4.02%
	Delivery	4.02%
	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	59.99%
Order Status	Accessed by last name	60.00%
Delivery	Skipped transactions	None

5.11 Checkpoint Count and Location

The number of checkpoints in the Measurement Interval, the time in seconds from the start of the Measurement Interval to the first checkpoint, and the Checkpoint Interval must be disclosed.

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

	start	end	duration
measurement	10:32:35	12:32:35	120 minutes
	start	End	duration
checkpoint 0	10:05:00	10:31:34	26:34
checkpoint 1	10:34:19	11:01:04	26:45
checkpoint 2	11:03:49	11:30:36	26:47
checkpoint 3	11:33:22	12:00:15	26:53
checkpoint 4	12:03:01	12:30:07	27:06

Clause 6 Related Items

6.1 RTE Descriptions

If the RTE is commercially available, then its inputs must be specified. Otherwise, a description must be supplied of what inputs (e.g., scripts) to the RTE had been used.

The RTE used was developed at Fujitsu Limited and is proprietary. It consists of an RTE management process as shown in Appendix C, which forks off the individual RTE processes and controls the run. After the run completes, a separate report generator program collects all the log files and generates the final statistics of a run.

Inputs to the RTE include the names of the RTE machine to run, client machines to attach to, the database scale, the ramp-up, measurement and ramp-down times. These come from the configuration script file for the RTE management process.

6.2 Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to 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.

6.3 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.

6.4 Networks

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

The bandwidth of the networks used in the tested/priced configuration must be disclosed.

Seven 100Mbps ethernet LAN connection was used between each client and the server. Twenty eight 100Mbps ethernet LAN connections were used between the client machines and the switches connected to the emulated users.

6.5 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 System Pricing

A detailed list of hardware and software used in the priced system must be reported. Each separately orderable item must have vendor part number, description, and release/revision level, and either general availability status or committed delivery 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 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.

Oracle Database 10g Enterprise Edition will be available no later than April 30,2004.

The other components are available now.

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 (tpmC):	595,702.31
Price per tpmC	\$12.43
Availability date	April 30,2004

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.*

None.

Clause 9 Related Items

9.1 Auditor's Report

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

This implementation of the TPC Benchmark C was audited by Francois Raab of InfoSizing.

InfoSizing, Inc.
1373 North Franklin St.
Colorado Springs, CO 80903-2527
(voice) 719/473-7555
(fax) 719/473-7554
<http://www.sizing.com>

9.2 Availability of the Full Disclosure Report

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

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

Transaction Processing Performance Council
Presidio of San Francisco
Building 572B (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

```

.....
TPCCPublicFunc\TPCCPublicFunc.mak
.....
# Microsoft Developer Studio Generated NMAKE File,
Based on TPCCPublicFunc.dsp
!IF "$(CFG)" == ""
CFG=TPCCPublicFunc - Win32 Debug
!MESSAGE TPCCPublicFunc - Win32 Debug
!ENDIF
!IF "$(CFG)" != "TPCCPublicFunc - Win32 Release" &&
"$(CFG)" != "TPCCPublicFunc - Win32 Debug"
!MESSAGE "$(CFG)"
!MESSAGE NMAKE
!MESSAGE :
!MESSAGE
!MESSAGE NMAKE /f "TPCCPublicFunc.mak"
CFG="TPCCPublicFunc - Win32 Debug"
!MESSAGE
!MESSAGE :
!MESSAGE
!MESSAGE "TPCCPublicFunc - Win32 Release"
("Win32 (x86) Static Library" )
!MESSAGE "TPCCPublicFunc - Win32 Debug"
("Win32 (x86) Static Library" )
!MESSAGE
!ERROR
!ENDIF
!IF "$(OS)" == "Windows_NT"
NULL=
!ELSE
NULL=nul
!ENDIF
!IF "$(CFG)" == "TPCCPublicFunc - Win32 Release"
OUTDIR=.Release
INTDIR=.Release
# Begin Custom Macros
OutDir=.Release
# End Custom Macros
ALL : "$(OUTDIR)\TPCCPublicFunc.lib"
CLEAN :
    -@erase "$(INTDIR)\uselog.obj"
    -@erase "$(INTDIR)\vc60.idb"
    -@erase "$(OUTDIR)\TPCCPublicFunc.lib"
"$(OUTDIR)" :
    if not exist "$(OUTDIR)\$(NULL)" mkdir "$(OUTDIR)"
CPP=cl.exe
CPP_PROJ=/nologo /MT /W3 /GX /O2 /D "WIN32" /D
"NDEBUG" /D "_MBCS" /D "_LIB"
/Fp "$(INTDIR)\TPCCPublicFunc.pch" /YX
/Fo "$(INTDIR)\\" /Fd "$(INTDIR)\\" /FD /c
.c $(INTDIR).obj:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cpp $(INTDIR).obj:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cxx $(INTDIR).obj:
    $(CPP) @<<
    $(CPP_PROJ) $<

```

```

<<
.c $(INTDIR).sbr:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cpp $(INTDIR).sbr:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cxx $(INTDIR).sbr:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
RSC=rc.exe
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/o "$(OUTDIR)\TPCCPublicFunc.bsc"
BSC32_SBRS= \

LIB32=link.exe -lib
LIB32_FLAGS=/nologo
/out: "$(OUTDIR)\TPCCPublicFunc.lib"
LIB32_OBJS= \
    "$(INTDIR)\uselog.obj"
"$(OUTDIR)\TPCCPublicFunc.lib" : "$(OUTDIR)"
$(DEF_FILE) $(LIB32_OBJS)
$(LIB32) @<<
$(LIB32_FLAGS) $(DEF_FLAGS) $(LIB32_OBJS)
<<
!ELSEIF "$(CFG)" == "TPCCPublicFunc - Win32
Debug"
OUTDIR=.Debug
INTDIR=.Debug
# Begin Custom Macros
OutDir=.Debug
# End Custom Macros
ALL : "$(OUTDIR)\TPCCPublicFunc.lib"
CLEAN :
    -@erase "$(INTDIR)\uselog.obj"
    -@erase "$(INTDIR)\vc60.idb"
    -@erase "$(INTDIR)\vc60.pdb"
    -@erase "$(OUTDIR)\TPCCPublicFunc.lib"
"$(OUTDIR)" :
    if not exist "$(OUTDIR)\$(NULL)" mkdir "$(OUTDIR)"
CPP=cl.exe
CPP_PROJ=/nologo /MTd /W4 /Gm /GX /ZI /Od /D
"WIN32" /D "_DEBUG" /D "_MBCS" /D "_LIB"
/Fp "$(INTDIR)\TPCCPublicFunc.pch" /YX
/Fo "$(INTDIR)\\" /Fd "$(INTDIR)\\" /FD /GZ /c
.c $(INTDIR).obj:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cpp $(INTDIR).obj:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cxx $(INTDIR).obj:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.c $(INTDIR).sbr:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cpp $(INTDIR).sbr:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cxx $(INTDIR).sbr:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<

```

```

.cxx $(INTDIR).sbr:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
RSC=rc.exe
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/o "$(OUTDIR)\TPCCPublicFunc.bsc"
BSC32_SBRS= \

LIB32=link.exe -lib
LIB32_FLAGS=/nologo
/out: "$(OUTDIR)\TPCCPublicFunc.lib"
LIB32_OBJS= \
    "$(INTDIR)\uselog.obj"
"$(OUTDIR)\TPCCPublicFunc.lib" : "$(OUTDIR)"
$(DEF_FILE) $(LIB32_OBJS)
$(LIB32) @<<
$(LIB32_FLAGS) $(DEF_FLAGS) $(LIB32_OBJS)
<<
!ENDIF
!IF "$(NO_EXTERNAL_DEPS)" != "1"
!IF EXISTS("TPCCPublicFunc.dep")
!INCLUDE "TPCCPublicFunc.dep"
!ELSE
!MESSAGE Warning: cannot find
"TPCCPublicFunc.dep"
!ENDIF
!ENDIF
!IF "$(CFG)" == "TPCCPublicFunc - Win32 Release" ||
"$(CFG)" == "TPCCPublicFunc - Win32 Debug"
SOURCE=.uselog.cpp
"$(INTDIR)\uselog.obj" : $(SOURCE) "$(INTDIR)"
!ENDIF
.....
TPCCPublicFunc\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 2002 *
*****/
#ifdef LOG_MODULE
#include "logmdl_level.h"
#else
#include "log_level.h"
#endif
#define LOG_ERR __FILE__, __LINE__, "ERR"
#define LOG_INF __FILE__, __LINE__, "INF"
#define LOG_WRN __FILE__, __LINE__, "WRN"
void userlog (char *file_name, int line_no, char
type_name, char *fmt, ...);
/* Log file path */
extern char GLB_LogFilePath[];
#define LOG_MUTEX_NAME "TPC-C
LOG Mutex"
#define LOG_FILE_NAME
"log\userlog.log"
.....
TPCCPublicFunc\logmdl_level.h
.....

```

```

/*****
 *
 * TPC-C Client Application Program Source
 *
 * CREATE by TSL 2003.02.07
 *
 * All Right Reserved, Copyright Co. FUJITSU
 LIMITED 2003
 *****/
#define PUT_INF_LOG
// Information log
#define PUT_FUNC_ENTRY_LOG
// Function entry point log
#define PUT_FUNC_EXIT_LOG
// Function exit log
/* Function entry point log macro */
#ifdef PUT_FUNC_ENTRY_LOG
#define MAC_PutFuncEntryLog(func)
userlog(LOG_INF, ">>>>> "func" start >>>>>");
#else
#define MAC_PutFuncEntryLog(func) ;
#endif
/* Function exit point log */
#ifdef PUT_FUNC_EXIT_LOG
#define MAC_PutFuncExitLog(func)
userlog(LOG_INF, "<<<<< "func" end <<<<<");
#else
#define MAC_PutFuncExitLog(func) ;
#endif
.....
TPCCPublicFunc/uselog.cpp
/*****
 *
 * TPC-C Client Application Program Source
 *
 * Entry Functions
 * (1) ClientMonitor
 * (2) ClientStartup
 * (3) ClientShutdown
 *
 * CREATE by TSL 2002.11.29
 *
 * All Right Reserved, Copyright Co. FUJITSU
 LIMITED 2002
 *****/
#include <windows.h>
#include <stdio.h>
#include "ObjectCntlInfo.h"
#include "TpcCntlInfo.h"
#include "Delivery.h"
#include "trans.h"
#include "ExecuteThread.h"
#include "GlobalArea.h"
#include "SvrApiBase.h"
#include "log.h"
#include "TpApiPrototype.h"
#include "SampleInfo.h"
#include "TpApiDBDependPrototype.h"
/*****
 * Client monitoring function.
 * Return Value
 * NONE
 * 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:
ClientStartup(html_buf);
break;

```

```

tld = GetCurrentThreadId();
// Get file name from a path.
fname = strchr(file_name, '\\');
if (fname == NULL) {
fname = file_name;
} else {
fname = fname + 1;
}
va_start(va, fmp);
fp = fopen(GLB_LogFilePath, "a");
fprintf(fp, "%02d:%02d:%02d [%4d] %-
32s(%4d) :%s: ",
nowtime->tm_hour, nowtime->tm_min,
nowtime->tm_sec, tld, fname, line_no, type_name);
vprintf(fp, fmp, va);
if (*(fmp + strlen(fmp) - 1) != '\n')
fprintf(fp, "\n");
va_end(va);
fclose(fp);
ReleaseMutex(hMutex);
CloseHandle(hMutex);
return;
}
.....
TpApiBase/ClientMonitor.cpp
.....
/*****
 *
 * TPC-C Client Application Program Source
 *
 * Entry Functions
 * (1) ClientMonitor
 * (2) ClientStartup
 * (3) ClientShutdown
 *
 * CREATE by TSL 2003.06.06
 *
 * All Right Reserved, Copyright Co. FUJITSU
 LIMITED 2003
 *****/
#include <windows.h>
#include <stdio.h>
#include "ObjectCntlInfo.h"
#include "TpcCntlInfo.h"
#include "Delivery.h"
#include "trans.h"
#include "ExecuteThread.h"
#include "GlobalArea.h"
#include "SvrApiBase.h"
#include "log.h"
#include "TpApiPrototype.h"
#include "SampleInfo.h"
#include "TpApiDBDependPrototype.h"
/*****
 * Client monitoring function.
 * Return Value
 * NONE
 * 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:
ClientStartup(html_buf);
break;

```

```

/* Client shutdown */
case -2:
ClientShutdown(html_buf);
break;
/* Added T. Kato 03.08.22 Client monitoring function */
/* Client monitor */
case -3:
ClientInfSample(html_buf);
break;
/* Added end */
default:
/* Error return */
return -1;
break;
}
return 0;
}
/*****
 * Client startup.
 * DLL loads at first time by executing this function.
 *
 * Then executes the process attach entry.
 * This function only returns HTML document.
 *
 * Return Value
 * NONE
 * Return Information
 * HTML document
 *****/
void ClientStartup(char* html_buf) {
#define CLIENT_STARTUP "\
<HTML><HEAD><TITLE>Client
Startup</TITLE></HEAD><BODY>\r\n\
<P>The %s startup may be success. Will be watch the
client log.<BR></P></BODY></HTML>\r\n"
char host_name[32];
host_name[0] = '\0';
gethostname(host_name, sizeof(host_name));
sprintf(html_buf, CLIENT_STARTUP,
host_name);
}
/*****
 * Client shutdown.
 * Return Value
 * NONE
 * Return Information
 * HTML document
 *****/
void ClientShutdown(char* html_buf) {
#define CLIENT_SHUTDOWN "\
<HTML><HEAD><TITLE>Client
shutdown</TITLE></HEAD><BODY>\r\n\
<P>The %s shutdown may be success. Will be watch
the client log.<BR></P></BODY></HTML>\r\n"
char host_name[32];
/* Terminate Delivery */
TermDelivery();
/* Terminate SvrAPL execution environment area
*/
TermSvrApiExecEnv();
host_name[0] = '\0';
gethostname(host_name, sizeof(host_name));
sprintf(html_buf, CLIENT_SHUTDOWN,
host_name);
}
/*****
 * Client control information Sampling
 * Return Value

```

```

* NONE *
* Return Information *
* HTML document *
*****/
void ClientInfSample(char* html_buf) {
    int empty_delivery_queues;
    DELIVERY_QUE* empty_que;
#define CLIENT_SAMPLE "\
<HTML><HEAD><TITLE>Client Sampling
information</TITLE></HEAD><BODY>\
<PRE>\
Information of %s \r\n\
Delivery Queue informations \r\n\
    Current Empty : %d \r\n\
    Current Overflow : %d \r\n\
    MAX Overflow : %d \r\n\
    Current Wwaiting Que. (100ms) : %d \r\n\
    MAX Waiting que. (100ms) : %d \r\n\
\r\n\
Response Information \r\n\
    MIN MAX AVR TRX\r\n\
    Get Object Time %-10d%-10d%-10d%-10d
\r\n\
    New Order Response %-10d%-10d%-10d%-10d
\r\n\
    Payment Response %-10d%-10d%-10d%-10d
\r\n\
    Order Status Response %-10d%-10d%-10d%-10d
\r\n\
    Derivery Response %-10d%-10d%-10d%-10d
\r\n\
    Stock Level Response %-10d%-10d%-10d%-10d
\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
information</TITLE></HEAD><BODY>\
<PRE>\
Information of %s \r\n\
Delivery Queue informations \r\n\
    Current Empty : 33 \r\n\
    Current Overflow : 10 \r\n\
    MAX Overflow : 20 \r\n\
    Current Wwaiting Que. (100ms) : 30 \r\n\
    MAX Waiting que. (100ms) : 40 \r\n\
\r\n\
Response Information \r\n\
    MIN MAX AVR TRX \r\n\
    Get Object Time 10 100 50
1000\r\n\
    New Order Response 20 200 60
2000\r\n\
    Payment Response 30 300 70
3000\r\n\
    Order Status Response 40 400 80
4000\r\n\
    Derivery Response 50 500 90
5000\r\n\
    Stock Level Response 60 600 100
6000\r\n\

```

```

</PRE></BODY></HTML>\r\n"
#endif
    char host_name[32];
    SAMPLE_INFO *bp;
    HRESULT ret_value;
    OBJECT_CNTL_INFO* ObjCntlInfo;
    long obj_avr, new_avr, pay_avr, odr_avr,
    del_avr, sto_avr;
    /* Get host name inserting to HTML */
    host_name[0] = '\0';
    gethostname(host_name, sizeof(host_name));
    /* Create SvrApl object & get information buffer */
    ObjCntlInfo = CreateSvrAplExecEnv();
    if (ObjCntlInfo == 0) {
        userlog (LOG_ERR, "Failure create
SvrAPL object(Extended function = -3)");
        return;
    }
    bp = ( SAMPLE_INFO * )ObjCntlInfo->TrxData;
    memset(bp, 0x00, sizeof(SAMPLE_INFO));
    /* Set sampling interface & get sampling data */
    /* NewOrder function has sampling interface. */
    bp->sample_function = -3;
    if ((ret_value = ObjCntlInfo->SvrAplObj-
>ExNewOrder((int *)&bp)) != S_OK) {
        sprintf(html_buf, EXT_FUNC3_ERROR);
        userlog (LOG_ERR, "Extend function 3
abnormal end ret_val=%d\n", ret_value);
        return;
    }
    /* Count empty delivery queues */
    EnterCriticalSection(&GLB_LockDeliveryQue);
    for (empty_que = GLB_DelQueEmptyTop,
empty_delivery_queues = 0;
empty_delivery_queues != 0;
empty_que = empty_que->Next,
empty_delivery_queues++);
    LeaveCriticalSection(&GLB_LockDeliveryQue);
    /* Make HTML data */
    obj_avr = bp->obj_cnt != 0? bp->obj_ttl / bp-
>obj_cnt: 0;
    new_avr = bp->new_cnt != 0? bp->new_ttl / bp-
>new_cnt: 0;
    pay_avr = bp->pay_cnt != 0? bp->pay_ttl / bp-
>pay_cnt: 0;
    odr_avr = bp->odr_cnt != 0? bp->odr_ttl / bp-
>odr_cnt: 0;
    del_avr = bp->del_cnt != 0? bp->del_ttl / bp-
>del_cnt: 0;
    sto_avr = bp->sto_cnt != 0? bp->sto_ttl / bp-
>sto_cnt: 0;
    sprintf(html_buf, CLIENT_SAMPLE ,
host_name,
empty_delivery_queues,
// No. of emptyr queue
GLB_DeliveryQueOver,
// No. of Current overflow queue
GLB_DeliveryQueOverPeek,
// No. of Max overflow queue
GLB_DeliveryQueWaitTime,
// Time of current retry (current thread MAX)
GLB_DeliveryQueWaitTimePeek,
// Time of MAX retry
bp->obj_min, bp->obj_max, obj_avr, bp-
>obj_cnt, // Time of object getting (under construnt)
bp->new_min, bp->new_max, new_avr, bp-
>new_cnt, // Response time of New Order

```

```

bp->pay_min, bp->pay_max, pay_avr, bp-
>pay_cnt, // Response time of Payment
bp->odr_min, bp->odr_max, odr_avr, bp-
>odr_cnt, // Response time of OrderStatus
bp->del_min, bp->del_max, del_avr, bp-
>del_cnt, // Response time of Delivery
bp->sto_min, bp->sto_max, sto_avr, bp-
>sto_cnt); // Response time of Stock Level
    /* Clear sampling ingormation for next sampling
interval */
    GLB_DeliveryQueWaitTime = 0;
#if 0
    /* Make HTML data */
    sprintf(html_buf, CLIENT_SAMPLE ,
host_name,
GLB_DeliveryQueOver, // No. of
Current overflow queue
GLB_DeliveryQueOverPeek, // No. of Max
overflow queue
GLB_DeliveryQueWaitTime, // Time of
current retry (current thread MAX)
GLB_DeliveryQueWaitTimePeek, // Time of
MAX retry
0, 0, 0, // Time of object
getting (under construnt)
0, 0, 0, // Response time
of New Order
0, 0, 0, // Response time
of Payment
0, 0, 0, // Response time
of OrderStatus
0, 0, 0, // Response time
of Delivery
0, 0, 0); // Response time
of Stock Level
#endif
}
TpAplBase/ConvInt.cpp
/*****
*
* 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 <windows.h>
#include "trans.h"
#include "ObjectCntlInfo.h"
#include "Delivery.h"
#include "ExecuteThread.h"
#include "TpAplPrototype.h"
#define numcheck(num) ( 0x30 <= num && num <=
0x39 ) /* 0 - 9 */
#define alpcheck(num) ( 0x41 <= num && num <=
0x5a ) /* A - Z */
/*
str2int :
takes a string, makes sure it's not too long, and
ensures that it

```

```

represents an integer.
If it does, the corresponding int value is returned.
-3: there is not string data.
-2: find not character data.
-1: string data is too many long
*/
int str2int(char *str, int field_len) {
    int x;
    //for warning
// if(str == 0 || !(x = strlen(str))) return -3;
if(str == 0 || (x = strlen(str)) == 0) return -3;
if(x > field_len){
    if (strchr (str, '%') != 0) /* 98.8.3 : */
        return -2;
    else
        return -1;
}
else{
    for( ; x <-){
        if (!numcheck(str[x-1])) {
            return -2;
        }
    }
}
return atoi(str);
}
/*
str2short :
takes a string, makes sure it's not too long, and
ensures that it
represents an integer.
If it does, the corresponding short value is returned.
-3: there is not string data.
-2: find not character data.
-1: string data is too many long
*/
short str2short(char *str, int field_len) {
    int x;
    //for warning
// if(str == 0 || !(x = strlen(str))) return -3;
if(str == 0 || (x = strlen(str)) == 0) return -3;
if(x > field_len){
    if (strchr (str, '%') != 0) /* 98.8.3 : */
        return -2;
    else
        return -1;
}
else {
    for( ; x <-){
        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 7bytes */
    /* char NUM[7];*/
    char NUM[16];
    /* Replaced end */
    char pointf = 0;
    int fcnt = 2; /* */
    //for warning
// if(str == 0 || !(x = strlen(str))) return -3;
if(str == 0 || (x = strlen(str)) == 0) return -3;
len = x;
if(x > field_len){
    if (strchr (str, '%') != 0) /* 98.8.3 : */
        return -2;
    else
        return -1;
}
else{
    /* check string data */
    for(x;x--){
        if(numcheck(str[x-1]));
        else if((str[x-1] == '.') && ((len - x) < 3));
        else if((str[x-1] == '-') && (x == 1));
        else if((str[x-1] == '+') && (x == 1));
        else return -2;
    }
}
/* delete the decimal point. As a result,do
hundredfold the value.*/
for (cnt = 0, x = 0; x < len; x++){
    if (str[x] == '.') {
        /* find the decimal point. set point flag.*/
        pointf = 1;
    } else {
        /* set character to work buffer.*/
        NUM[cnt] = str[x]; cnt++;
        /* The figure below the decimal point was
detected */
        if (pointf == 1) {fcnt--;}
    }
}
if (pointf == 1 && fcnt > 0){
    /*There was no figure below the decimal
point or only one digit was
found.: */
    for ( ; fcnt > 0; fcnt--){
        NUM[cnt++] = '0';
    }
}
else if (pointf == 0) {
    /* There is no decimal point.: */
    NUM[cnt++] = '0'; NUM[cnt++] = '0';
}
NUM[cnt] = 0;
return (atoi(NUM));
}
.....
TpAplBase/ConvOther.cpp
.....
/*
*
* 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 <windows.h>
#include <stdio.h>
#include <time.h>
#include "trans.h"
#include "ObjectCntlInfo.h"
#include "Delivery.h"
#include "ExecuteThread.h"
#include "TpAplPrototype.h"
/*
para_split :
(QueryString)
: .NULL
NULL
Split divides up a string based on the first instance of
a specified
delimiter ('sp'). The first instance of 'sp' is converted
to a NULL
and the address of the first character of the second
half is returned.
Thus the user has the first half (which he passed in
and still has) and
the second half (which was returned) with a NULL
between them. Yay.
(Yes, strtok does this, sort of, but I can't nest strtok
calls.)
*/
char *para_split(char *para, char delimita) {
    char *point = para;;
    /* The address of the delimitation character is
calculated */
    /* */
// if ((point = strchr (para, delimita)) == NULL)
// return (char *)0;
for( !(*point == '\0' || *point == delimita); point++;)
if (*point == '\0')
return (char *)0;
/* The delimitation character is replaced with NULL*/
*point = '\0'; /* NULL */
/* The first position of the analyzed variable is
returned.*/
return (point + 1); /* */
}
/*
check HTML form
*/
int checkHTMLform( char *str, char *buffer)
{
    char* src = str;
    char* dst = buffer;
    while (*src != '\0'){
        if (*src) == '&'{
            *(dst) = '&'; dst++;
            *(dst) = 'a'; dst++;
            *(dst) = 'm'; dst++;
            *(dst) = 'p'; dst++;
            *(dst) = ';'; dst++;
        }
        else if (*src) == '<' {

```

```

        *(dst) = '&'; dst++;
        *(dst) = '!'; 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 )
{
    struct tm* tim;
    time_t tt = (time_t)time;
    tim = localtime( &tt );
    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 )
{
    struct tm* tim;
    time_t tt = (time_t)time;
    tim = localtime( &tt );
    sprintf( save_p, "%02d-%02d-%04d",
            tim->tm_mday, tim->tm_mon + 1, tim-
>tm_year + 1900 );
}
.....
TpAplBase/ConvString.cpp
.....
/*****
*
* 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 <windows.h>
#include <stdio.h>
#include "trans.h"
#include "ObjectCntlInfo.h"
#include "Delivery.h"
#include "ExecuteThread.h"
#include "TpAplPrototype.h"
#include "log.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;
    char *string;
    string = ecvt(num, len-1, &dec, &sign);
    /* dec = , sign = 0,1,string= */
    if (dec > 0) {
        /* if the integer part is not zero ..
Example :num data is 1234.56 */
        cnt = (len - 3) - dec;
        /* : "0012" -> " 12" */
        /* If the high-order digit is zero , zero is changed at
the blank */
        for (i = 0; i < cnt; i++){
            /* pad with blank in the high part of the
number */
            str[i] = ' ';
        }
        /* The high-order digit set to the output
area: */
        for (; i < (len - 3); i++){
            str[i] = *(string++);
        }
    }
    else {
        /* If the integer part is zero ... Example:
num data is 0.12 */
        cnt = len - 4;
        for (i = 0; i < cnt; i++){
            /* pad with blank in the high part of
the number */
            str[i] = ' ';
        }
        str[i++] = '0';
        str[i++] = '0';
        for (; dec < 0 && i < len; dec++, i++){
            /* pad with 0's in the high part of the
fraction */
            str[i] = '0';
        }
        for (; i < len; i++){
            /* copy the decimal portion (2 places) */
            str[i] = *(string++);
        }
    }
}
/*
sigdec2str:
Converts a double precision floating point value to a
string of
a specified length and outputs the string to the
supplied buffer.
If the value is negative, the first character will be a
minus sign (-).
field = the destination field
field_size = number of characters to output
value = floating point number to be displayed
*/
void sigdec2str(char *str, int len, double num)
{
    if (num >= 0.0) {

```

```

        str[0] = '-';
        dec2str (&str[1], len - 1, num);
    } else {
        str[0] = '-';
        dec2str (&str[1], len - 1, -num);
    }
}
/*
str2str :
makes sure the string exists and isn't too long.
-1: string data is too many long
-2: find not figure data.
0: there is not string data.
1: normal end
*/
int str2str(char *str, int field_len) {
    int x;
    //for warning
    // if (str == 0 || !(x = strlen (str))) return 0;
    // if (str == 0 || (x = strlen (str)) == 0) return 0;
    if(x > field_len) {
        if ( strchr (str, '%') != 0) /* 98.8.3 :*/
            return -2;
        else
            return -1;
    }
}
/*
else {
    for (; x < x-){
        if (!alpcheck(str[x-1]))
            return -2;
    }
}
*/
return 1;
}
/*
alp2str : Outputs a string into the memory space
supplied.
field = the destination field
field_size = number of characters to output
string = alpha string to be displayed
*/
void alp2str(char *str, int len, char *alp)
{
    int cnt;

    cnt = strlen (alp);
    strncpy (str, alp, len); /* copy to destination area */
    /* len*/
    /* If not coming up to the specified length then set the
blank. */
    if ( len - cnt > 0 )
        memset ( &str[cnt], ' ', len - cnt);
}
/*
date2str : Outputs a date in the supplied buffer in the
following format:
DD-MM-YYYY
field = the destination field
date = date to be converted and displayed
*/
void date2str(char *str, char *time)
{
    int year, month, day;
#ifdef DBPRT
    fprintf (test_fp, "date2: %s\n", time);
#endif
    sscanf( time, "%d-%d-%d", &day, &month, &year );

```

```

int3str (str, 2, day);
str[2] = '-';
int3str (&str[3], 2, month);
str[5] = '-';
int3str (&str[6], 4, year);
}
/*
zip2str:
Outputs a zipcode in the supplied buffer in the
following format:
XXXXX-XXXX
str = the destination field
zip = the zipcode to be output
*/
void zip2str (char *str, char *zip)
{
    alp2str (str, 5, zip);
    str[5] = '-';
    alp2str (&str[6], 4, &zip[5]);
}
/*
phone2str:
Outputs a phone number in the supplied buffer in the
following format:
XXXXXX-XXX-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]);
}
.....
TpApBase/Delivery.h
.....
/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions *
* Delivery structure object. *
*
* CREATE by TSL 2002.12.11 *
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****/
typedef struct _delivery_que {
    _delivery_que *Next;
    byte TnxData[4];
} DELIVERY_QUE;
#define EVENT_NAME_START
"TPCC_Delivery_StartEvent"
#define EVENT_NAME_CHECK
"TPCC_Delivery_CheckEvent"
/* Added T.Kato 03.08.22 Change up Delivery */
#define EVENT_NAME_QUEUE_EMPTY
"TPCC_Delivery_Queue_empty"
/* Added end */
.....
TpApBase/ErrPage.cpp
.....

```

```

/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions *
* (1) set_errHTML *
* (2) set_SvrApiErr *
* (3) set_errpage *
*
* CREATE by TSL 2002.10.01 *
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****/
#include <windows.h>
#include <stdio.h>
#include "trans.h"
#include "tpcweb.h"
#include "tpccis.h"
#include "ObjectCntlInfo.h"
#include "TpcCntlInfo.h"
#include "Delivery.h"
#include "tpapl.h"
#include "ExecuteThread.h"
#include "GlobalArea.h"
#include "TpApiPrototype.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"OracleSymfo(set_errHTML)
! set_orraerr :
! this function make error message of the Oracle
application program.
!*/
!int set_orraerr (char *page, char *err_inf, int cookie ) {
!
!#ifdef Symfo
! sprintf(page, symfoerr, err_inf, SOPATH, cookie );
!#else
! sprintf(page, orraerr, 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) {
#endef
int set_SvrApiErr (char *page, char *err_inf, int cookie) {
/* Replaced end */
    sprintf(page, tuxerr, err_inf, SOPATH, cookie);
    return 0;
}
}

```

```

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

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

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

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

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

"The Carrier ID you entered isn't valid.\r\n%s\r\n"
"It must be an integer in the range 1 to 10.\r\n",
/* 5 */

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

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

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

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

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

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

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

"On entry line %d, the data you entered for the %s field
isn't valid.\r\n%s\r\n", /* 13 */
"Supply Warehouse ID",
/* 14 */

"Item ID",
/* 15 */

"Quantity",
/* 16 */

>Your entry was outside the range.",
/* 17 */

>You didn't entry anything for the field.",
/* 18 */

>Your entry contained too many characters.",
/* 19 */

"The input data is wrong data type, must be numeric.",
/* 20 */

"It must be an integer in the range 1 to %d.",
/* 21 */

"The input data is wrong data type, must be english
capital letter.", /* 22 */
};
/*
set_errpage:
RTE
a generic error page generator. If the user does
anything screwy,
s/he gets here. The function generates an error page
based on the
two errlvl arguments and returns it for the user..

When err_no is 13 or more, Order Line Data is
Abnormal.
(err_no is the error data line number )
98.8.3 :
*/
int set_errpage (char *buf, int user, int err_no, int err_inf,
int sub_inf, int sub_inf2) {
char errmsg[1024];
int nchar;
int length;
//for warning
sub_inf;
nchar;
if(err_no >= 13) { /* OrderLine
Data(Neworder) is Abnormal */
switch(err_inf) {
case -5: /* S_W_ID data is
abnormal */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[14], errstrings[20]);
sub_inf2 = GLB_Numwh;
break;
case -8: /* S_W_ID data is
uninput */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[14], errstrings[18]);
sub_inf2 = GLB_Numwh;
break;
case -15: /* S_W_ID data is
outside range */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[14], errstrings[17]);
sub_inf2 = GLB_Numwh;
break;
case -1: /* I_ID data is
uninput */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[15], errstrings[18]);
sub_inf2 = 100000;
break;
case -6: /* I_ID data is
abnormal */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[15], errstrings[20]);
sub_inf2 = 100000;
break;
case -16: /* I_ID data is
outside range */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[15], errstrings[17]);
sub_inf2 = 100000;
break;
case -7: /* Quantity data is
abnormal */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[16], errstrings[20]);
sub_inf2 = 10;
break;
case -2: /* Quantity data is
uninput */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[16], errstrings[18]);
sub_inf2 = 10;
break;
case -17: /* Quantity data is
outside range */
sprintf(errmsg, errstrings[13], err_no-
12, errstrings[16], errstrings[17]);
sub_inf2 = 10;
break;
default:
break;
}
length = strlen(errmsg);
sprintf(&errmsg[length], errstrings[21],
sub_inf2);
user);
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;
}
.....
TpAplBase/ExecuteThread.h
.....
/*****
*
*      TPC-C Client Application Program Source
*
*
*      Entry Functions
*      Transaction executing information
*
*      CREATE by TSL 2003.02.05
*
*      All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003
*****/
#ifndef EXECUTETHREAD_H
#define EXECUTETHREAD_H
/* Transaction executing thread control information */
typedef struct _execute_thread_info {
    /* Thread control information */
    _execute_thread_info *Before;
    _execute_thread_info *Next;
    DWORD                ThreadId;
    HANDLE               ExThreadPostEvent;
    HANDLE               ExThreadRespEvent;
    int                  ThreadTerminate;
#define EXEC_THREAD_BEGIN 0
#define EXEC_THREAD_TERM 1
    /* Executing transaction information */
    char                StringBuffer;
    RTE_INPUT_DATA*    RteInputData;
    int                 Cookie;
    int                 ReturnValue;
} EXECUTE_THREAD_INFO;
/* Event base name (when using, add the thread id) */
#define EXEC_TDREAD_EVENT_NAME_POST
"TPCC_ExecTread_PostEvent"

```

```

#define
EXEC_TDREAD_EVENT_NAME_RESP
"TPCC_ExecTread_RespEvent"
#ifdef // EXECUTETHREAD_H
.....
TpAplBase/ExecuteTrans.cpp
.....
/*****
*
*      TPC-C Client Application Program Source
*
*
*      Entry Functions
*      (1) ExecuteTransaction
*      (2) TransExCntlThread
*      (3) DeleteExThreadInfo
*      (4) DetachExecuteThread
*      (5) TermExecuteThread
*
*      CREATE by TSL 2003.02.05
*
*      All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003
*****/
#include <windows.h>
#include <process.h>
#include <stdio.h>
#include "trans.h"
#include "TpcCntlInfo.h"
#include "Delivery.h"
#include "ObjectCntlInfo.h"
#include "ExecuteThread.h"
#include "GlobalArea.h"
#include "TpAplPrototype.h"
#include "TpAplDBDependPrototype.h"
#include "log.h"
#include "menupage.h"
/*****
*      Executing Transaction
*      Return Value
*      Transaction executed result
*      0:Normal end
*      !0:Abnormal end
*****/
int XExecuteTransaction(char *s_buf,
RTE_INPUT_DATA *in_data, int cookie) {
    EXECUTE_THREAD_INFO*
execute_thread_info = 0;
    int
return_value;
    char                event_name[64];
    char                thread_id[16];
    static int          tls_zero = 0;
    MAC_PutFncEntryLog("ExecuteTransaction");
    /* Initialize information of the Transaction Execute
Control Thread */
    execute_thread_info =
(EXECUTE_THREAD_INFO *)TlsGetValue(GLB_TlsIndex);
    if(execute_thread_info == 0) {
        /* First time execution, lock the thread */
        EnterCriticalSection(&GLB_LockTransExecCntl);
        /* Allocate control information area */
        execute_thread_info =
(EXECUTE_THREAD_INFO *)calloc(1,
sizeof(EXECUTE_THREAD_INFO));
        if(execute_thread_info == NULL) {

```

```

        userlog (LOG_ERR, "Execute thread
Information area fail to allocation. [size=%d]\n",
sizeof(EXECUTE_THREAD_INFO));
        goto INIT_ERR_END_UNLOCK;
    }
    /* Get controlling event each thread */
    itoa(GetCurrentThreadId(), thread_id,
sizeof(thread_id));
    strcpy(event_name,
EXEC_TDREAD_EVENT_NAME_POST);
    strcat(event_name, thread_id);
    execute_thread_info->ExThreadPostEvent
= NULL;
    if ((execute_thread_info-
>ExThreadPostEvent =
CreateEvent(NULL, FALSE, FALSE,
event_name)) == NULL)
    {
        userlog (LOG_ERR, "Executing
thread controlling event fail to create (%s)\n",
event_name);
        goto INIT_ERR_END;
    }
    strcpy(event_name,
EXEC_TDREAD_EVENT_NAME_RESP);
    strcat(event_name, thread_id);
    execute_thread_info->ExThreadRespEvent
= NULL;
    if ((execute_thread_info-
>ExThreadRespEvent =
CreateEvent(NULL, FALSE, FALSE,
event_name)) == NULL) {
        userlog (LOG_ERR, "Executing
thread controlling event fail to create (%s)\n",
event_name);
        goto INIT_ERR_END;
    }
    /* Set control information pointer to TLS */
    return_value = TlsSetValue(GLB_TlsIndex,
(void *)execute_thread_info);
    if (return_value == 0) {
        userlog (LOG_ERR, "TlsSetValue
abnormal end (TlsIndex=%d)\n", GLB_TlsIndex);
        goto INIT_ERR_END;
    }
    /* Create The Transaction Execute
Controlling thread */
    return_value =
(int)_beginthread(TransExCntlThread, 0, (void
*)execute_thread_info);
    if (return_value == -1) {
        userlog (LOG_ERR, "Transaction
Execute Controlling thread fail to create.(errno:%d)",
errno);
        goto INIT_ERR_END;
    }
    /* Connect information queue */
    if (GLB_ExThreadInfoLast == 0) {
        /* 1st position queued */
        GLB_ExThreadInfoTop =
execute_thread_info;
    } else {
        execute_thread_info->Before = 0;
        /* Other position queue */

```

```

        GLB_ExThreadInfoLast->Next =
execute_thread_info;
        execute_thread_info->Before =
GLB_ExThreadInfoLast;
    }
    GLB_ExThreadInfoLast =
execute_thread_info;
    execute_thread_info->Next = 0;
    /* Unlock thread & Sleep few sec.*/

    LeaveCriticalSection(&GLB_LockTransExecCntl);
    Sleep(1000*10);
}
/* Set transaction executing parameter into
control information */
execute_thread_info->StringBuffer = s_buf;
execute_thread_info->RtInpData = in_data;
execute_thread_info->Cookie = cookie;
/* Request transaction executing, and wait
executing end */
if (!SetEvent(execute_thread_info-
>ExThreadPostEvent)) {
    userlog (LOG_ERR, "Transaction
executing fail to request (SetEvent).
[GetLastError:%d]\n",
        GetLastError());
    goto EXEC_ERR_END;
}
if (WaitForSingleObject(execute_thread_info-
>ExThreadRespEvent, INFINITE) !=
WAIT_OBJECT_0) {
    userlog (LOG_ERR, "Transaction
executing fail to wait (WaitForSingleObject).
[GetLastError:%d]\n",
        GetLastError());
    goto EXEC_ERR_END;
}
/* Transaction normal execute, return to caller */
return execute_thread_info->ReturnValue;
/*****
 * Error processing
 *****/
INIT_ERR_END:
if (execute_thread_info->ExThreadPostEvent !=
0)
    CloseHandle(execute_thread_info-
>ExThreadPostEvent);
if (execute_thread_info->ExThreadRespEvent !=
0)
    CloseHandle(execute_thread_info-
>ExThreadRespEvent);

free(execute_thread_info);
return_value = TlsSetValue(GLB_TlsIndex, (void
*)tls_zero);
if (return_value == 0) {
    userlog (LOG_ERR, "TlsSetValue
abnormal end in create the control thread.
(TlsIndex=%d) \n", GLB_TlsIndex);
}
INIT_ERR_END_UNLOCK:
LeaveCriticalSection(&GLB_LockTransExecCntl);
sprintf(s_buf, initerr);
return -100;
EXEC_ERR_END:
sprintf(s_buf, cntlerr);
return -101;
}
/*****

```

```

 * Transaction Execute control thread
 * Return Value
 * Transaction executed result
 * None return value
 *****/
void XTransExCntlThread(void* thread_info) {
    EXECUTE_THREAD_INFO* ExecThdInfo;
    MAC_PutFncEntryLog("TransExCntlThread");
    userlog (LOG_INF, "Transaction execute control
thread start");
    /* Save execute thred information */
    ExecThdInfo = (EXECUTE_THREAD_INFO
*)thread_info;
    ExecThdInfo->ThreadId = GetCurrentThreadId();
    for (;;) {
        /* Wait Transaction executing request */
        if (WaitForSingleObject(ExecThdInfo-
>ExThreadPostEvent, INFINITE) != WAIT_OBJECT_0)
        {
            userlog (LOG_ERR, "Transaction
execute request waiting failed (WaitForSingleObject)
[GetLastError:%d]\n", GetLastError());
            goto End_Thread;
        }
        /* End thread check */
        if (ExecThdInfo->ThreadTerminate ==
EXEC_THREAD_TERM) {
            goto End_Thread;
        }
        /* check transaction type */
        switch(ExecThdInfo->RtInpData->form[0]) {
            case 'N':
                ExecThdInfo->ReturnValue = NewOrder
(ExecThdInfo->StringBuffer, ExecThdInfo-
>RtInpData, ExecThdInfo->Cookie);
                break;
            case 'D':
                ExecThdInfo->ReturnValue =
Delivery(ExecThdInfo->StringBuffer, ExecThdInfo-
>RtInpData, ExecThdInfo->Cookie);
                break;
            case 'P':
                ExecThdInfo->ReturnValue = Payment
(ExecThdInfo->StringBuffer, ExecThdInfo-
>RtInpData, ExecThdInfo->Cookie);
                break;
            case 'S':
                ExecThdInfo->ReturnValue =
StockLevel(ExecThdInfo->StringBuffer, ExecThdInfo-
>RtInpData, ExecThdInfo->Cookie);
                break;
            case 'O':
                ExecThdInfo->ReturnValue =
OrderStatus (ExecThdInfo->StringBuffer, ExecThdInfo-
>RtInpData, ExecThdInfo->Cookie);
                break;
            default:
                /* uninput transaction type */
                set_errpage(ExecThdInfo-
>StringBuffer, ExecThdInfo->Cookie, 1, -4, 0, 0);
                ExecThdInfo->ReturnValue = 1;
                break;
        }
        if (!SetEvent(ExecThdInfo-
>ExThreadRespEvent)) {

```

```

            userlog (LOG_ERR, "Transaction
executing end fail to post (SetEvent).
[GetLastError:%d]\n", GetLastError());
            goto End_Thread;
        }
    }
End_Thread:
#ifdef PUT_INF_LOG
    userlog (LOG_INF, "Transaction execute control
thread terminate started");
#endif
    DetachExtensionThread();
    if (!SetEvent(ExecThdInfo-
>ExThreadRespEvent)) {
        userlog (LOG_ERR, "Transaction
executing terminating fail to post (SetEvent).
[GetLastError:%d]\n", GetLastError());
    }
#ifdef PUT_INF_LOG
    userlog (LOG_INF, "Transaction execute control
thread terminate ended");
#endif
    _endthread();
    return;
}
/*****
 * Terminate transaction excute control thread .
 * Delete transaction execute thread control information.
 *
 * Argument
 * Execute Thread information pointer
 *****/
void
XDeleteExThreadInfo(EXECUTE_THREAD_INFO*
execute_thread_info) {
    MAC_PutFncEntryLog("DeleteExThreadInfo");

    /* Check request thread terminate */
    if (execute_thread_info == 0)
        return;

#ifdef PUT_INF_LOG
    userlog (LOG_INF, "@@@ Execute control
therad terminateing start [Thread id = %d]\n",
        execute_thread_info-
>ThreadId);
#endif

    /* Execute equest */
    EnterCriticalSection(&GLB_LockTransExecCntl);
    execute_thread_info->ThreadTerminate =
EXEC_THREAD_TERM;
    if (!SetEvent(execute_thread_info-
>ExThreadPostEvent)) {
        userlog (LOG_ERR, "Terminate requesting
fail for execute control thread (SetEvent).
[GetLastError:%d]\n",
            GetLastError());
    }
    if
(WaitForSingleObject((HANDLE)execute_thread_info-
>ExThreadRespEvent, 3*60*1000) !=
WAIT_OBJECT_0) {
        userlog (LOG_ERR, "Terminate waiting
end of thread fail for execute control thread "
"(WaitForSingleObject). [GetLastError:%d]\n",
            GetLastError());
    }
}

```

```

/* Close event handle */
if (!CloseHandle(execute_thread_info->ExThreadPostEvent)) {
    userlog (LOG_ERR, "Handle of execute
thread control fail to close (for post).
[GetLastError:%d]\n",
                GetLastError());
}
if (!CloseHandle(execute_thread_info->ExThreadRespEvent)) {
    userlog (LOG_ERR, "Handle of execute
thread control fail to close (for Response).
[GetLastError:%d]\n",
                GetLastError());
}
/* Re-connect ect control information */
if (execute_thread_info->Before != 0) {
    /* Not first queue */
    execute_thread_info->Before->Next =
execute_thread_info->Next;
}
if (execute_thread_info->Next != 0) {
    /* Not last queue */
    execute_thread_info->Next->Before =
execute_thread_info->Before;
}
if (execute_thread_info->Before == 0) {
    /* First queue */
    GLB_ExThreadInfoTop =
execute_thread_info->Next;
}
if (execute_thread_info->Next == 0) {
    /* Last queue */
    GLB_ExThreadInfoLast =
execute_thread_info->Before;
}
/* Free object control information */
free((void *)execute_thread_info);
LeaveCriticalSection(&GLB_LockTransExecCntl);
#ifdef PUT_INF_LOG
    userlog (LOG_INF, "@@@ Execute control
therad terminateing end\n");
#endif

return;
}
/*****
 * THREAD_DETACH Process.
 * Terminate the Transaction Execute Control Thread.
 *
 * Return Value
 * None
 *****/
void XDetachExecuteThread() {
    EXECUTE_THREAD_INFO*
execute_thread_info;
    static int x = 0;
    MAC_PutFncEntryLog("DetachExecuteThread");
    /* Get object control information pointer */
    execute_thread_info =
(EXECUTE_THREAD_INFO*)TlsGetValue(GLB_TlsInd
ex);
    if(execute_thread_info == 0) {
        userlog (LOG_WRN, "ExecuteControl
thread not initialized (Control thread not created)\n");
        return;
    }
    /* Free object control information */
    DeleteExThreadInfo(execute_thread_info);
}

/* Initialize object control information */
TlsSetValue(GLB_TlsIndex, (void*)x);
return;
}
/*****
 * PROCESS_DETACH Process.
 * Terminate all transaction execute control thread
 *
 * Return Value
 * None
 *****/
void XTermExecuteThread() {
    int i;
    EXECUTE_THREAD_INFO* ex_tread_inf;

    MAC_PutFncEntryLog("TermExecuteThread");
    for (i = 0, ex_tread_inf = GLB_ExThreadInfoTop;
ex_tread_inf != 0;
i++, ex_tread_inf = ex_tread_inf->Next);
    userlog (LOG_INF, "All execute control therad
terminateing start [%d threads]\n", i);
    while (GLB_ExThreadInfoTop != 0) {

        DeleteExThreadInfo(GLB_ExThreadInfoTop);
    }
    return;
}
:-----:
TpAplBase/GlobalArea.cpp
:-----:
/*****
 *
 * TPC-C Client Application Program Source
 *
 * Entry Functions
 * Global Area definition for common.
 *
 * CREATE by TSL 2002.10.01
 *
 * All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002
 *****/
#include <windows.h>
#include "tpccis.h"
#include "tpapl.h"
#include "ObjectCntlInfo.h"
#include "TpcCntlInfo.h"
#include "Delivery.h"
#include "trans.h"
#include "ExecuteThread.h"
int TRN_ID[5];
#ifdef SCRTEST
char SOPATH[] = "dbgSD/tpapl.dll"; /*
DEBG Mode: SCRTEST & DBPRT */
#else
char SOPATH[] = "/tpc/tpapl.dll"; /* DEBG
Mode: SCRTEST */
#endif
#else
char SOPATH[] = "/tpc/tpapl.dll"; /* DEGB
Mode: */
#else
char SOPATH[] = "/tpc/tpapl.dll"; /* Release
Mode: */
#endif
#endif
}

#ifdef DBPRT /* for debug */
FILE *test_fp;
#endif
/* CriticalSection of SvrAPL execution environment
process */
CRITICAL_SECTION GLB_LockSvrAplExecEnv;
/* CriticalSection of Transaction execute controlling
thread */
CRITICAL_SECTION GLB_LockTransExecCntl;
/* Thread Local Storage key */
DWORD GLB_TlsIndex;
/* Queue pointer */
OBJECT_CNTRL_INFO* GLB_ObjMngTop = 0;
OBJECT_CNTRL_INFO* GLB_ObjMngLast = 0;
/* Registry Key */
HKEY m_tpccregkey;
/* Environment of operation */
int GLB_TermBase;
int GLB_Numwh;
int GLB_Maxconnect;
int GLB_Maxterm;
int GLB_C_FLAG;
int GLB_NumDeliveryThreads;
int GLB_NumDeliveryQueus;

/* Added T.kato 03.07.28 Speed up */
int GLB_DeliveryPriority;
/* Added end */
/* Terminal Control Information */
TC_CONTEXT* GLB_TcContext;
TERMINAL_CNTRL GLB_TerminalCntl;
TERMINAL_CNTRL *srv =
&GLB_TerminalCntl;
/* Own DLL path */
char GLB_DefaultPath[MAX_PATH];
/* Log file path (extern written by log.h) */
char GLB_LogFilePath[MAX_PATH];
/* Initialization failure flag */
int GLB_InitCond;
#ifdef KEEP_OBJ
/* Transaction thread control information */
!EXECUTE_THREAD_INFO* GLB_ExThreadInfoTop
= 0;
!EXECUTE_THREAD_INFO* GLB_ExThreadInfoLast
= 0;
#endif
/* ===== */
/* for Delivery Variable */
/* ===== */
/* All queue area */
void* GLB_DelQueArea = 0;
/* Queue pointer */
DELIVERY_QUE* GLB_DelQueEmptyTop
= 0;
DELIVERY_QUE* GLB_DelQueEmptyLast
= 0;
DELIVERY_QUE* GLB_DelQueProcTop =
0;
DELIVERY_QUE* GLB_DelQueProcLast =
0;
/* CriticalSection of Delivery queue */
CRITICAL_SECTION GLB_LockDeliveryQue;
/* Start event */
HANDLE GLB_DelStrEvent =
NULL;

```

```

/* Check event */
HANDLE          GLB_DelConfEvent =
NULL;
/* Number of created Delivery threads */
int
    GLB_NumCreateThreads = 0;
/* DeliveryThread return value */
int
    GLB_ThreadCond = 0;
/* Thread end flag */
byte
    GLB_ThreadEndReq = 0;
/* Create Delivery thread flag */
int
    GLB_InitDelivery = 0;
/* Added T.Kato 03.08.04 Delivery queue overflow */
int
    GLB_DeliveryQueueOver =
0;
int
    GLB_DeliveryQueueOverPeek = 0;
int
    GLB_DeliveryQueueWaitTime = 0;
int
    GLB_DeliveryQueueWaitTimePeek = 0;
/* Added end */
/* Added T.Kato 03.08.22 Delivery change up */
HANDLE          GLB_DelQueueWaitEvent
= NULL;
/* Added end */
.....
TpAplBase/GlobalArea.h
.....
*
*   TPC-C Client Application Program Source
*
*
*   Entry Functions
*   Global Area definition for common.
*
*   CREATE by TSL 2002.10.01
*
*   All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
.....
#ifndef GLOBALAREA_H
#define GLOBALAREA_H
/* Deleted 03.01.15 */
#if 0
!extern int  Term_Base;
!extern int  maxterm;
!extern int  maxwh;
!extern int  maxconnect;
!extern int  C_FLAG;
#endif
/* Deleted end */
extern int  TRN_ID[5];
/* Deleted 03.01.15 */
#if 0
!extern HKEY m_tpccregkey;
!extern TPINIT *tpinf;
!extern DWORD TLSIsTpnitedKey;
#endif
/* Deleted end */
#ifdef SCRTST
#ifdef DBPRT
extern char SOPATH[]; /* DEBG Mode:
SCRTST & DBPRT */
#else
extern char SOPATH[]; /* DEBG Mode:
SCRTST */
#endif
#endif
#else
#ifdef NOSCR
extern char SOPATH[]; /* DEGB Mode: */
#else
extern char SOPATH[]; /* Release Mode:
*/
#endif
#endif
#ifdef DBPRT /* for debug */
extern FILE *test_fp;
#endif
/*
=====
/* HttpExtensionProc_Thread Workspace (Test
Program) */
/*
=====
/* CriticalSection of SvrAPL execution environment
process */
extern CRITICAL_SECTION
    GLB_LockSvrAplExecEnv;
/* CriticalSection of Transaction execute controlling
thread */
extern CRITICAL_SECTION
    GLB_LockTransExecCntl;
/* Thread Local Storage key */
extern DWORD          GLB_TlsIndex;
/* Queue pointer */
extern OBJECT_CNTRL_INFO* GLB_ObjMngTop;
extern OBJECT_CNTRL_INFO* GLB_ObjMngLast;
/* Registry Key */
extern HKEY
    m_tpccregkey;
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 int          GLB_NumDeliveryThreads;
#define DEFAULT_NUMDELIVERYTHREADS
80
extern int          GLB_NumDeliveryQueus;
#define DEFAULT_NUMDELIVERYQUEUEUS 800
/* Added T.kato 03.07.28 Speed up */
extern int
    GLB_DeliveryPriority;
#define DEFAULT_DELIVERYPRIORITY 0
#define DELIVERYPRIORITY_1
1
#define DELIVERYPRIORITY_2
2
#define DELIVERYPRIORITY_3
3
#define DELIVERYPRIORITY_4
4
#define DELIVERYPRIORITY_5
5
/* Added end */
/* Terminal Control Information */
extern TC_CONTEXT* GLB_TcContext;
extern TERMINAL_CNTRL
    GLB_TerminalCntl;
extern TERMINAL_CNTRL *srv;
/* Own DLL path */
extern char
    GLB_DefaultPath[MAX_PATH];
/* Log file path (extern written by log.h) */
extern char
    GLB_LogFilePath[MAX_PATH];
/* Initialization failure flag */
extern int          GLB_InitCond;
#if 0 // KEEP-OBJ
!/* Transaction thread contol information */
!extern EXECUTE_THREAD_INFO*
    GLB_ExtThreadInfoTop;
!extern EXECUTE_THREAD_INFO*
    GLB_ExtThreadInfoLast;
#endif
/*
=====
/* Delivery Queue Workspace (Test Program)
*/
/*
=====
/* All queue area */
extern void*          GLB_DelQueueArea;
/* Queue pointer */
extern DELIVERY_QUE* GLB_DelQueueEmptyTop;
extern DELIVERY_QUE* GLB_DelQueueEmptyLast;
extern DELIVERY_QUE* GLB_DelQueueProcTop;
extern DELIVERY_QUE* GLB_DelQueueProcLast;
/* CriticalSection of Delivery queue */
extern CRITICAL_SECTION GLB_LockDeliveryQueue;
/* Start event */
extern HANDLE          GLB_DelStrEvent;
/* Check event */
extern HANDLE          GLB_DelConfEvent;
/* Number of created Delivery threads */
extern int
    GLB_NumCreateThreads;
/* DeliveryThread return value */
extern int          GLB_ThreadCond;
/* Thread end flag */
extern byte          GLB_ThreadEndReq;
/* Create Delivery thread flag */
extern int          GLB_InitDelivery;
/* Added T.Kato 03.08.04 Delivery queue overflow */
extern int          GLB_DeliveryQueueOver;
extern int
    GLB_DeliveryQueueOverPeek;
extern int
    GLB_DeliveryQueueWaitTime;
extern int
    GLB_DeliveryQueueWaitTimePeek;
/* Added end */
/* Added T.Kato 03.08.22 Delivery change up */
extern HANDLE          GLB_DelQueueWaitEvent;
/* Added end */
#endif // GLOBALAREA_H
.....
TpAplBase/InitDelivery.cpp
.....

```

```

/*****
 *
 * TPC-C Client Application Program Source
 *
 *
 * Entry Functions
 * (1) InitDelivery
 * (2) InitDelivery_Queue
 * (3) InitDelivery_Thread
 * (4) DeliveryThread
 * (5) DeliveryRequest
 * (6) TermDelivery
 * (7) SetDeliveryThreadPriority (Added 03.07.28)
 *
 *
 * CREATE by TSL 2002.12.11
 *
 * All Right Reserved, Copyright Co. FUJITSU
 LIMITED 2002 *
 *****/

#include <windows.h>
#include <process.h>
#include "trans.h"
#include "ObjectCntlInfo.h"
#include "TpcCntlInfo.h"
#include "Delivery.h"
#include "ExecuteThread.h"
#include "GlobalArea.h"
#include "SvrAplBase.h"
#include "TpAplPrototype.h"
#include "TpAplDBDependPrototype.h"
#include "log.h"
/*****
 * Delivery queue and delivery thread are initialized.
 *
 * Return Value
 * 0 : Normal end
 * -1 : Get memory error
 * -2 : Initialize delivery thread
 * -3 : Get event error
 * -4 : [NumDeliveryQueus] is ZERO
 * -5 : [NumDeliveryThreads] is ZERO
 *****/

int InitDelivery() {
    int return_value;
    MAC_PutFncEntryLog("InitDelivery");
    /* Initialize delivery queue */
    return_value = InitDelivery_Queue();
    if (return_value != 0) {
        userlog (LOG_ERR, "Error end
InitDelivery_Queue() [Return_Value:%d]\n",
return_value);
        goto InitDeliveryErr;
    }
    /* CriticalSection is initialized */
    InitializeCriticalSection(&GLB_LockDeliveryQue);
    /* Initialize delivery thread */
    return_value = InitDelivery_Thread();
    if (return_value != 0) {
        userlog (LOG_ERR, "Error end
InitDelivery_Thread() [Return_Value:%d]\n",
return_value);
        userlog (LOG_INF, "Number of
threads[%d/%d]\n", GLB_NumCreateThreads,
GLB_NumDeliveryThreads);
        goto InitDeliveryErr;
    }
    return return_value;
InitDeliveryErr:

```

```

    TermDelivery();
    return return_value;
}
/*****
 * Delivery queue is initialized.
 * Return Value
 * 0 : Normal end
 * -1 : Get memory error
 * -4 : [NumDeliveryQueus] is ZERO
 *****/

int InitDelivery_Queue() {
    int struct_size;
    unsigned long del_que_size;
    DELIVERY_QUE* que_pointer;
    int que_count;
    MAC_PutFncEntryLog("InitDelivery_Queue");
    if (GLB_NumDeliveryQueus <= 0) {
        userlog (LOG_ERR, "[NumDeliveryQueus]
is ZERO\n");
        return -4;
    }
    /* Delivery request queue size */
    struct_size = sizeof(DELIVERY_QUE) -
sizeof(((DELIVERY_QUE*)0)->TnxData);
    del_que_size = (struct_size +
GetDeliveryDataLen() + 3) / 4 * 4;
    GLB_DelQueArea =
calloc(GLB_NumDeliveryQueus, del_que_size);
    if (GLB_DelQueArea == NULL) {
        userlog (LOG_ERR, "Get queue memory
error\n");
        return -1;
    }
    /* Connect queue */
    que_pointer = (DELIVERY_QUE
*)GLB_DelQueArea;
    for (que_count = 0; que_count <
(int)GLB_NumDeliveryQueus - 1; que_count++) {
        que_pointer->Next = (DELIVERY_QUE
*)((byte *)que_pointer + del_que_size);
        que_pointer = que_pointer->Next;
    }
    que_pointer->Next = 0;
    /* Top Queue and Last Queue are memorized */
    GLB_DelQueEmptyTop = (DELIVERY_QUE
*)GLB_DelQueArea;
    GLB_DelQueEmptyLast = que_pointer;
    /* Added T.Kato 03.08.22 Change up Delivery */
    GLB_DelQueWaitEvent = NULL;
    /* Create check event */
    if ((GLB_DelQueWaitEvent =
CreateEvent(NULL,
FALSE,
FALSE,
EVENT_NAME_QUE_EMPTY)) == NULL)
    {
        userlog (LOG_ERR, "failure create
Event(EventName:%s)\n",
EVENT_NAME_QUE_EMPTY);
        return -4;
    }
    /* Added end */
    return 0;
}
/*****
 * Delivery thread is initialized.
 * Return Value
 * 0 : Normal end

```

```

 * -2 : Create thread error
 * -3 : Create event error
 * -5 : [NumDeliveryThreads] is ZERO
 *****/

int InitDelivery_Thread() {
    long thread_count;
    unsigned long return_value;
    MAC_PutFncEntryLog("InitDelivery_Thread");
    if (GLB_NumDeliveryThreads <= 0) {
        userlog (LOG_ERR,
"[NumDeliveryThreads] is ZERO\n");
        return -5;
    }
    GLB_DelStrEvent = NULL;
    /* Create start event */
    if ((GLB_DelStrEvent =
CreateEvent(NULL,
FALSE,
FALSE,
EVENT_NAME_START)) == NULL)
    {
        userlog (LOG_ERR, "failure create
StartEvent(EventName:%s)\n",
EVENT_NAME_START);
        return -3;
    }
    GLB_DelConfEvent = NULL;
    /* Create check event */
    if ((GLB_DelConfEvent =
CreateEvent(NULL,
FALSE,
FALSE,
EVENT_NAME_CHECK)) == NULL)
    {
        userlog (LOG_ERR, "failure create
CheckEvent(EventName:%s)\n",
EVENT_NAME_CHECK);
        return -3;
    }
    GLB_ThreadCond = 0;
    GLB_NumCreateThreads = 0;
    for (thread_count = 0; thread_count <
GLB_NumDeliveryThreads; thread_count++) {
        /* Begin thread */
        return_value =
_beginthread(DeliveryThread, 0, (void
*)GLB_NumCreateThreads);
        if (return_value == -1) {
            userlog (LOG_ERR, "failure
_beginthread.CreateThreads = %d(erro:%d)"
"[Return_Value:%d]\n", GLB_NumCreateThreads,
errno, return_value);
            return -2;
        }
    }
    /* Added T.Kato 03.07.28 Speed up */
    SetDeliveryThreadPriority((HANDLE)return_valu
e);
    /* Added end */
    /* Wait for thread initialize end */
    /* Replaced T.Kato 03.02.12 Too short timeout 10sec
>3min */
    /* if
(WaitForSingleObject(GLB_DelConfEvent, 1000 *
10) != WAIT_OBJECT_0) {*/

```

```

        if
        (WaitForSingleObject(GLB_DelConfEvent, 1000 * 60 *
        3) != WAIT_OBJECT_0) {
        /* Replaced end */
            userlog (LOG_ERR,
            "WaitForSingleObject/Initialize thread(TimeOut 3min)
            (ThreadNo:%d)[GetLastError:%d]\n",
            GLB_NumCreateThreads, GetLastError());
            return -2;
        }
        /* Create thread error */
        if (GLB_ThreadCond < 0) {
            userlog (LOG_ERR, "Failure initialize
            thread(ThreadNo:%d)[Return_Value:%d]\n",
            GLB_NumCreateThreads, GLB_ThreadCond);
            return -2;
        }
        /* Create thread count */
        GLB_NumCreateThreads += 1;
    }
    return 0;
}
/* Delivery thread execute processing.
* Argument
* Thread discernment
*/
void DeliveryThread(void* thread_no) {
    unsigned long del_size = 0;
    ISvrApl* pISvrApl;
    void* del_area = 0;
    DELIVERY_QUE* get_que;
    int return_value;
    int thread_number;
    int proc_count = 0;

    MAC_PutFncEntryLog("DeliveryThread");
#ifdef PUT_INF_LOG
    userlog (LOG_INF, "Delivery Thread initialize
    start [thread no. = %d]\n", (unsigned long)thread_no);
#endif

    thread_number = (int)thread_no;
    for (;;) {
        /* Create SvrAPL object */
        pISvrApl = CreateSvrAplObject(TRUE);
        if (pISvrApl == 0) {
            userlog (LOG_ERR, "Failure create
            SvrAPL object(ThreadNo:%d)\n", thread_number);
            GLB_ThreadCond = -2;
            break;
        }
        /* Allocate delivery area */
        del_size = (GetDeliveryDataLen() + 3) / 4 *
        4;
        del_area = CoTaskMemAlloc(del_size);
        if (del_area == NULL) {
            userlog (LOG_ERR, "Failure allocate
            delivery memory(ThreadNo:%d,%dByte)\n",
            thread_number, del_size);
            GLB_ThreadCond = -2;
            break;
        }
        /* Check event */

```

```

        if (!SetEvent(GLB_DelConfEvent)) {
            userlog (LOG_ERR, "SetEvent/Initialize
            check(ThreadNo:%d)[GetLastError:%d]\n",
            thread_number,
            GetLastError());
            GLB_ThreadCond = -2;
            goto end_thread;
        }
#ifdef PUT_INF_LOG
        userlog (LOG_INF, "Delivery Thread initialize end
        [thread no. = %d]\n", (unsigned long)thread_no);
#endif
        /* Delivery execute transaction */
        for (;;) {
            /* Wait event */
            for (;;) {
                /* Process queue is gotten */
                get_que = GetProcDeliveryQue();
                if (get_que != 0) {
                    proc_count += 1;
                    break;
                }
                if
                (WaitForSingleObject(GLB_DelStrEvent, INFINITE) !=
                WAIT_OBJECT_0) {
                    userlog (LOG_ERR,
                    "WaitForSingleObject/Get process
                    queue(ThreadNo:%d)"
                    "[GetLastError:%d]\n", thread_number, GetLastError());
                    goto end_thread;
                }
                /* End thread check */
                if (GLB_ThreadEndReq == 1) {
                    goto end_thread;
                }
                memcpy(del_area, (void *)&get_que-
                >TxnData, del_size);
                /* Empty queue is connected */
                SetEmptyDeliveryQue(get_que);
                /* Delivery Execute */
#ifdef SCRTTEST
                return_value = 0;
            #else
                return_value = DeliveryExec(del_area,
                pISvrApl);
            #endif
            if (return_value != 0) {
                userlog (LOG_ERR, "failure execute
                delivery transaction(ThreadNo:%d)"
                "[Return_Value:%d]\n", thread_number, return_value);
            }
            /* Thread end transaction */
        }
        end_thread:
#ifdef PUT_INF_LOG
        userlog (LOG_INF, "Delivery Thread terminate
        start\n");
#endif
        /* End SvrAPL */
        DeleteSvrAplObject(pISvrApl);

```

```

        /* Free delivery area */
        if (del_area != 0)
            CoTaskMemFree(del_area);
        if (!SetEvent(GLB_DelConfEvent)) {
            userlog (LOG_ERR, "SetEvent/End
            Check(ThreadNo:%d)[GetLastError:%d]\n",
            thread_number,
            GetLastError());
        }
        userlog (LOG_INF, "Delivery Thread which it is
        going to end performed %d trasactions.\n",
        proc_count);
        _endthread();
        return;
    }
    /* Set process queue and request thread.
    * Argument
    * Top empty queue pointer
    * Return Value
    * 0 : Normal end
    * -1 : SetEvent error
    */
    int DeliveryRequest(DELIVERY_QUE* delivery_que) {
        MAC_PutFncEntryLog("DeliveryRequest");
        SetProcDeliveryQue(delivery_que);
        if (!SetEvent(GLB_DelStrEvent)) {
            userlog (LOG_ERR, "SetEvent/Set process
            queue[GetLastError:%d]\n", GetLastError());
            return -1;
        }
        return 0;
    }
    /* Delivery thread is terminated.
    * Return Value
    * None
    */
    void TermDelivery() {
        long thread_count;
        BOOL return_value;
        MAC_PutFncEntryLog("TermDelivery");
        /* Added T.Kato 03.08.04 Delivery queue overflow */
        userlog (LOG_INF, "Delivery queue overflow inf.
        [peek %d queues] [max retry %d ms]\n",
        GLB_DeliveryQueueOverPeek,
        GLB_DeliveryQueueWaitTimePeek);
        /* Added end */
        /* End is requested */
        GLB_ThreadEndReq = 1;
        /* All thread end process */
        for (thread_count = 0; thread_count <
        GLB_NumCreateThreads; thread_count++) {
            if (!SetEvent(GLB_DelStrEvent)) {
                userlog (LOG_ERR, "SetEvent/End
                execute[GetLastError:%d]\n", GetLastError());
            }
        }
        /* Replaced T.Kato 03.02.12 Too short timeout (10sec
        >3min) */
        /* if
        (WaitForSingleObject(GLB_DelConfEvent, 1000 *
        10) != WAIT_OBJECT_0) */
        if
        (WaitForSingleObject(GLB_DelConfEvent, 1000 * 60 *
        3) != WAIT_OBJECT_0) {
            /* Replaced end */

```

```

        userlog (LOG_ERR,
"WaitForSingleObject/End thread(TimeOut
3min)[GetLastError:%d]\n", GetLastError());
    }
}
GLB_NumCreateThreads = 0;
if (GLB_DelStrEvent != 0) {
    /* Close start event */
    return_value =
CloseHandle(GLB_DelStrEvent);
    GLB_DelStrEvent = 0;
    if (return_value == 0) {
        userlog (LOG_ERR, "Close execute
event[GetLastError:%d]\n", GetLastError());
    }
}
if (GLB_DelConfEvent != 0) {
    /* Close check event */
    return_value =
CloseHandle(GLB_DelConfEvent);
    GLB_DelConfEvent = 0;
    if (return_value == 0) {
        userlog (LOG_ERR, "Close check
event[GetLastError:%d]\n", GetLastError());
    }
}
/* Added T.kato 03.08.22 change up Delivery */
if (GLB_DelQueueWaitEvent != 0) {
    return_value =
CloseHandle(GLB_DelQueueWaitEvent);
    GLB_DelQueueWaitEvent = 0;
    if (return_value == 0) {
        userlog (LOG_ERR, "Close delivery
queue empty event[GetLastError:%d]\n",
GetLastError());
    }
}
/* Added end */
if (GLB_DelQueueArea != 0) {
    /* Free all queue area */
    free(GLB_DelQueueArea);
    GLB_DelQueueArea = 0;
}
return;
}
/* *****
* Set priority of delivery thread. *
* Return Value *
* None *
***** */
void SetDeliveryThreadPriority(HANDLE
thread_handle) {
    int thread_priority;
    int old_priority;
    /* Choose the priority of thread */
    switch(GLB_DeliveryPriority) {
    /* Default priority (No priority set) */
    case DEFAULT_DELIVERYPRIORITY:
        return;
    /* HIGHEST (+2) priority */
    case DELIVERYPRIORITY_1:
        thread_priority =
THREAD_PRIORITY_HIGHEST;
        break;
    /* ABOVE_NORMAL (+1) priority */
    case DELIVERYPRIORITY_2:
        thread_priority =
THREAD_PRIORITY_ABOVE_NORMAL;
        break;

```

```

    /* NORMAL (+0) priority */
    case DELIVERYPRIORITY_3:
        thread_priority =
THREAD_PRIORITY_NORMAL;
        break;
    /* BELOW_NORMAL (-1) priority */
    case DELIVERYPRIORITY_4:
        thread_priority =
THREAD_PRIORITY_BELOW_NORMAL;
        break;
    /* LOWEST (-2) priority */
    case DELIVERYPRIORITY_5:
        thread_priority =
THREAD_PRIORITY_LOWEST;
        break;
    /* Over range */
    default:
        userlog (LOG_ERR, "Derivery thread
priority over range [priority=%d]\n",
GLB_DeliveryPriority);
        return;
    }
    /* Set the priority */
    if ((old_priority =
GetThreadPriority(thread_handle)) ==
THREAD_PRIORITY_ERROR_RETURN) {
        userlog (LOG_ERR, "Error found at get old
thread priority (handle=%08x)\n", thread_handle);
        return;
    }
    if (!SetThreadPriority(thread_handle,
thread_priority)) {
        userlog (LOG_ERR, "Error found at set
new thread priority (handle=%08x)\n", thread_handle);
        return;
    }
    userlog (LOG_INF, "Set new thread priority to
derivery thread (handle=%08x old=%d new=%d)\n",
thread_handle,
old_priority, thread_priority);
    return;
}
.....
TpApIBase/InitThread.cpp
.....
/* *****
* TPC-C Client Application Program Source *
***** */
* Entry Functions *
* (1) GetRegistryInfo *
* (2) InitTerminalInfo *
* CREATE by TSL 2002.12.27 *
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
***** */
#include <windows.h>
#include <io.h>
#include "ObjectCntlInfo.h"
#include "TpcCntlInfo.h"
#include "Delivery.h"
#include "trans.h"
#include "ExecuteThread.h"
#include "GlobalArea.h"
#include "SvrApIBase.h"
#include "log.h"

```

```

#ifdef REG_USE
/*****
* Get registry control information. *
* Return Value *
* None *
***** */
void GetRegistryInfo() {
    DWORD lng_kind;
    DWORD lng_type;
    DWORD lng_size;
    int def_base = DEFAULT_TERMBASE;
    int def_warehouse =
DEFAULT_MAXWH;
    int def_maxusers =
DEFAULT_MAXCONNECT;
    int def_maxterm =
DEFAULT_MAXTERM;
    int del_flag = DEFAULT_CFLAG;
    int del_delthrd =
DEFAULT_NUMDELIVERYTHREADS;
    int del_delque =
DEFAULT_NUMDELIVERYQUEUEUS;
    union dtg {
        BYTE bit[32];
        char chara[32];
        DWORD data[8];
    } reg_d;
    /* Open registry Area */
    RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE", 0,
KEY_READ | KEY_WRITE,
&m_tpccregkey);
    /* Create registry key "Fujitsu" */
    RegCreateKeyEx(m_tpccregkey, "Fujitsu", 0,
NULL,
REG_OPTION_NON_VOLATILE,
KEY_READ | KEY_WRITE, NULL,
&m_tpccregkey, &lng_kind);
    /* Create registry key "TPC-C ISAPI Application"
*/
    RegCreateKeyEx(m_tpccregkey, "TPC-C ISAPI
Application", 0, NULL,
REG_OPTION_NON_VOLATILE,
KEY_READ | KEY_WRITE, NULL,
&m_tpccregkey, &lng_kind);
    /* If the key is newly made, the default value is
set in the key. */
    if(lng_kind == REG_CREATED_NEW_KEY) {
        RegSetValueEx(m_tpccregkey,
"Term_Base",
0, REG_DWORD,(const unsigned
char *)&def_base, 4);
        RegSetValueEx(m_tpccregkey,
"NumWarehouses",
0, REG_DWORD,(const unsigned
char *)&def_warehouse, 4);
        RegSetValueEx(m_tpccregkey,
"MaxUsers",
0, REG_DWORD,(const unsigned
char *)&def_maxusers, 4);
        RegSetValueEx(m_tpccregkey, "MaxTerm
of Client",
0, REG_DWORD,(const unsigned
char *)&def_maxterm, 4);
        RegSetValueEx(m_tpccregkey,
"CONTROL_Flag",
0, REG_DWORD,(const unsigned
char *)&del_flag, 4);
    }
}

```

```

        RegSetValueEx(m_tpccregkey,
"NumDeliveryThreads",
        0, REG_DWORD, (const unsigned
char *)&del_delthrd, 4);
        RegSetValueEx(m_tpccregkey,
"NumDeliveryQueus",
        0, REG_DWORD, (const unsigned
char *)&del_delque, 4);
    }
    /* Get registry data */
    RegQueryValueEx(m_tpccregkey, "Term_Base",
0, &lng_type, (unsigned char *)&reg_d.bit,
&lng_size);
    /* Start terminal(user) number */
    GLB_TermBase = reg_d.data[0];
    RegQueryValueEx(m_tpccregkey,
"NumWarehouses",
        0, &lng_type, (unsigned char *)&reg_d.bit,
&lng_size);
    /* Max warehouse scale */
    GLB_Numwh = reg_d.data[0];
    RegQueryValueEx(m_tpccregkey, "MaxUsers",
0, &lng_type, (unsigned char *)&reg_d.bit,
&lng_size);
    /* Max terminal(user) number */
    GLB_Maxconnect = reg_d.data[0];
    RegQueryValueEx(m_tpccregkey, "MaxTerm of
Client",
        0, &lng_type, (unsigned char *)&reg_d.bit,
&lng_size);
    /* Max terminal(user) number of client */
    GLB_Maxterm = reg_d.data[0];
    RegQueryValueEx(m_tpccregkey,
"CONTROL_Flag",
        0, &lng_type, (unsigned char *)&reg_d.bit,
&lng_size);
    /* debug flag */
    GLB_C_FLAG = reg_d.data[0];
    RegQueryValueEx(m_tpccregkey,
"NumDeliveryThreads",
        0, &lng_type, (unsigned char *)&reg_d.bit,
&lng_size);
    /* Delivery Thread number */
    GLB_NumDeliveryThreads = reg_d.data[0];
    RegQueryValueEx(m_tpccregkey,
"NumDeliveryQueus",
        0, &lng_type, (unsigned char *)&reg_d.bit,
&lng_size);
    /* Delivery Queue number */
    GLB_NumDeliveryQueus = reg_d.data[0];
    return;
}
#else
/*****
* Get INI file control information.
* Return Value
* None
*****/
char IniPath[MAX_PATH];
void GetIniFileInfo() {
#define INI_FILE_NAME "TpApl.ini"
int GetIniFileInfo_Get(char* key_name);
/* Make INI file path */
strcpy(IniPath, GLB_DefaultPath);
strcat(IniPath, INI_FILE_NAME);
/* Check INI file exist */
if (_access(IniPath, 0x00) != 0) {
/* INI file no exist, using default value */
#endifdef PUT_INF_LOG

```

```

        userlog(LOG_WRN, "INI file nothing, using
default value");
#endifif
        GLB_TermBase =
DEFAULT_TERMBASE;
        GLB_Numwh =
DEFAULT_MAXWH;
        GLB_Maxconnect =
DEFAULT_MAXCONNECT;
        GLB_Maxterm =
DEFAULT_MAXTERM;
        GLB_C_FLAG =
DEFAULT_CFLAG;
        GLB_NumDeliveryThreads =
DEFAULT_NUMDELIVERYTHREADS;
        GLB_NumDeliveryQueus =
DEFAULT_NUMDELIVERYQUEUS;
/* Added T.Kato 03.07.28 Speed up*/
        GLB_DeliveryPriority =
DEFAULT_DELIVERYPRIORITY;
/* Added end */
        return;
    }
    /* Get execution informations
*/
    /* If undefined key and illegal value, using default
value */
    if ((GLB_TermBase =
GetIniFileInfo_Get("Term_Base")) <= 0) {
        GLB_TermBase =
DEFAULT_TERMBASE;
    }
    if ((GLB_Numwh =
GetIniFileInfo_Get("NumWarehouses")) <= 0) {
        GLB_Numwh = DEFAULT_MAXWH;
    }
    if ((GLB_Maxconnect =
GetIniFileInfo_Get("MaxUsers")) <= 0) {
        GLB_Maxconnect =
DEFAULT_MAXCONNECT;
    }
    if ((GLB_Maxterm =
GetIniFileInfo_Get("MaxTerm of Client")) <= 0) {
        GLB_Maxterm =
DEFAULT_MAXTERM;
    }
    if ((GLB_C_FLAG =
GetIniFileInfo_Get("CONTROL_Flag")) < 0) {
        GLB_C_FLAG = DEFAULT_CFLAG;
    }
    if ((GLB_NumDeliveryThreads =
GetIniFileInfo_Get("NumDeliveryThreads")) <= 0) {
        GLB_NumDeliveryThreads =
DEFAULT_NUMDELIVERYTHREADS;
    }
    if ((GLB_NumDeliveryQueus =
GetIniFileInfo_Get("NumDeliveryQueus")) <= 0) {
        GLB_NumDeliveryQueus =
DEFAULT_NUMDELIVERYQUEUS;
    }
}
/* Added T.Kato 03.07.28 Speed up*/
        GLB_DeliveryPriority =
GetIniFileInfo_Get("DeliveryPriority");
        if (GLB_DeliveryPriority <
DEFAULT_DELIVERYPRIORITY ||
        GLB_DeliveryPriority >
DELIVERYPRIORITY_5) {
            GLB_DeliveryPriority =
DEFAULT_DELIVERYPRIORITY;

```

```

    }
    /* Added end */
}
/* Get information in the INI file */
int GetIniFileInfo_Get(char* key_name) {
    char value_buf[64];
    int i;
    for (i = 0; i < 3; i++) {
        GetPrivateProfileString("TPCC_INFO",
key_name, "",
        value_buf, sizeof(value_buf), IniPath);
        if (value_buf[0] == "") {
            /* if Key is nothing, retry getting */
            continue;
        }
        break;
    }
}
#ifdefdef PUT_INF_LOG
        userlog(LOG_INF, "INI file information [%s]=[%s]",
key_name, value_buf);
#endifif
        if (value_buf[0] == "") {
            /* Target key was nothing */
            return (-1);
        }
        return(atoi(value_buf));
    }
}
#endifif
/*****
* Initialize terminal control information.
* Return Value
* IO : Allocate error
* 0 : normal end
*****/
int InitTerminalInfo() {
    TC_CONTEXT* ctxt = 0;
    int term_count;
    MAC_PutFncEntryLog("InitTerminalInfo");
    /* Allocate terminal control information area */
    GLB_TcContext = (TC_CONTEXT
*)calloc(GLB_Maxterm, (unsigned int)(ctxt + 1));
    if (GLB_TcContext == NULL) {
        userlog (LOG_ERR, "Allocate
TPCCControlInformation area\n");
        return -1;
    }
    GLB_TerminalCntl.m_tctxt = GLB_TcContext;
    for (term_count = 0; term_count < GLB_Maxterm;
term_count++) {
        /* Set w_id, d_id */
        GLB_TerminalCntl.m_tctxt[term_count].w_id =
(GLB_TermBase + term_count - 1) / 10 + 1;
        GLB_TerminalCntl.m_tctxt[term_count].d_id =
(GLB_TermBase + term_count - 1) % 10 + 1;
        /* CriticalSection is initialized (Terminal
Control Information) */
        InitializeCriticalSection(&GLB_TerminalCntl.m_tc
ctxt[term_count].user);
    }
    return 0;
}
/*****
TpAplBase/ObjectCntlInfo.h
*****/

```

```

*
* TPC-C Client Application Program Source
*
* Entry Functions
* SvrAPL control information structure object.
* CREATE by TSL 2002.12.27
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002
*****/
#include "SvrAplBase.h"
typedef struct _object_cntl_info {
    _object_cntl_info* Before;
    _object_cntl_info* Next;
    void* TrxDData;
    ISvrApl* SvrAplObj;
    DWORD ThreadId;
    unsigned long NumThreadEx;
} OBJECT_CNTL_INFO;
*****/
#include "TpAplBase/ProcDelivery.cpp"
*****/
*
* TPC-C Client Application Program Source
*
* Entry Functions
* (1) GetEmptyDeliveryQue
* (2) SetEmptyDeliveryQue
* (3) GetProcDeliveryQue
* (4) SetProcDeliveryQue
*
* CREATE by TSL 2002.12.11
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002
*****/
#include <windows.h>
#include "ObjectCntlInfo.h"
#include "TpcCntlInfo.h"
#include "Delivery.h"
#include "trans.h"
#include "ExecuteThread.h"
#include "GlobalArea.h"
#include "log.h"
*****/
* Empty queue is gotten.
* Return Value
* Always empty queue pointer
*****/
DELIVERY_QUE* GetEmptyDeliveryQue() {
    DELIVERY_QUE* get_que;

/* Added T.Kato 03.08.04 Delivery queue overflow */
    int retry_count = 0;
    DWORD s_time = 0;
    DWORD e_time = 0;
/* Added end */
    for(;;) {

        EnterCriticalSection(&GLB_LockDeliveryQue);
        if (GLB_DelQueEmptyTop != 0) {
            break;
        }
        /* Nothing empty queue */
    }
#endif PUT_INF_LOG

```

```

userlog (LOG_WRN, "Nothing Empty
Queue.[GLB_DelQueEmptyTop:%d]\n",
GLB_DelQueEmptyTop);
#endif
/* Added T.Kato 03.08.04 Delivery queue overflow */
    if (retry_count == 0) {
        s_time = GetTickCount();
        GLB_DeliveryQueOver++;
        if (GLB_DeliveryQueOver >
GLB_DeliveryQueOverPeek) {
            GLB_DeliveryQueOverPeek =
GLB_DeliveryQueOver;
        }
        retry_count++;
/* Added end */

        LeaveCriticalSection(&GLB_LockDeliveryQue);
/* Replaced T.Kato 03.08.04 Change retry timing (Too
long sleeping??) */
        /* Sleep(1000);*/
/* Replaced T.kato 03.08.22 Delivery improve */
        /* Sleep(100);*/
        if
(WaitForSingleObject(GLB_DelQueWaitEvent,
INFINITE) != WAIT_OBJECT_0) {
            userlog (LOG_ERR,
"WaitForSingleObject/Waiting for que empty"
"[GetLastError:%d]\n", GetLastError());
            Sleep(60000);
        }
/* Replaced end */
/* Added T.Kato 03.08.04 Delivery queue overflow */
        if (retry_count != 0) {
            /* Compute waiting time */
            e_time = GetTickCount() - s_time;

            if (e_time > (unsigned
long)GLB_DeliveryQueWaitTime) {
                GLB_DeliveryQueWaitTime =
e_time;
            }
            if (GLB_DeliveryQueWaitTime >
GLB_DeliveryQueWaitTimePeek) {
                GLB_DeliveryQueWaitTimePeek =
GLB_DeliveryQueWaitTime;
            }
            GLB_DeliveryQueOver--;
        }
/* Added end */
        /* Move top empty queue */
        get_que = GLB_DelQueEmptyTop;
        GLB_DelQueEmptyTop =
GLB_DelQueEmptyTop->Next;
        if (GLB_DelQueEmptyTop == 0) {
            /* All empty queue is allocated */
            GLB_DelQueEmptyLast = 0;
        }
        LeaveCriticalSection(&GLB_LockDeliveryQue);
        return get_que;
    }
*****/
* Empty queue is connected.
* Argument
* Get process queue pointer

```

```

*****/
void SetEmptyDeliveryQue(DELIVERY_QUE*
new_que) {
    EnterCriticalSection(&GLB_LockDeliveryQue);
    if (GLB_DelQueEmptyLast == 0) {
        /* Nothing empty queue */
        GLB_DelQueEmptyTop = new_que;
    } else {
        GLB_DelQueEmptyLast->Next = new_que;
    }
    /* Move last empty queue */
    GLB_DelQueEmptyLast = new_que;
    new_que->Next = 0;
/* Added T.Kato 03.08.22 Change up Delivery */
    /* Post empty queue connected */
    if (!SetEvent(GLB_DelQueWaitEvent)) {
        userlog (LOG_ERR, "SetEvent/Empty
queue is connected [GetLastError:%d]\n",
GetLastError());
    }
/* Added end */
    LeaveCriticalSection(&GLB_LockDeliveryQue);
    return;
}
*****/
* Process queue is gotten.
* Return Value
* !0 : Process queue pointer
* 0 : Nothing process queue
*****/
DELIVERY_QUE* GetProcDeliveryQue() {
    DELIVERY_QUE* get_que;
    EnterCriticalSection(&GLB_LockDeliveryQue);
    if (GLB_DelQueProcTop == 0) {
        /* Nothing process queue */

        LeaveCriticalSection(&GLB_LockDeliveryQue);
        return 0;
    }
    /* Move top process queue */
    get_que = GLB_DelQueProcTop;
    GLB_DelQueProcTop = GLB_DelQueProcTop-
>Next;
    if (GLB_DelQueProcTop == 0) {
        /* Last process queue is lost */
        GLB_DelQueProcLast = 0;
    }
    LeaveCriticalSection(&GLB_LockDeliveryQue);
    return get_que;
}
*****/
* Process queue is connected.
* Argument
* Get empty queue pointer
*****/
void SetProcDeliveryQue(DELIVERY_QUE* new_que)
{
    EnterCriticalSection(&GLB_LockDeliveryQue);
    if (GLB_DelQueProcLast == 0) {
        /* Nothing process queue */
        GLB_DelQueProcTop = new_que;
    } else {
        GLB_DelQueProcLast->Next = new_que;
    }
    /* Move last process queue */
    GLB_DelQueProcLast = new_que;
    new_que->Next = 0;
    LeaveCriticalSection(&GLB_LockDeliveryQue);
    return;
}

```

```

}
.....
TpAplBase/SampleInfo.h
.....
/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* Sampling information. It's paformance data.
*
* CREATE by TSL 2003.08.24
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****/
typedef struct _sample_info {
    long sample_function;
    int obj_max, obj_min, obj_ttl,
obj_cnt;
    int new_max, new_min, new_ttl,
new_cnt;
    int pay_max, pay_min, pay_ttl,
pay_cnt;
    int odr_max, odr_min, odr_ttl,
odr_cnt;
    int del_max, del_min, del_ttl,
del_cnt;
    int sto_max, sto_min, sto_ttl,
sto_cnt;
} SAMPLE_INFO;
.....
TpAplBase/SvrAplObject.cpp
.....
/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* (1) CreateSvrAplObject
* (2) DeleteSvrAplObject
* (3) CreateSvrAplExecEnv
*
* CREATE by TSL 2002.12.27
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****/
#define _WIN32_WINNT 0x0400
#include <windows.h>
#define _WIN32_DCOM
#include <comdef.h>
#include "trans.h"
#include "ObjectCntllInfo.h"
#include "TpcCntllInfo.h"
#include "Delivery.h"
#include "SvrAplBase.h"
#include "ExecuteThread.h"
#include "GlobalArea.h"
#include "TpAplDBDependPrototype.h"
#include "log.h"
/*****
* Create Server APL object.
* Return Value
* Server APL object pointer
*****/
ISvrApl* CreateSvrAplObject(bool init) {

```

```

    MAC_PutFncEntryLog("CreateSvrAplObject");
#ifdef SCRTEST
    return (ISvrApl *)1; /* Return 1 because of
nomal return assumed */
#else
    HRESULT hr = 0;
    ISvrApl* pISvrApl;
    /* CoInitializeEx */
    if (init) {
        /* If true, initializing COM+ entry */
        hr =
CoInitializeEx(NULL, COINIT_MULTITHREADED);
        if (hr != S_OK) {
            userlog (LOG_ERR, "failure
CoInitializeEx[Return_Value:%08x(%d)]\n", hr, hr);
        }
    } else {
        /* If false, HttpExtensionProc already
initialized by IIS. */
        /* Then, noting to do. */
    }
    /* CoCreateInstance */
    hr = CoCreateInstance(CLSID_SvrApl, NULL,
CLSCTX_SERVER, IID_ISvrApl, (void*)&pISvrApl);
    if (hr != S_OK) {
        userlog (LOG_ERR, "failure
CoCreateInstance[Return_Value:%08x(%d)]\n", hr, hr);
        return 0;
    }
    /* EntryTrans */
    hr = pISvrApl->EntryTrans();
    if (hr != S_OK) {
        userlog (LOG_ERR, "Set
complete[Return_Value:%08x(%d)]\n", hr, hr);
        return 0;
    }
    /* Initialize SvrApl Executing Environment */
    hr = pISvrApl->InitSvrApl();
    if (hr != S_OK) {
        userlog (LOG_ERR, "Failure initialize
SvrApl Executing
Environment[Return_Value:%08x(%d)]\n", hr, hr);
        return 0;
    }
    return pISvrApl;
#endif
}
/*****
* Delete Server APL object.
* Argument
* Server APL object pointer
*****/
void DeleteSvrAplObject(ISvrApl* pISvrApl) {
    MAC_PutFncEntryLog("DeleteSvrAplObject");
#ifdef SCRTEST
    /* for warning */
    pISvrApl;
#else
    HRESULT hr = 0;
    if (pISvrApl != 0) {
        hr = pISvrApl->TermSvrApl();
        if (hr != S_OK) {
            userlog (LOG_ERR, "failure
TermSvrApl[Return_Value:%08x(%d)]\n", hr, hr);
            return;
        }
    }
}

```

```

    pISvrApl->Release();
}
CoUninitialize();
#endif
return;
}
/*****
* SvrAPL execution environment process is created.
*
* Return Value
* !0 : Object control information pointer
* 0 : Allocate error
*****/
OBJECT_CNTL_INFO* CreateSvrAplExecEnv() {
    OBJECT_CNTL_INFO* obj_pointer;
    int return_value;
    MAC_PutFncEntryLog("CreateSvrAplExecEnv");
    /* Get object control information pointer */
    obj_pointer =
(OBJECT_CNTL_INFO*)TlsGetValue(GLB_TlsIndex);
    if (obj_pointer == 0) {
        /* If the key data is not set */
        EnterCriticalSection(&GLB_LockSvrAplExecEnv);
        /* Allocate object control information area */
        obj_pointer = (OBJECT_CNTL_INFO
*)calloc(1, sizeof(OBJECT_CNTL_INFO));
        if (obj_pointer == NULL) {
            LeaveCriticalSection(&GLB_LockSvrAplExecEnv
);
            userlog (LOG_ERR, "Allocate
ObjectControlInformation area\n");
            return 0;
        }
        /* Create Object and set object control
information */
        obj_pointer->SvrAplObj =
CreateSvrAplObject(FALSE);
        if (obj_pointer->SvrAplObj == 0) {
            LeaveCriticalSection(&GLB_LockSvrAplExecEnv
);
            userlog (LOG_ERR, "Create SvrAPL
object\n");
            goto Alloc_Err;
        }
        /* Allocate transaction data area */
        obj_pointer->TrxData =
CoTaskMemAlloc(GetGenericDataLen());
        if (obj_pointer->TrxData == NULL) {
            LeaveCriticalSection(&GLB_LockSvrAplExecEnv
);
            userlog (LOG_ERR, "Allocate
transaction sata area\n");
            goto Alloc_Err;
        }
        /* Get thread ID */
        obj_pointer->ThreadId =
GetCurrentThreadId();
        /* Connect last queue */
        if (GLB_ObjMngLast == 0) {
            /* 1st thread */
            GLB_ObjMngTop = obj_pointer;
            obj_pointer->Before = 0;
        } else {
            GLB_ObjMngLast->Next =
obj_pointer;

```

```

        obj_pointer->Before =
GLB_ObjMngLast;
    }
    GLB_ObjMngLast = obj_pointer;
    obj_pointer->Next = 0;
    /* Set object control information pointer */
    return_value = TlsSetValue(GLB_TlsIndex,
(void *)obj_pointer);
    if (return_value == 0) {

        LeaveCriticalSection(&GLB_LockSvrAplExecEnv
);
        userlog (LOG_ERR, "Failure
TlsSetValue\n");
        goto Alloc_Err;
    }
    /* Free CriticalSection */

    LeaveCriticalSection(&GLB_LockSvrAplExecEnv
);
    }
    /* Thread execute number + 1 */
    obj_pointer->NumThreadEx = obj_pointer-
>NumThreadEx + 1;
    return obj_pointer;
Alloc_Err:
    if (obj_pointer->SvrAplObj != 0) {
        /* Free object and initialize COM+ */
        DeleteSvrAplObject(obj_pointer-
>SvrAplObj);
        obj_pointer->SvrAplObj = 0;
    }
    if ((int)obj_pointer->TrxDta != 0) {
        /* Free transaction data area */
        CoTaskMemFree(obj_pointer->TrxDta);
        obj_pointer->TrxDta = 0;
    }
    /* Free object control information */
    free((void *)obj_pointer);
    obj_pointer = 0;
    /* Free CriticalSection */
    LeaveCriticalSection(&GLB_LockSvrAplExecEnv
);
    return 0;
}
.....
TpAplBase/TermThread.cpp
.....
/*
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* (1) DetachExtensionThread
* (2) DeleteObjCntlInfo
* (3) TermSvrAplExecEnv
* (4) TermTerminalInfo
*
* CREATE by TSL 2002.12.27
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 */
.....
#include <windows.h>
#include <process.h>
#include "trans.h"
#include "ObjectCntlInfo.h"
#include "TpcCntlInfo.h"

```

```

#include "Delivery.h"
#include "ExecuteThread.h"
#include "GlobalArea.h"
#include "TpAplPrototype.h"
#include "log.h"
/*****
* Thread release processing.
* Return Value
* None
*****/
void DetachExtensionThread() {
    OBJECT_CNTL_INFO* obj_pointer;
    static int x = 0;
    MAC_PutFuncEntryLog("DetachExtensionThread"
);
    /* Get object control information pointer */
    obj_pointer =
(OBJECT_CNTL_INFO*)TlsGetValue(GLB_TlsIndex);
    if(obj_pointer == 0) return;
    /* Free object control information */
    DeleteObjCntlInfo(obj_pointer);
    /* Initialize object control information */
    TlsSetValue(GLB_TlsIndex, (void*)x);
    return;
}
/*****
* Delete object control information.
* Argument
* Object control information pointer
*****/
void DeleteObjCntlInfo(OBJECT_CNTL_INFO*
Object_Cntl_Info) {
    MAC_PutFuncEntryLog("DeleteObjCntlInfo");
    /* Thread process count is outputted to log file */
#ifdef PUT_INF_LOG
    userlog (LOG_INF, "Released SvrApl object
executes %d transactions, while activate.\n",
Object_Cntl_Info-
>NumThreadEx);
#endif
    Object_Cntl_Info->NumThreadEx = 0;
    if (Object_Cntl_Info->SvrAplObj != 0) {
        /* Free object and initialize COM+ */
        /* Comment out, because of invalid
interface pointer at running */
        DeleteSvrAplObject(Object_Cntl_Info-
>SvrAplObj);
        Object_Cntl_Info->SvrAplObj = 0;
    }
    if ((int)Object_Cntl_Info->TrxDta != 0) {
        /* Free transaction data area */
        CoTaskMemFree(Object_Cntl_Info-
>TrxDta);
        Object_Cntl_Info->TrxDta = 0;
    }
    /* Re-connect object control information */
    if (Object_Cntl_Info->Before != 0) {
        Object_Cntl_Info->Before->Next =
Object_Cntl_Info->Next;
    }
    if (Object_Cntl_Info->Next != 0) {
        Object_Cntl_Info->Next->Before =
Object_Cntl_Info->Before;
    }
    if (Object_Cntl_Info->Before == 0) {
        GLB_ObjMngTop = Object_Cntl_Info-
>Next;
    }
}

```

```

    if (Object_Cntl_Info->Next == 0) {
        GLB_ObjMngLast = Object_Cntl_Info-
>Before;
    }
    /* Free object control information */
    free((void *)Object_Cntl_Info);
    return;
}
/*****
* All object control informations are deleted.
* Return Value
* None
*****/
void TermSvrAplExecEnv() {
    while (GLB_ObjMngTop != 0) {
        DeleteObjCntlInfo(GLB_ObjMngTop);
    }
    return;
}
/*****
* Delete terminal control information.
* Return Value
* None
*****/
void TermTerminalInfo() {
    int term_count;
    MAC_PutFuncEntryLog("TermTerminalInfo");
    if (GLB_TcContext == 0) {
        return;
    }
    for (term_count = 0; term_count < GLB_Maxterm;
term_count++) {
        /* CriticalSection is deleted (Terminal
Control Information) */

        DeleteCriticalSection(&GLB_TerminalCntl.m_tcct
xt[term_count].user);
    }
    /* Free terminal control information */
    free((void *)GLB_TcContext);
    return;
}
.....
TpAplBase/TpAplDBDependPrototype.h
.....
/*
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Function definition for common.
*
* CREATE by TSL 2002.10.01
*
* 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);

```



```

<INPUT TYPE="submit" NAME="b" VALUE="Order
Status">\
<INPUT TYPE="submit" NAME="b" VALUE="Stock
Levell">\
<INPUT TYPE="submit" NAME="b"
VALUE="Quit">\
</FORM></BODY></HTML>"
/* Offset to field which should set data */
int newwp[] = {
0xa, 0x1c, 0x3c,
0x5c, 0x69, 0x84, 0x90,
0xa5, 0xc0, 0xd1, 0xe0,
0x138, 0x140, 0x148, 0x163, 0x169, 0x170, 0x175,
0x17e,
0x5f9, 0x62d
};
.....
TpAplBase/odrpge.h
.....
/*
*
* COPYRIGHT FUJITSU LIMITED 2002
* CREATE:1999.08.19 FJH
*
*****/
/* -----
-
odrpge.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\
$ . - - \r\n\
$ . - - \r\n\
$ . - - \r\n\
"
/* Trailer 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
Levell">\
<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 */
.....
TpAplBase/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"
/*
\r\n\
Credit: \r\n\
%%Disc: . \r\n\
*/
#define h_pay5 "\
\r\n\
Amount Paid: $ . New Cust-Balance:
$ . \r\n\
Credit Limit: $ . \r\n\
\r\n\
Cust-Data: "
/*
Cust-Data: \r\n\

```

```

\r\n\
\r\n\
\r\n\
*/
/* Trailer data */
#define h_pay3 "\
</PRE>\
<FORM ACTION="%s" METHOD="GET">\r\n\
<INPUT TYPE="hidden" NAME="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
Levell">\
<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);
.....
TpAplBase/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\

```



```

        userlog (LOG_INF, "===
TpAplBase.dll thread detach ===\n");
#endif
// Deleted 03.07.16 If thread detach occurs, all
execution information is deleted by this function.

//          DetachExtensionThread();
// Deleted end
#if 0 // KEEP-OBJ
        DetachExecuteThread();
#endif
        break;
        default:
        break;
    }
    return TRUE;
}
//
=====
//          GetExtensionVersion
//
=====
BOOL WINAPI GetExtensionVersion(OUT
HSE_VERSION_INFO * pVer)
{
    pVer->dwExtensionVersion =
MAKELONG( HSE_VERSION_MINOR,
HSE_VERSION_MAJOR);
    strcpy( pVer->lpszExtensionDesc,
        "IIS SDK Simple ISAPI Extension");
    return TRUE;
}
////////////////////////////////////////////////////////////////////
// HttpExtensionProc
////////////////////////////////////////////////////////////////////
// This method is called every time the user makes a
// request. We don't do anything here except catch a
successful
// return value and substitute a value that permits the
// network connection to be kept alive.
DWORD WINAPI HttpExtensionProc(IN
EXTENSION_CONTROL_BLOCK * pECB)
{
    /* Deleted 03.01.15 */
    #if 0
    !   char QueryString[1024];
    #endif
    /* Deleted end */
    char *page;
    int cookie = -1;
    /* Deleted 03.01.15 */
    #if 0
    !   int len;
    #endif
    /* Deleted end */
    int rtn;
    char S_BUF [BUF_S];
    char S_WORK[WORK_S];
    RTE_INPUT_DATA in_data_area;
    #ifdef DBPRT
    char dbg_query[1024];
    long s_pid;
    char path_w[96];
    #endif
    //for warning
    page;
    S_WORK;

```

```

#ifdef PUT_INF_LOG
    userlog (LOG_INF, "=== HttpExtensionProc
start===\n");
#endif
    if (GLB_InitCond != INTCOND_ERROR) {
        /* Create Delivery transaction enviroment */
        if (GLB_InitDelivery == 0) {
            /* If the key data is not set */

            EnterCriticalSection(&GLB_LockSvrAplExecEnv);
            if (GLB_InitDelivery == 0) {
                rtn = InitDelivery();
                if (rtn != 0) {
                    userlog (LOG_ERR,
"Error end InitDelivery() [Return_Value:%d]\n", rtn);
                    GLB_InitCond =
INTCOND_ERROR;
                }
                GLB_InitDelivery = 1;
                userlog (LOG_INF, "//
Normal End //");
                userlog (LOG_INF,
"////////////////////////");
            }
            /* Free CriticalSection */
            LeaveCriticalSection(&GLB_LockSvrAplExecEnv
);
        }
    }
    if (GLB_InitCond == INTCOND_ERROR) {
        userlog (LOG_ERR, "Can't Initialize\n");
        /* Initialization failure */
        sprintf (S_BUF, initer);
        OutputResultForm(pECB, S_BUF);
        return
HSE_STATUS_SUCCESS_AND_KEEP_CONN;
    }
    /* Replaced 03.01.15 */
    #if 0
    !   strcpy(QueryString, (char *)pECB-
>lpszQueryString);
    !
    #ifdef DBPRT
    !   strcpy(dbg_query, (char *)pECB-
>lpszQueryString);
    #endif
    !
    !   struct_init ( &in_data_area );
    !
    !   cookie = any_para ((char *)QueryString,
&in_data_area);
    #endif
    memset(&in_data_area, 0x00,
sizeof(in_data_area));
    cookie = any_para ((char *)pECB-
>lpszQueryString, &in_data_area );
    /* Replaced end */
    #ifdef DBPRT
    if ( cookie >= 0){
        sprintf (path_w, "c:\tlog\log%d", cookie);
        test_fp = fopen (path_w, "aw+");
    }
    #endif
    // Terminal Number Check
    // If terminal number is not valid then send error
message.
    if ( cookie < GLB_TermBase || cookie >=
(GLB_TermBase + GLB_Maxterm) ){

```

```

        //moved after client monitoring
        //userlog (LOG_ERR, "Terminal number
over the range [Terminal number:%d]\n", cookie);
        sprintf (S_BUF, badterm, GLB_TermBase,
GLB_TermBase + GLB_Maxterm - 1, cookie);
        /* Replaced 03.01.15 */
        #if 0
        !   len = strlen (S_BUF);
        !   isize = sprintf (szHeader, "200 OK");
        !   sprintf (szHeader1,
"Connection: keep-alive\r\nContent-
type: text/html\r\nContent-length: %d\r\n\r\n",
len);
        !
        !   rtncode = (*pECB-
>ServerSupportFunction)(pECB->ConnID,
HSE_REQ_SEND_RESPONSE_HEADER,
!
!   szHeader, (unsigned long *)&isize,
(LPDWORD)szHeader1);
        !
        !   rtncode = (*pECB->WriteClient)(pECB-
>ConnID, &S_BUF[0], (unsigned long *)&len, 0);
        #endif
        /* Added T.Kato 03.06.06 Client monitoring function */
        if (ClientMonitor(cookie, S_BUF) == 0) {
            if (cookie != -3)
                userlog (LOG_INF, "Extended
function executing [function number:%d]\n", cookie);
        }
        else {
            userlog (LOG_ERR, "Terminal
number over the range [Terminal number:%d]\n",
cookie);
        }
        /* Added end */
        OutputResultForm(pECB, S_BUF);
        /* Replaced end */
        #ifdef DBPRT
        if ( cookie >= 0){
            fprintf (test_fp, "--> QUERY: %s\n",
dbg_query);
            fprintf (test_fp, "%s\n", S_WORK,
S_BUF);
            fflush (test_fp);
            fclose (test_fp);
        }
        #endif
        return
HSE_STATUS_SUCCESS_AND_KEEP_CONN;
    }
    //
    rtn = select_trn ( &in_data_area, S_BUF,
cookie );
    //
    /* Replaced 03.01.15 */
    #if 0
    !   len = strlen (S_BUF);
    !
    !   isize = sprintf (szHeader, "200 OK");
    !
    !   sprintf (szHeader1,
"Connection: keep-alive\r\nContent-
type: text/html\r\nContent-length: %d\r\n\r\n",
len);
    !
    !   rtncode = (*pECB-
>ServerSupportFunction)(pECB->ConnID,
HSE_REQ_SEND_RESPONSE_HEADER,

```

```

!
    szHeader, (unsigned long *)(&isize),
    (LPDWORD)szHeader1);
!
!   rrcode = (*pECB->WriteClient)(pECB->ConnID,
&S_BUF[0], (unsigned long *)&len, 0);
#endif
    OutputResultForm(pECB, S_BUF);
/* Replaced end */

#ifndef DBPRT
    if (cookie >= 0){
        fprintf(test_fp, "--> QUERY: %s\n",
dbg_query);
        fprintf(test_fp, "%s\n", S_WORK,
S_BUF);
        fflush(test_fp);
        fclose(test_fp);
    }
#endif

// For Debug : output error message
if ((GLB_C_FLAG & PRT_ELOG) != 0) {
    if (strstr(S_BUF, "ERROR") != 0) {
        {
            FILE *test_fp2;
            char w_path[96];
            sprintf(w_path, "c:\\tcllog\\log%d", cookie);
            test_fp2 = fopen(w_path, "aw+");
            fprintf(test_fp2, "%s\n", S_BUF);
            fflush(test_fp2);
            fclose(test_fp2);
        }
    }
}
return
HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}
/*****
* Output Processing result form.
* Argument
* pECB :
* EXTENSION_CONTROL_BLOCK
structure object pointer
* buf_body :
* Output message on screen
*****/
void
OutputResultForm(EXTENSION_CONTROL_BLOCK *
pECB, char* buf_body) {
    char szHeader[128];
    char szHeader1[128];
    int len;
    int isize;
    BOOL rrcode;
    len = strlen(buf_body);
    isize = sprintf(szHeader, "200 OK");
    sprintf(szHeader1,
        "Connection: keep-alive\r\nContent-
type: text/html\r\nContent-length: %d\r\n\r\n",
len);
    rrcode = (*pECB-
>ServerSupportFunction)(pECB->ConnID,
HSE_REQ_SEND_RESPONSE_HEADER,

    szHeader, (unsigned long *)(&isize),
    (LPDWORD)szHeader1);

```

```

    rrcode = (*pECB->WriteClient)(pECB->ConnID,
&buf_body[0], (unsigned long *)&len, 0);
    return;
}
.....
TpAplBase/tpapl.def
.....
EXPORTS
    GetExtensionVersion
    HttpExtensionProc
.....
TpAplBase/tpapl.h
.....
/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* struct definition.
*
* CREATE by TSL 2002.10.01
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002
*****/
#define WIN32_LEAN_AND_MEAN
typedef struct tc_tbl {
    void *trans_b;
    int w_id;
    int d_id;
    char n_otname [8];
    char tm_id[5];
    CRITICAL_SECTION user;
} tc_context;
// Class --> struct
typedef struct _theExtension
{
    CRITICAL_SECTION crit;
    tc_context m_tcctx[MAXTERM];
}THE_EXTENSION;
.....
TpAplBase/tpapl.mak
.....
# Microsoft Developer Studio Generated NMAKE File,
Based on tpapl.dsp
!IF "$(CFG)" == ""
CFG=tpapl - Win32 Debug
!MESSAGE tpapl - Win32 Debug
!ENDIF
!IF "$(CFG)" != "tpapl - Win32 Release" && "$(CFG)" !=
"tpapl - Win32 Debug"
!MESSAGE "$(CFG)"
!MESSAGE NMAKE
!MESSAGE :
!MESSAGE
!MESSAGE NMAKE /f "tpapl.mak" CFG="tpapl - Win32
Debug"
!MESSAGE
!MESSAGE :
!MESSAGE
!MESSAGE "tpapl - Win32 Release" ("Win32 (x86)
Dynamic-Link Library")
!MESSAGE "tpapl - Win32 Debug" ("Win32 (x86)
Dynamic-Link Library")
!MESSAGE
!ERROR
!ENDIF
!IF "$(OS)" == "Windows_NT"

```

```

NULL=
!ELSE
NULL=nul
!ENDIF
!IF "$(CFG)" == "tpapl - Win32 Release"
OUTDIR=.\\Release
INTDIR=.\\Release
# Begin Custom Macros
OutDir=.\\Release
# End Custom Macros
ALL : "$(OUTDIR)\tpapl.dll"
CLEAN :
    -@erase "$(INTDIR)\ClientMonitor.obj"
    -@erase "$(INTDIR)\ConvInt.obj"
    -@erase "$(INTDIR)\ConvOther.obj"
    -@erase "$(INTDIR)\ConvString.obj"
    -@erase "$(INTDIR)\ErrPage.obj"
    -@erase "$(INTDIR)\GlobalArea.obj"
    -@erase "$(INTDIR)\InitDelivery.obj"
    -@erase "$(INTDIR)\InitThread.obj"
    -@erase "$(INTDIR)\ProcDelivery.obj"
    -@erase "$(INTDIR)\SvrAplBase_i.obj"
    -@erase "$(INTDIR)\SvrAplObject.obj"
    -@erase "$(INTDIR)\TermThread.obj"
    -@erase "$(INTDIR)\tpapl.obj"
    -@erase "$(INTDIR)\tpaplFunction.obj"
    -@erase "$(INTDIR)\vc60.idb"
    -@erase "$(OUTDIR)\tpapl.dll"
    -@erase "$(OUTDIR)\tpapl.exp"
    -@erase "$(OUTDIR)\tpapl.lib"
"$(OUTDIR)" :
    if not exist "$(OUTDIR)/$(NULL)" mkdir "$(OUTDIR)"
CPP=cl.exe
CPP_PROJ=/nologo /MT /W4 /GX /O2 /I
"..\\TPCCPublicFunc" /I "..\\SvrAplBase" /D "WIN32" /D
"NDEBUG" /D "_WINDOWS" /D "_MBCS" /D
"_USRDLL" /D "TPAPL_EXPORTS"
/Fp"$(INTDIR)\tpapl.pch" /YX /Fo"$(INTDIR)\\"
/Fd"$(INTDIR)\\" /FD /c
.c{$(INTDIR)}.obj:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cpp{$(INTDIR)}.obj:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cxx{$(INTDIR)}.obj:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.c{$(INTDIR)}.sbr:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cpp{$(INTDIR)}.sbr:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cxx{$(INTDIR)}.sbr:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
MTL=midl.exe
MTL_PROJ=/nologo /D "NDEBUG" /mktyplib203
/win32
RSC=rc.exe
BSC32=bscmake.exe
BSC32_FLAGS=/nologo /o"$(OUTDIR)\tpapl.bsc"

```

```

BSC32_SBRs= \

LINK32=link.exe
LINK32_FLAGS=ws2_32.lib TpDBDepend.lib
TPCCPublicFunc.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /dll /incremental:no /pdb:"$(OUTDIR)\tpapl.pdb"
/machine:i386 /def:". \tpapl.def"
/out:"$(OUTDIR)\tpapl.dll" /implib:"$(OUTDIR)\tpapl.lib"
/libpath:". \.. \TpDBDepend_oracle/Release"
/libpath:". \.. \TPCCPublicFunc/Release"
DEF_FILE= \
    ". \tpapl.def"
LINK32_OBJS= \
    "$(INTDIR)\ClientMonitor.obj" \
    "$(INTDIR)\ConvInt.obj" \
    "$(INTDIR)\ConvOther.obj" \
    "$(INTDIR)\ConvString.obj" \
    "$(INTDIR)\ErrPage.obj" \
    "$(INTDIR)\GlobalArea.obj" \
    "$(INTDIR)\InitDelivery.obj" \
    "$(INTDIR)\InitThread.obj" \
    "$(INTDIR)\ProcDelivery.obj" \
    "$(INTDIR)\SvrAplBase_i.obj" \
    "$(INTDIR)\SvrAplObject.obj" \
    "$(INTDIR)\TermThread.obj" \
    "$(INTDIR)\tpapl.obj" \
    "$(INTDIR)\tpaplFunction.obj"
"$(OUTDIR)\tpapl.dll" : "$(OUTDIR)" $(DEF_FILE)
$(LINK32_OBJS)
$(LINK32) @<<
$(LINK32_FLAGS) $(LINK32_OBJS)
<<
SOURCE="$(InputPath)"
PostBuild_Desc=BIN
DS_POSTBUILD_DEP=$(INTDIR)\postbld.dep
ALL : $(DS_POSTBUILD_DEP)
# Begin Custom Macros
OutDir=. \Release
# End Custom Macros
$(DS_POSTBUILD_DEP) : "$(OUTDIR)\tpapl.dll"
    copy . \Release\tpapl.dll .. \bin \Release\tpapl.dll
    echo Helper for Post-build step >
"$(DS_POSTBUILD_DEP)"
!ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
OUTDIR=. \Debug
INTDIR=. \Debug
# Begin Custom Macros
OutDir=. \Debug
# End Custom Macros

ALL : "$(OUTDIR)\tpapl.dll" "$(OUTDIR)\tpapl.bsc"
CLEAN :
    -@erase "$(INTDIR)\ClientMonitor.obj"
    -@erase "$(INTDIR)\ClientMonitor.sbr"
    -@erase "$(INTDIR)\ConvInt.obj"
    -@erase "$(INTDIR)\ConvInt.sbr"
    -@erase "$(INTDIR)\ConvOther.obj"
    -@erase "$(INTDIR)\ConvOther.sbr"
    -@erase "$(INTDIR)\ConvString.obj"
    -@erase "$(INTDIR)\ConvString.sbr"
    -@erase "$(INTDIR)\ErrPage.obj"
    -@erase "$(INTDIR)\ErrPage.sbr"
    -@erase "$(INTDIR)\GlobalArea.obj"
    -@erase "$(INTDIR)\GlobalArea.sbr"
    -@erase "$(INTDIR)\InitDelivery.obj"
    -@erase "$(INTDIR)\InitDelivery.sbr"
    -@erase "$(INTDIR)\InitThread.obj"

```

```

    -@erase "$(INTDIR)\InitThread.sbr"
    -@erase "$(INTDIR)\ProcDelivery.obj"
    -@erase "$(INTDIR)\ProcDelivery.sbr"
    -@erase "$(INTDIR)\SvrAplBase_i.obj"
    -@erase "$(INTDIR)\SvrAplBase_i.sbr"
    -@erase "$(INTDIR)\SvrAplObject.obj"
    -@erase "$(INTDIR)\SvrAplObject.sbr"
    -@erase "$(INTDIR)\TermThread.obj"
    -@erase "$(INTDIR)\TermThread.sbr"
    -@erase "$(INTDIR)\tpapl.obj"
    -@erase "$(INTDIR)\tpapl.sbr"
    -@erase "$(INTDIR)\tpaplFunction.obj"
    -@erase "$(INTDIR)\tpaplFunction.sbr"
    -@erase "$(INTDIR)\vc60.idb"
    -@erase "$(INTDIR)\vc60.pdb"
    -@erase "$(OUTDIR)\tpapl.bsc"
    -@erase "$(OUTDIR)\tpapl.dll"
    -@erase "$(OUTDIR)\tpapl.exp"
    -@erase "$(OUTDIR)\tpaplilk"
    -@erase "$(OUTDIR)\tpapl.lib"
    -@erase "$(OUTDIR)\tpapl.pdb"
"$(OUTDIR)" :
    if not exist "$(OUTDIR)\$(NULL)" mkdir "$(OUTDIR)"
CPP=cl.exe
CPP_PROJ=/nologo /MTd /W4 /Gm /GX /ZI /Od /I
    .. \TPCCPublicFunc /I .. \SvrAplBase /D "WIN32" /D
    "_DEBUG" /D "_WINDOWS" /D "_MBCS" /D
    "_USRDL" /D "TPAPL_EXPORTS" /FR"$(INTDIR)\
    /Fp"$(INTDIR)\tpapl.pch /YX /Fo"$(INTDIR)\
    /Fd"$(INTDIR)\ /FD /GZ /c
.c{$(INTDIR)}.obj:
    $(CPP) @<<
    $(CPP_PROJ) <
<<
.cpp{$(INTDIR)}.obj:
    $(CPP) @<<
    $(CPP_PROJ) <
<<
.cxx{$(INTDIR)}.obj:
    $(CPP) @<<
    $(CPP_PROJ) <
<<
.c{$(INTDIR)}.sbr:
    $(CPP) @<<
    $(CPP_PROJ) <
<<
.cpp{$(INTDIR)}.sbr:
    $(CPP) @<<
    $(CPP_PROJ) <
<<
.cxx{$(INTDIR)}.sbr:
    $(CPP) @<<
    $(CPP_PROJ) <
<<
MTL=midl.exe
MTL_PROJ=/nologo /D "_DEBUG" /mktyplib203 /win32
RSC=rc.exe
BSC32=bscmake.exe
BSC32_FLAGS=/nologo /o"$(OUTDIR)\tpapl.bsc"
BSC32_SBRs= \
    "$(INTDIR)\ClientMonitor.sbr" \
    "$(INTDIR)\ConvInt.sbr" \
    "$(INTDIR)\ConvOther.sbr" \
    "$(INTDIR)\ConvString.sbr" \
    "$(INTDIR)\ErrPage.sbr" \
    "$(INTDIR)\GlobalArea.sbr" \
    "$(INTDIR)\InitDelivery.sbr" \
    "$(INTDIR)\InitThread.sbr" \
    "$(INTDIR)\ProcDelivery.sbr" \

```

```

    "$(INTDIR)\SvrAplBase_i.sbr" \
    "$(INTDIR)\SvrAplObject.sbr" \
    "$(INTDIR)\TermThread.sbr" \
    "$(INTDIR)\tpapl.sbr" \
    "$(INTDIR)\tpaplFunction.sbr"
"$(OUTDIR)\tpapl.bsc" : "$(OUTDIR)" $(BSC32_SBRs)
$(BSC32) @<<
$(BSC32_FLAGS) $(BSC32_SBRs)
<<
LINK32=link.exe
LINK32_FLAGS=TpDBDepend.lib TPCCPublicFunc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /dll
/incremental:yes /pdb:"$(OUTDIR)\tpapl.pdb" /debug
/machine:i386 /def:". \tpapl.def"
/out:"$(OUTDIR)\tpapl.dll" /implib:"$(OUTDIR)\tpapl.lib"
/pdbtype:sept
/libpath:". \.. \TpDBDepend_oracle/Debug"
/libpath:". \.. \TPCCPublicFunc/Debug"
DEF_FILE= \
    ". \tpapl.def"
LINK32_OBJS= \
    "$(INTDIR)\ClientMonitor.obj" \
    "$(INTDIR)\ConvInt.obj" \
    "$(INTDIR)\ConvOther.obj" \
    "$(INTDIR)\ConvString.obj" \
    "$(INTDIR)\ErrPage.obj" \
    "$(INTDIR)\GlobalArea.obj" \
    "$(INTDIR)\InitDelivery.obj" \
    "$(INTDIR)\InitThread.obj" \
    "$(INTDIR)\ProcDelivery.obj" \
    "$(INTDIR)\SvrAplBase_i.obj" \
    "$(INTDIR)\SvrAplObject.obj" \
    "$(INTDIR)\TermThread.obj" \
    "$(INTDIR)\tpapl.obj" \
    "$(INTDIR)\tpaplFunction.obj"
"$(OUTDIR)\tpapl.dll" : "$(OUTDIR)" $(DEF_FILE)
$(LINK32_OBJS)
$(LINK32) @<<
$(LINK32_FLAGS) $(LINK32_OBJS)
<<
OURCE="$(InputPath)"
PostBuild_Desc=bin
DS_POSTBUILD_DEP=$(INTDIR)\postbld.dep
ALL : $(DS_POSTBUILD_DEP)
# Begin Custom Macros
OutDir=. \Debug
# End Custom Macros
$(DS_POSTBUILD_DEP) : "$(OUTDIR)\tpapl.dll"
"$(OUTDIR)\tpapl.bsc"
    rem copy . \Debug\tpapl.dll .. \bin \Debug\tpapl.dll
    echo Helper for Post-build step >
"$(DS_POSTBUILD_DEP)"
!ENDIF
!IF "$(NO_EXTERNAL_DEPS)" != "1"
!IF EXISTS("tpapl.dep")
!INCLUDE "tpapl.dep"
!ELSE
!MESSAGE Warning: cannot find "tpapl.dep"
!ENDIF
!ENDIF
!IF "$(CFG)" == "tpapl - Win32 Release" || "$(CFG)" ==
"tpapl - Win32 Debug"
SOURCE=. \ClientMonitor.cpp
!IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\ClientMonitor.obj" : $(SOURCE)
"$(INTDIR)"

```

```

ELSEIF "$(CFG)" == "tpapl - Win32 Debug"

"$(INTDIR)\ClientMonitor.obj"
    "$(INTDIR)\ClientMonitor.sbr" : $(SOURCE)
"$(INTDIR)"
ENDIF
SOURCE=.\ConvInt.cpp
IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\ConvInt.obj" : $(SOURCE) "$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\ConvInt.obj" "$(INTDIR)\ConvInt.sbr" :
$(SOURCE) "$(INTDIR)"
ENDIF
SOURCE=.\ConvOther.cpp
IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\ConvOther.obj" : $(SOURCE) "$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\ConvOther.obj"
    "$(INTDIR)\ConvOther.sbr" : $(SOURCE)
"$(INTDIR)"
ENDIF
SOURCE=.\ConvString.cpp
IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\ConvString.obj" : $(SOURCE) "$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\ConvString.obj"
    "$(INTDIR)\ConvString.sbr" : $(SOURCE)
"$(INTDIR)"
ENDIF
SOURCE=.\ErrPage.cpp
IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\ErrPage.obj" : $(SOURCE) "$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\ErrPage.obj"
    "$(INTDIR)\ErrPage.sbr" : $(SOURCE)
"$(INTDIR)"
ENDIF
SOURCE=.\GlobalArea.cpp
IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\GlobalArea.obj" : $(SOURCE) "$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\GlobalArea.obj"
    "$(INTDIR)\GlobalArea.sbr" : $(SOURCE)
"$(INTDIR)"
ENDIF
SOURCE=.\InitDelivery.cpp
IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\InitDelivery.obj" : $(SOURCE) "$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\InitDelivery.obj"
    "$(INTDIR)\InitDelivery.sbr" : $(SOURCE)
"$(INTDIR)"
ENDIF
SOURCE=.\InitThread.cpp
IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\InitThread.obj" : $(SOURCE) "$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\InitThread.obj"
    "$(INTDIR)\InitThread.sbr" : $(SOURCE)
"$(INTDIR)"
ENDIF
SOURCE=.\ProcDelivery.cpp
IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\ProcDelivery.obj" : $(SOURCE)
"$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\ProcDelivery.obj"
    "$(INTDIR)\ProcDelivery.sbr" : $(SOURCE)
"$(INTDIR)"
ENDIF
SOURCE=.\SvrAplBase\SvrAplBase_i.c
IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\SvrAplBase_i.obj" : $(SOURCE)
"$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\SvrAplBase_i.obj"
    "$(INTDIR)\SvrAplBase_i.sbr" : $(SOURCE)
"$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\SvrAplBase_i.sbr" : $(SOURCE)
"$(INTDIR)"
ENDIF
SOURCE=.\SvrAplObject.cpp
IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\SvrAplObject.obj" : $(SOURCE)
"$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\SvrAplObject.obj"
    "$(INTDIR)\SvrAplObject.sbr" : $(SOURCE)
"$(INTDIR)"
ENDIF
SOURCE=.\TermThread.cpp
IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\TermThread.obj" : $(SOURCE) "$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\TermThread.obj"
    "$(INTDIR)\TermThread.sbr" : $(SOURCE)
"$(INTDIR)"
ENDIF
SOURCE=.\tpapl.cpp
IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\tpapl.obj" : $(SOURCE) "$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\tpapl.obj" "$(INTDIR)\tpapl.sbr" :
$(SOURCE) "$(INTDIR)"
ENDIF
SOURCE=.\tpaplFunction.cpp
IF "$(CFG)" == "tpapl - Win32 Release"
"$(INTDIR)\tpaplFunction.obj" : $(SOURCE)
"$(INTDIR)"
ELSEIF "$(CFG)" == "tpapl - Win32 Debug"
"$(INTDIR)\tpaplFunction.obj"
    "$(INTDIR)\tpaplFunction.sbr" : $(SOURCE)
"$(INTDIR)"
ENDIF
ENDIF
TpAplBase\tpaplFunction.cpp
/*****
*
* 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 <windows.h>
#include <stdio.h>
#include "trans.h"
#include "tpccis.h"
#include "tpccinf.h"
#include "ObjectCntlInfo.h"
#include "TpcCntlInfo.h"
#include "Delivery.h"
#include "log.h"
// HTML-Page Data
#include "tpcweb.h"
#include "tpcinweb.h"
#include "menupage.h"
#include "tpapl.h"
#include "ExecuteThread.h"
#include "GlobalArea.h"
#include "TpAplPrototype.h"
#include "TpAplDBDependPrototype.h"
/*
anly_para :
QueryString

:
Gets the query string and finds every variable=value
pair contained
within it. For every pair, it runs the variable name
through a really
big compound switch statement that matches for
specific variables we
want to catch. When we find a known variable name,
we stick a pointer
to its corresponding value into the appropriate
member of 'ptrs.'
query - a 1024 byte buffer that contains the query
string.
ptrs - a raw_form_data structure to hold pointers.
*/
int anly_para(char *para, RTE_INPUT_DATA *in_data)
{
char *val, *rest;
if(!para) return 0;
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;
}
para = rest;
}
if (in_data->cookie != 0)
return(atoi (in_data->cookie));
else
return(0);
}
/* -----
select_trn:
RTE
:
s_bufHTML
interprets information from the user's input data to
determine which
page should be displayed back to the user.
query - the query string that comes back from
ParseFormData
ptrs - a pointer to a raw_form_data structure with
pointers
to values in 'query'.
----- */
int select_trn ( RTE_INPUT_DATA *in_data, char
*s_buf, int cookie ) {
int length = 0;
int user_id = cookie - GLB_TermBase;
int rtn = 0;
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{
EnterCriticalSection(&(srv-
>m_tcctxt[user_id].user)); // Make it thread safe.
/* check transaction type */
switch(in_data->form[0]) {
case 'N':
rtn = NewOrder (s_buf, in_data,
cookie);
break;
case 'D':
rtn = Delivery(s_buf, in_data, cookie);
break;
case 'P':
rtn = Payment (s_buf, in_data,
cookie);
break;
case 'S':
rtn = StockLevel(s_buf, in_data,
cookie);

```

```

break;
case 'O':
rtn = OrderStatus (s_buf,
in_data, cookie);
break;
default:
/* uninput transaction type */
set_errpage(s_buf, cookie, 1, -
4, 0, 0);
rtn = 1;
break;
}
LeaveCriticalSection(&(srv-
>m_tcctxt[user_id].user));
return rtn;
}
else if(in_data->button) {
EnterCriticalSection(&(srv-
>m_tcctxt[user_id].user)); // Make it thread safe.
/* 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);
strcpy(s_buf+length - 1 ,
in_newpage2);
break;
case 'D':
sprintf(s_buf, in_delpage, SOPATH, cookie, srv-
>m_tcctxt[user_id].w_id);
break;
case 'P':
sprintf(s_buf, in_paypage, SOPATH, cookie,
srv->m_tcctxt[user_id].w_id);
break;
case 'S':
sprintf(s_buf, in_stkpage, SOPATH, cookie,
srv-
>m_tcctxt[user_id].w_id, srv->m_tcctxt[user_id].d_id);
break;
case 'O':
sprintf(s_buf, in_odrpage, SOPATH, cookie, srv-
>m_tcctxt[user_id].w_id);
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;
}
LeaveCriticalSection(&(srv-
>m_tcctxt[user_id].user));
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
.....
TpAplBase/tpaplMain.h
.....
/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* Function definition for API.
*
* CREATE by TSL 2002.10.01
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
.....
BOOL APIENTRY DllMain( HANDLE hModule,
DWORD
ul_reason_for_call,
LPVOID lpReserved
);
BOOL WINAPI GetExtensionVersion(OUT
HSE_VERSION_INFO * pVer);
DWORD WINAPI HttpExtensionProc(IN
EXTENSION_CONTROL_BLOCK * pECB);
void
OutputResultForm(EXTENSION_CONTROL_B
OCK * pECB, char * buf_body);
.....
TpAplBase/tpccis.h
.....
/*****
*
* COPYRIGHT FUJITSU LIMITED 2002
* CREATE:1999.10.28 FJH
*
.....
//
// Client Application Header file
//
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <time.h>
#include <math.h>
#include <stdarg.h>
#include <signal.h>

```

```

#include <sys/stat.h>
#include <fcntl.h>
#include <malloc.h>
#include <process.h>
/* TPC-C transaction functions
extern int TPCinit ();
extern int TPCnew ();
extern int TPCpay ();
extern int TPCord ();
extern int TPCdel ();
extern int TPCsto ();
extern int TPCexit ();
extern int TPCdumpinit ();
extern int TPCdumpnew ();
extern int TPCdumppay ();
extern int TPCdumpord ();
extern int TPCdumpdel ();
extern int TPCdumpsto ();
extern int TPCdumpexit ();
*/
/* 99.8.19 w_id 5 keta *
#define MAXCONNECT 25000
#define MAXTERM 6400
#define MAXWH 2500 /* Max Warehouse
scale *
*/
#define MAXCONNECT 250000
#define MAXTERM 64000
#define MAXWH 25000 /* Max Warehouse
scale */
/* number of Transaction */
#define TRANNEW 1
#define TRANPAY 2
#define TRANORD 3
#define TRANDEL 4
#define TRANSTO 5
/* Error codes : send from TP application program */
#define RECOVERR -10
#define IRRECERR -20
#define NOERR 111
#define NG 0
#define OK 1
#define BUF_S 4096 /* size of the send buffer area
*/
#define WORK_S 2400 /* size of the work buffer area
*/
#define TERM_V 0x1 // Terminal Verification
#define PRT_ELOG 0x8 // Output Debug
Message
// SymfoWare Use Only
#define TX_NEWORDER 1
#define TX_PAYMENT 2
#define TX_ORDERSTAT 3
#define TX_DELIVERY 4
#define TX_STOCKLVL 5
/* Debug Print proc define : debug use only */
#ifdef DBPRT
#if (DBPRT > 5)
#define DBGP(proc) proc
#define DBGR(proc) proc
#else
#define DBGP(proc)
#define DBGR(proc) proc
#endif
#else
#define DBGP(proc)
#define DBGR(proc)
#endif
.....

```

```

TpAplBase/tpccinf.h
.....
/*****
*
* COPYRIGHT FUJITSU LIMITED 2002
* CREATE:1999.11.19 FJH
*
.....
/*=====+
=====+
FILENAME : tpccinf.h
DESCRIPTION
+=====+
=====+
/
#ifndef TPCCINF_H
#define TPCCINF_H
#define trans_size 1104 /* interfase area size */
#define VLDATA "Ver 1.0 COM+ (Process)"
#ifdef SCRTEST
#ifdef DBPRT
#define MDDATA "SCR And DP"
#else
#define MDDATA "SCR"
#endif
#endif
#else
#ifdef NOSCR
#define MDDATA "DBG"
#else
#define MDDATA "REL"
#endif
#endif
#endif
.....
TpAplBase/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=\"DI\">\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 "\

```



```

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"
.....
TpAplBase/tpcweb.h
.....
/* -----
*
* COPYRIGHT FUJITSU LIMITED 2002
* CREATE:1998.08.06 FJH
*
******/
/* -----
--
    tpcweb.h
-----
*/
/* If transaction input data is abnormal then use this
format. */
#define errhtml "\
<HTML><HEAD><TITLE>ERROR: TPC-
C</TITLE></HEAD><BODY>\

```

```

<p>You did something bad. The error message
was:</p>\
<PRE>%s</PRE>\
<p>Either hit the "back" button on your browser and
fix the problem, \
or hit the "Quit" button below to terminate this session.
</P><HR>\
<P><FORM ACTION="%s" METHOD="GET">\
<INPUT TYPE="hidden" NAME="c" VALUE=%d>\
<INPUT TYPE="submit" NAME="b"
VALUE="Quit">\
</FORM></P></BODY></HTML>\r\n"
/* If TP application terminated abnormally then use this
format. */
#define tuxerr "\
<HTML><HEAD><TITLE>ERROR: Tuxedo
</TITLE></HEAD><BODY>\
<P>The database could not process your request. \
tpcall terminated abnormally.</P>\
<HR><PRE>%s</PRE><HR>\
<FORM ACTION="%s" METHOD="GET">\
<INPUT TYPE="hidden" NAME="c" VALUE=%d>\
<INPUT TYPE="submit" NAME="b"
VALUE="Quit">\
</BODY></HTML>"
/* If application terminated abnormally then use this
format. */
#define errorpage "\
<HTML><HEAD><TITLE>ERROR: %s
</TITLE></HEAD><BODY>\
<P>The database could not process your request. \
Transaction terminated abnormally.</P>\
<HR><PRE>%s</PRE><HR>\
<FORM ACTION="%s" METHOD="GET">\
<INPUT TYPE="hidden" NAME="c" VALUE=%d>\
<INPUT TYPE="submit" NAME="b"
VALUE="Quit">\
</BODY></HTML>"
#if 0 /* oraerr,symfoerr --> errorpage */
/* [oraerr][symfoerr]"TITLE"*/
/* Since "TITLE" was only different, [oraerr] and
[symfoerr] were changed so that it might be common
and could use.*/
/* If Oracle application terminated abnormally then use
this format. */
#define oraerr "\
!<HTML><HEAD><TITLE>ERROR: ORACLE
</TITLE></HEAD><BODY>\
!<P>The database could not process your request. \
!Transaction terminated abnormally.</P>\
!<HR><PRE>%s</PRE><HR>\
!<FORM ACTION="%s" METHOD="GET">\
!<INPUT TYPE="hidden" NAME="c" VALUE=%d>\
!<INPUT TYPE="submit" NAME="b"
VALUE="Quit">\
!</BODY></HTML>"
!
/* If SymfoWare application terminated abnormally
then use this format. */
#define symfoerr "\
!<HTML><HEAD><TITLE>ERROR:
SYMFOWARE</TITLE></HEAD><BODY>\
!<P>The database could not process your request. \
!Transaction terminated abnormally.</P>\
!<HR><PRE>%s</PRE><HR>\
!<FORM ACTION="%s" METHOD="GET">\
!<INPUT TYPE="hidden" NAME="c" VALUE=%d>\
!<INPUT TYPE="submit" NAME="b"
VALUE="Quit">\

```

```

!</BODY></HTML>"
#endif
/* If TPINIT() abnormally then use this format. */
#define tuxerr "\
<HTML><HEAD><TITLE>ERROR: Tuxedo-init
</TITLE></HEAD><BODY>\
<P>The database could not process your request. \
%s terminated abnormally.</P>\
</BODY></HTML>"
.....
TpAplBase/trans.h
.....
/* -----
*
* COPYRIGHT FUJITSU LIMITED 2002
* CREATE:1999.10.28 FJH
*
******/
/* -----
=====
FILENAME : trans.h
the work struct according to transaction is declared.

=====
+=====
+=====*/
/* RTE - Client interface struct */
typedef struct {
    char *button,
    *cookie,
    *form,
    *O_CARRIER_ID,
    *threshold,
    *D_ID,
    *C_ID,
    *C_W_ID,
    *C_D_ID,
    *C_LAST,
    *H_AMOUNT,
    *OL_SUPPLY_W_ID[15],
    *OL_I_ID[15],
    *OL_QUANTITY[15];
} RTE_INPUT_DATA;
//} rte_input_data;
.....
TpDBDepend_Oracle/ConvTime.cpp
.....
/* -----
*
* 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 <windows.h>
#include <stdio.h>
#include "trans.h"
// Oracle Only -----
#include "tpcc_info.h"
// -----
#include "OracleFunction.h"
#include "ObjectCntlInfo.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
    sscanf (time, "%2d-%3s-%2d.%2d:%2d:%2d",
        &day, month, &year, &hour, &min, &sec);

    if (strcmp (month, "jan") == 0)
        strcpy (month, "01");
    if (strcmp (month, "feb") == 0)
        strcpy (month, "02");
    if (strcmp (month, "mar") == 0)
        strcpy (month, "03");
    if (strcmp (month, "apr") == 0)
        strcpy (month, "04");
    if (strcmp (month, "may") == 0)
        strcpy (month, "05");
    if (strcmp (month, "jun") == 0)
        strcpy (month, "06");
    if (strcmp (month, "jul") == 0)
        strcpy (month, "07");
    if (strcmp (month, "aug") == 0)
        strcpy (month, "08");
    if (strcmp (month, "sep") == 0)
        strcpy (month, "09");
    if (strcmp (month, "oct") == 0)
        strcpy (month, "10");
    if (strcmp (month, "nov") == 0)
        strcpy (month, "11");
    if (strcmp (month, "dec") == 0)
        strcpy (month, "12");

    int3str (str, 2, day);
    str[2] = ':';
    mon = atoi (month);
    int3str (&str[3], 2, mon);
    str[5] = ':';
    /* */
    if (year >= 70)
        year += 1900;
    else
        year += 2000;
    int3str (&str[6], 4, year);

    str[10] = ':';
    int3str (&str[11], 2, hour);
    str[13] = ':';
    int3str (&str[14], 2, min);
    str[16] = ':';
    int3str (&str[17], 2, sec);
}
.....
TpDBDepend_Oracle/CreateTranErrReason.cpp
.....
*
* TPC-C Client Application Program Source
*

```

```

* Entry Functions
* (1) CreateTranErrReason
*
* CREATE by TSL 2003.01.17
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002
.....
#include <windows.h>
#include "tpcc.h"
#include "SvrAplErrorCode.h"
int CreateTranErrReason(HRESULT ret_code, int
reason_code, char** reason_message) {
    switch (ret_code) {
        /* Normal end */
        case S_OK:
            return 0;
            break;
        /* Transaction execute error */
        case ERR_TRANS_PROC:
            if (reason_code != IRRECERR) {
                return 1; /* Retry execution */
            }
            *reason_message = "Irrecoverable error in
transaction processing.";
            return -1; /* Execution error */
            break;
        /* Initialize incomplete */
        case ERR_INIT_INCOMPLETE:
            *reason_message = "Initialize incomplete
for transaction executing.";
            return -1; /* Execution error */
            break;
        /* Transaction no execute */
        case ERR_NO_EXEC:
            *reason_message = "Transaction no
execute by reason of illegal environment.";
            return -1; /* Execution error */
            break;
        /* Process control error */
        case ERR_CNTL:
            *reason_message = "Internal controlling
error occurred.";
            return -1; /* Execution error */
            break;
        /* Other (return by COM+) */
        default:
            *reason_message = "Any error found by
COM+ controller.";
            return -1;
            break;
    }
}
.....
TpDBDepend_Oracle/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 (newstruct *bp, int cnt,
RTE_INPUT_DATA *in_data, int svcnt);
int set_NOdata (char *s_work, int OF, int cnt, newstruct
*bp, RTE_INPUT_DATA *in_data);
// -----
// TestProc.cpp
// -----
void dummy_delivery (delstruct *bp);
void dummy_stocklvl (stestruct *bp);
void dummy_payment (paystruct *bp);
void dummy_orderstat (ordstruct *bp);
void dummy_neworder (newstruct *bp);
void oder_dsp (RTE_INPUT_DATA *in_data, ordstruct
*bp, int w_id, int d_flag);
void pay_dsp (RTE_INPUT_DATA *in_data, paystruct
*bp, int w_id, int d_flag);
void sto_dsp (RTE_INPUT_DATA *in_data, stestruct
*bp, int w_id, int d_id, int d_flag);
void new_dsp (RTE_INPUT_DATA *in_data, newstruct
*bp, int w_id, int d_flag, int cnt);
int CreateTranErrReason (HRESULT ret_code, int
reason_code, char** reason_message);
// OracleSymfo
// used in common by Oracle and Symfo.
#define MAC_errHTML (page, err_inf, cookie)
set_errHTML (page, err_inf, cookie, "ORACLE");
.....
TpDBDepend_Oracle/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
*
.....
#ifdef ORACLEINFO_H
#define ORACLEINFO_H
#define INTNULL 0
#endif
.....
TpDBDepend_Oracle/TestFunction.cpp
.....
.....
*
* 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 <windows.h>
#include "trans.h"
// Oracle Only -----
#include "tpcc_info.h"
// -----
#include "tpccis.h"
#include "OracleFunction.h"
//
// dmy.h : DLL
//
// SCRTEST#define
// DBPRT#define
//
//
//
//
#define 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( delstruct *bp)
{
    bp->delout.terror = NOERR;
    return;
}
void dummy_stocklvl( stostruct *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( paystruct *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( ordstruct *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( newstruct *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
//
//
//
#define DBPRT
void oder_dsp(RTE_INPUT_DATA *in_data,
    ordstruct *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_balanc=%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,
            paystruct *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_balanc=%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,
            stostruct *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,
            newstruct *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_credi=%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
.....
TpDBDepend_Oracle/TpDBDepend.mak
.....

```

```
# Microsoft Developer Studio Generated NMAKE File,
Based on TpDBDepend.dsp
!IF "$(CFG)" == ""
CFG=TpDBDepend - Win32 Debug
!MESSAGE TpDBDepend - Win32 Debug
!ENDIF
!IF "$(CFG)" != "TpDBDepend - Win32 Release" &&
"$(CFG)" != "TpDBDepend - Win32 Debug"
!MESSAGE "$(CFG)"
!MESSAGE NMAKE
!MESSAGE :
!MESSAGE
!MESSAGE NMAKE /f "TpDBDepend.mak"
CFG="TpDBDepend - Win32 Debug"
!MESSAGE
!MESSAGE :
!MESSAGE :
!MESSAGE "TpDBDepend - Win32 Release" ("Win32
(x86) Static Library" )
!MESSAGE "TpDBDepend - Win32 Debug" ("Win32
(x86) Static Library" )
!MESSAGE
!ERROR
!ENDIF
!IF "$(OS)" == "Windows_NT"
NULL=
!ELSE
NULL=nul
!ENDIF
!IF "$(CFG)" == "TpDBDepend - Win32 Release"
OUTDIR=.Release
INTDIR=.Release
# Begin Custom Macros
OutDir=.Release
# End Custom Macros
ALL : "$(OUTDIR)\TpDBDepend.lib"
"$(OUTDIR)\TpDBDepend.bsc"
CLEAN :
-@erase "$(INTDIR)\ConvTime.obj"
-@erase "$(INTDIR)\ConvTime.sbr"
-@erase "$(INTDIR)\CreateTranErrReason.obj"
-@erase "$(INTDIR)\CreateTranErrReason.sbr"
-@erase "$(INTDIR)\TestFunction.obj"
-@erase "$(INTDIR)\TestFunction.sbr"
-@erase "$(INTDIR)\TransactionDataLen.obj"
-@erase "$(INTDIR)\TransactionDataLen.sbr"
-@erase "$(INTDIR)\TrxDelivery.obj"
-@erase "$(INTDIR)\TrxDelivery.sbr"
-@erase "$(INTDIR)\TrxNewOrder.obj"
-@erase "$(INTDIR)\TrxNewOrder.sbr"
-@erase "$(INTDIR)\TrxOrderStatus.obj"
-@erase "$(INTDIR)\TrxOrderStatus.sbr"
-@erase "$(INTDIR)\TrxPayment.obj"
-@erase "$(INTDIR)\TrxPayment.sbr"
-@erase "$(INTDIR)\TrxStockLevel.obj"
-@erase "$(INTDIR)\TrxStockLevel.sbr"
-@erase "$(INTDIR)\vc60.idb"
-@erase "$(OUTDIR)\TpDBDepend.bsc"
-@erase "$(OUTDIR)\TpDBDepend.lib"
"$(OUTDIR)" :
if not exist "$(OUTDIR)\$(NULL)" mkdir "$(OUTDIR)"
CPP=cl.exe
CPP_PROJ=/nologo /MT /W3 /GX /O2 /I "..\TpAplBase"
/I "..\TPCCPublicFunc" /I "..\SvrAplBase" /I
"..\SvrDBDepend_Oracle\10g\THREAD_SvrDBDepend
_3tier" /I
"..\SvrDBDepend_Oracle\10g\oracle\oci\include" /D
"WIN32" /D "NDEBUG" /D "_MBCS" /D "_LIB"

```

```
/FR"$(INTDIR)\\" /Fp"$(INTDIR)\TpDBDepend.pch" /YX
/Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /c
.c{$(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cpp{$(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cxx{$(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.c{$(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cpp{$(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cxx{$(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
RSC=rc.exe
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/o"$(OUTDIR)\TpDBDepend.bsc"
BSC32_SBRS= \
"$(INTDIR)\ConvTime.sbr" \
"$(INTDIR)\CreateTranErrReason.sbr" \
"$(INTDIR)\TestFunction.sbr" \
"$(INTDIR)\TransactionDataLen.sbr" \
"$(INTDIR)\TrxDelivery.sbr" \
"$(INTDIR)\TrxNewOrder.sbr" \
"$(INTDIR)\TrxOrderStatus.sbr" \
"$(INTDIR)\TrxPayment.sbr" \
"$(INTDIR)\TrxStockLevel.sbr"
"$(OUTDIR)\TpDBDepend.bsc" : "$(OUTDIR)"
$(BSC32_SBRS)
$(BSC32) @<<
$(BSC32_FLAGS) $(BSC32_SBRS)
<<
LIB32=link.exe -lib
LIB32_FLAGS=/nologo
/out:"$(OUTDIR)\TpDBDepend.lib"
LIB32_OBJS= \
"$(INTDIR)\ConvTime.obj" \
"$(INTDIR)\CreateTranErrReason.obj" \
"$(INTDIR)\TestFunction.obj" \
"$(INTDIR)\TransactionDataLen.obj" \
"$(INTDIR)\TrxDelivery.obj" \
"$(INTDIR)\TrxNewOrder.obj" \
"$(INTDIR)\TrxOrderStatus.obj" \
"$(INTDIR)\TrxPayment.obj" \
"$(INTDIR)\TrxStockLevel.obj"
"$(OUTDIR)\TpDBDepend.lib" : "$(OUTDIR)"
$(DEF_FILE) $(LIB32_OBJS)
$(LIB32) @<<
$(LIB32_FLAGS) $(DEF_FLAGS) $(LIB32_OBJS)
<<
!ELSEIF "$(CFG)" == "TpDBDepend - Win32 Debug"
OUTDIR=.Debug
INTDIR=.Debug
# Begin Custom Macros
OutDir=.Debug
# End Custom Macros

```

```
ALL : "$(OUTDIR)\TpDBDepend.lib"
"$(OUTDIR)\TpDBDepend.bsc"
CLEAN :
-@erase "$(INTDIR)\ConvTime.obj"
-@erase "$(INTDIR)\ConvTime.sbr"
-@erase "$(INTDIR)\CreateTranErrReason.obj"
-@erase "$(INTDIR)\CreateTranErrReason.sbr"
-@erase "$(INTDIR)\TestFunction.obj"
-@erase "$(INTDIR)\TestFunction.sbr"
-@erase "$(INTDIR)\TransactionDataLen.obj"
-@erase "$(INTDIR)\TransactionDataLen.sbr"
-@erase "$(INTDIR)\TrxDelivery.obj"
-@erase "$(INTDIR)\TrxDelivery.sbr"
-@erase "$(INTDIR)\TrxNewOrder.obj"
-@erase "$(INTDIR)\TrxNewOrder.sbr"
-@erase "$(INTDIR)\TrxOrderStatus.obj"
-@erase "$(INTDIR)\TrxOrderStatus.sbr"
-@erase "$(INTDIR)\TrxPayment.obj"
-@erase "$(INTDIR)\TrxPayment.sbr"
-@erase "$(INTDIR)\TrxStockLevel.obj"
-@erase "$(INTDIR)\TrxStockLevel.sbr"
-@erase "$(INTDIR)\vc60.idb"
-@erase "$(INTDIR)\vc60.pdb"
-@erase "$(OUTDIR)\TpDBDepend.bsc"
-@erase "$(OUTDIR)\TpDBDepend.lib"
"$(OUTDIR)" :
if not exist "$(OUTDIR)\$(NULL)" mkdir "$(OUTDIR)"
CPP=cl.exe
CPP_PROJ=/nologo /MTd /W3 /Gm /GX /ZI /Od /X /I
"..\TpAplBase" /I "..\TPCCPublicFunc" /I "..\SvrAplBase"
/I
"..\SvrDBDepend_Oracle\10i\THREAD_SvrDBDepend_
3tier" /I
"..\SvrDBDepend_Oracle\10i\oracle\oci\include"
/D "WIN32" /D "_DEBUG" /D "_MBCS" /D "_LIB"
/FR"$(INTDIR)\\" /Fp"$(INTDIR)\TpDBDepend.pch" /YX
/Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /GZ /c
.c{$(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cpp{$(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cxx{$(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.c{$(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cpp{$(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cxx{$(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
RSC=rc.exe
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/o"$(OUTDIR)\TpDBDepend.bsc"
BSC32_SBRS= \
"$(INTDIR)\ConvTime.sbr" \
"$(INTDIR)\CreateTranErrReason.sbr" \
"$(INTDIR)\TestFunction.sbr" \
"$(INTDIR)\TransactionDataLen.sbr" \

```

```

"$(INTDIR)\TrxDelivery.sbr" \
"$(INTDIR)\TrxNewOrder.sbr" \
"$(INTDIR)\TrxOrderStatus.sbr" \
"$(INTDIR)\TrxPayment.sbr" \
"$(INTDIR)\TrxStockLevel.sbr" \
$(OUTDIR)\TpDBDepend.bsc" : "$(OUTDIR)"
$(BSC32_SBRS)
$(BSC32) @<<
$(BSC32_FLAGS) $(BSC32_SBRS)
<<
LIB32=link.exe -lib
LIB32_FLAGS=/nologo
/out:"$(OUTDIR)\TpDBDepend.lib"
LIB32_OBJS= \
"$(INTDIR)\ConvTime.obj" \
"$(INTDIR)\CreateTranErrReason.obj" \
"$(INTDIR)\TestFunction.obj" \
"$(INTDIR)\TransactionDataLen.obj" \
"$(INTDIR)\TrxDelivery.obj" \
"$(INTDIR)\TrxNewOrder.obj" \
"$(INTDIR)\TrxOrderStatus.obj" \
"$(INTDIR)\TrxPayment.obj" \
"$(INTDIR)\TrxStockLevel.obj"
$(OUTDIR)\TpDBDepend.lib" : "$(OUTDIR)"
$(DEF_FILE) $(LIB32_OBJS)
$(LIB32) @<<
$(LIB32_FLAGS) $(DEF_FLAGS) $(LIB32_OBJS)
<<
!ENDIF
!IF "$(NO_EXTERNAL_DEPS)" != "1"
!IF EXISTS("TpDBDepend.dep")
!INCLUDE "TpDBDepend.dep"
!ELSE
!MESSAGE Warning: cannot find "TpDBDepend.dep"
!ENDIF
!ENDIF
!IF "$(CFG)" == "TpDBDepend - Win32 Release" ||
"$(CFG)" == "TpDBDepend - Win32 Debug"
SOURCE=. \ConvTime.cpp
"$(INTDIR)\ConvTime.obj"
"$(INTDIR)\ConvTime.sbr" : $(SOURCE)
"$(INTDIR)"
SOURCE=. \CreateTranErrReason.cpp
"$(INTDIR)\CreateTranErrReason.obj"
"$(INTDIR)\CreateTranErrReason.sbr" :
$(SOURCE) "$(INTDIR)"
SOURCE=. \TestFunction.cpp
"$(INTDIR)\TestFunction.obj"
"$(INTDIR)\TestFunction.sbr" : $(SOURCE)
"$(INTDIR)"
SOURCE=. \TransactionDataLen.cpp
"$(INTDIR)\TransactionDataLen.obj"
"$(INTDIR)\TransactionDataLen.sbr" :
$(SOURCE) "$(INTDIR)"
SOURCE=. \TrxDelivery.cpp
"$(INTDIR)\TrxDelivery.obj"
"$(INTDIR)\TrxDelivery.sbr" : $(SOURCE)
"$(INTDIR)"
SOURCE=. \TrxNewOrder.cpp
"$(INTDIR)\TrxNewOrder.obj"
"$(INTDIR)\TrxNewOrder.sbr" : $(SOURCE)
"$(INTDIR)"
SOURCE=. \TrxOrderStatus.cpp
"$(INTDIR)\TrxOrderStatus.obj"
"$(INTDIR)\TrxOrderStatus.sbr" : $(SOURCE)
"$(INTDIR)"
SOURCE=. \TrxPayment.cpp

```

```

"$(INTDIR)\TrxPayment.obj"
"$(INTDIR)\TrxPayment.sbr" : $(SOURCE)
"$(INTDIR)"
SOURCE=. \TrxStockLevel.cpp
"$(INTDIR)\TrxStockLevel.obj"
"$(INTDIR)\TrxStockLevel.sbr" : $(SOURCE)
"$(INTDIR)"
!ENDIF
.....
TpDBDepend_Oracle/TransactionDataLen.cpp
.....
/*****
*
* 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 <windows.h>
#include "tpcc_info.h"
/*****
* Get transaction data size.
* Return Value
* transaction data size
.....
long GetGenericDataLen() {
    long max_len = 0;
    if (max_len < sizeof(struct newstruct)) max_len =
sizeof(struct newstruct);
    if (max_len < sizeof(struct ordstruct)) max_len =
sizeof(struct ordstruct);
    if (max_len < sizeof(struct paystruct)) max_len =
sizeof(struct paystruct);
    if (max_len < sizeof(struct stostruct)) max_len =
sizeof(struct stostruct);
    return max_len;
}
/*****
* Get delivery transaction data size.
* Return Value
* Delivery transaction data size
.....
long GetDeliveryDataLen() {
    return sizeof(struct delstruct);
}
.....
TpDBDepend_Oracle/TrxDelivery.cpp
.....
/*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* (1) Delivery
* (2) DeliveryExec
*
* CREATE by TSL 2002.10.01
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
.....

```

```

#include <windows.h>
#include "trans.h"
// Oracle Only -----
#include "tpcc_info.h"
// -----
#include "tpccis.h"
#include "fdtbl.h"
#include "delpage.h"

#include "ObjectCntlInfo.h"
#include "TpcCntlInfo.h"
#include "Delivery.h"
#include "ExecuteThread.h"
#include "TpApiPrototype.h"
#include "TpApiDBDependPrototype.h"
#include "log.h"
#include "tpapl.h"
#include "GlobalArea.h" // Common
#include "OracleFunction.h"
/*
Delivery : this function processes the delivery
transaction.
*/
int Delivery (char *s_buf, RTE_INPUT_DATA *in_data,
int cookie)
{
    delstruct *bp;
    int user_id;
    char S_WORK[WORK_S];
    struct tm times;
    SYSTEMTIME systemTime;
    DELIVERY_QUE* del_que;
    int ret_val;
    MAC_PutFncEntryLog("Delivery");
    user_id = cookie - GLB_TermBase;
    /* Get empty delivery queue */
    del_que = GetEmptyDeliveryQue();
    bp = (delstruct *)del_que->TrxDData;
    memset(bp, 0x00, sizeof(delstruct));
    // bp->retval = 0;
    // bp->delout.error = 0;
    /* ----- Check the Input
data */
    /* Replaced T.Kato 03.08.14 w-id is short range */
    /* bp->delin.w_id = (short)srv->m_tcctxt[user_id].w_id;*/
    bp->delin.w_id = srv->m_tcctxt[user_id].w_id;
    /* Replaced end */
    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) {
        userlog (LOG_ERR, "Input data error [DEL]
(o_carrier_id = %s)[Return_Value:%d]\n",
in_data-
>O_CARRIER_ID, bp->delin.o_carrier_id);
        /* Empty queue is connected */
        SetEmptyDeliveryQue(del_que);
        return set_errpage(s_buf, cookie, 5, (int)bp-
>delin.o_carrier_id, 0, 0);
    }
}
/* ----- Execute Delivery
transaction */
GetLocalTime(&systemTime);
// GetLocalTime, mktime
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;
bp->delin.startsec = (long)mkttime (&times);
bp->delin.startusec =
(long)systemTime.wMilliseconds;

DBGR(fpprintf ( test_fp, "tpacall delivery!\n" ));
DBGR(tsp(0));
ret_val = DeliveryRequest(del_que);
if (ret_val != 0) {
    sprintf(S_WORK, "Can not Request Delivery
[DEL] (%d) \n", ret_val);
    MAC_errHTML(s_buf, S_WORK, cookie );
    userlog (LOG_ERR, S_WORK);
    return (-1);
}
DBGR(tsp(1));
/* ----- 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);
int 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
int h_del1_leng;
int h_del3_leng;
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 */
return 0;
}
/*-----*/
* Execute delivery transaction *
* Argument *
* Object control information pointer *
* Return Value *
* !0 : Oracle failed to process Delivery Transaction
*
* 0 : normal end *
/*-----*/
int DeliveryExec(void* delivery_tnx_area, ISvrApl*
pISvrApl) {
    delstruct *bp;
    char S_WORK[WORK_S];
    HRESULT ret_value;
    char* tran_errmsg;
    int ret_val;

```

```

MAC_PutFncEntryLog("DeliveryExec");
bp = ( delstruct *)delivery_tnx_area;
if ((ret_value = pISvrApl->ExDelivery((int
*)&bp)) != S_OK) {
    ret_val = CreateTranErrReason(ret_value,
bp->delout.terror, &tran_errmsg);
    sprintf(S_WORK, "Oracle failed to process
Delivery Transaction.(%s)\n"
"ret_value = %d w_id = %d
o_carrier_id = %d\n",
    tran_errmsg, ret_value,
bp->delin.w_id, bp-
>delin.o_carrier_id);
    userlog (LOG_ERR, S_WORK);
    return (-1);
}
return 0;
}
/*-----*/
TpDBDepend_Oracle/TrxNewOrder.cpp
/*-----*/
/*-----*/
* 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 <windows.h>
#include "trans.h"
// Oracle Only -----
#include "tpcc_info.h"
// -----
#include "tpccis.h"
#include "fldtbl.h"
#include "newpage.h"
#include "ObjectCntlInfo.h"
#include "TpcCntlInfo.h"
#include "Delivery.h"
#include "ExecuteThread.h"
#include "TpApiDBDependPrototype.h"
#include "log.h"
#include "tpapl.h"
#include "GlobalArea.h" // Common
#include "OracleFunction.h"
/*-----*/
NewOrder : this function processes the NewOrder
transaction.
/*-----*/
int NewOrder (char *s_buf, RTE_INPUT_DATA
*in_data, int cookie)
{
    newstruct *bp;
    int user_id, i;
    int ol_cnt, cnt, rtn;
    char S_WORK[WORK_S];
    char time_data[64];
    double entry;
    //SvrAPL return value

```

```

#ifndef SCRTEST
    HRESULT ret_value;
#endif
int
    OBJECT_CNTL_INFO* ObjCntlInfo;
    int return_value;
#ifndef SCRTEST
    char* tran_errmsg;
    int ret_val;
#endif
//for warning
time_data;
entry;
MAC_PutFncEntryLog("NewOrder");
user_id = cookie - GLB_TermBase;
/* Create execution environment */
ObjCntlInfo = CreateSvrAplExecEnv();
if (ObjCntlInfo == 0) {
    sprintf(S_WORK, "Execution environment was not
created [NEW]\n");
    MAC_errHTML(s_buf, S_WORK, cookie );
    userlog (LOG_ERR, S_WORK);
    return (-1);
}
bp = ( newstruct *)ObjCntlInfo->TrxData;
memset(bp, 0x00, sizeof(newstruct));
// bp->retval = 0;
// bp->newout.terror = 0;
/* ----- check the
Input data */
/* Replaced T.Kato 03.08.14 w-id is short range */
/* bp->newin.w_id = (short)srv-
>m_tcctxt[user_id].w_id; */
bp->newin.w_id = srv->m_tcctxt[user_id].w_id;
/* Replaced end */
if((bp->newin.d_id = str2int (in_data->D_ID, 2)) < 1) {
    userlog (LOG_ERR, "Input data error
[NEW] (d_id = %s)[Retuen_value:%d]\n",
in_data->D_ID,
bp->newin.d_id);
    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) {
    userlog (LOG_ERR, "Input data error
[NEW] (c_id = %s)[Retuen_value:%d]\n",
in_data->C_ID,
bp->newin.c_id);
    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){
        userlog (LOG_ERR, "Error end
chk_NOdata() [NEW] (Line:%d)[Return_Value:%d]\n",
cnt, rtn);
        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) {
    userlog (LOG_ERR, "nothing order line
data [NEW]\n");
    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;
    DBGRC(new_dsp(in_data, bp, srv-
>m_tcctxt[user_id].w_id, 0, ol_cnt));
    /* ----- Execute NewOrder
transaction */
    #ifndef SCRTEST
    resend_neworder:
        DBGRC(tsp(0));
        if ((ret_value = ObjCntInfo->SvrAplObj-
>ExNewOrder(int *)&bp) != S_OK) {
            ret_val = CreateTranErrReason(ret_value,
bp->newout.terror, &tran_errmsg);
            if (ret_val == 1) {
                goto resend_neworder; /*
Retry NewOrder transaction */
            } else {
                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 );
                userlog (LOG_ERR, S_WORK);
                return (-1);
            }
        }
        DBGRC(tsp(1));
    #else
        dummy_neworder( bp );
    #endif

    DBGRC(new_dsp(in_data, bp, srv-
>m_tcctxt[user_id].w_id, 1, 1));
    /* Replaced T.Kato 03.04.18 Speed up */
    //sprintf (S_WORK, h_new2);
    strcpy(S_WORK, h_new2);
    int h_new2_leng = strlen(S_WORK);
    /* 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) {
                userlog (LOG_ERR, "Error end
setNOdata() [NEW] (Line:%d)[Return_Value:%d]\n",
                    i,
return_value);
            }
        }
        /* "Item number is not valid" or "" ('\0') */
        // Oracle Web Server use
        if (strcmp(bp->newout.status, "I") > 0)
            alp2str ((S_WORK + newp[19]), 24, bp-
>newout.status);
        dec2str ((S_WORK + newp[20]), 8, (double)(bp-
>newout.total_amount)); // check
    /* ----- The execution result data notified RTE is make
by the HTML form */
    /* Replaced T.Kato 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
    int h_new1_leng;
    int h_new3_leng;
    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 */
    return (0);
}
#define SUPPLY_NG 0x01
#define L_ID_NG 0x02
#define QUANTITY_NG 0x04
/* -----
chk_NOdata :
VerifyNewOrderLine verifies that a user's inputs for a
line in
the New Order form are okay.
return -5 : w_id abnormal value : Not Number
return -6 : i_id abnormal value : Not Number
return -7 : ol_quantity abnormal value : Not
Number
98.8.3 : (-15, -16, -17: outside range )
-----
- */
int chk_NOdata (newstruct *bp, int cnt,
RTE_INPUT_DATA *in_data, int svcnt)
{
    char flag = 0;
    if (in_data->OL_SUPPLY_W_ID[cnt] == 0 &&
in_data->OL_I_ID[cnt] == 0 &&
in_data->OL_QUANTITY[cnt] == 0 ){
        /* Order line nothing : 1 */

```

```

        return 16; /* change return code */
    }
    if (in_data->OL_SUPPLY_W_ID[cnt] != 0){
        if((bp->newin.ol_supply_w_id[svcnt] =
str2int (in_data-
>OL_SUPPLY_W_ID[cnt], 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 99.12.20 */
        else if (bp->newin.ol_i_id[svcnt] == 0)
            bp->newin.ol_i_id[svcnt] = -1;
    }
    else{
        flag |= L_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 */
    DBGRC(sprintf(test_fp, "neworder ol data check
flag=%d\n",flag));
    if((flag & SUPPLY_NG) != 0) return -8;
    if((flag & L_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,
newstruct *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;
}
}
.....
TpDBDepend_Oracle/TrxOrderStatus.cpp
.....
*
*          *
*   TPC-C Client Application Program Source
*
*          *
*   Entry Functions          *
*   (1) OrderStatus          *
*
*   CREATE by TSL 2002.10.01   *
*
*   All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
.....
#include <windows.h>
#include "trans.h"
// Oracle Only -----
#include "tpcc_info.h"
// -----
#include "tpccis.h"
#include "fidtbl.h"
#include "odrpage.h"
#include "ObjectCntlInfo.h"
#include "TpcCntlInfo.h"
#include "Delivery.h"
#include "ExecuteThread.h"
#include "TpAplDBDependPrototype.h"
#include "log.h"
#include "tpapl.h"
#include "GlobalArea.h" // Common
#include "OracleInfo.h"
#include "OracleFunction.h"
.....
OrderStatus : this function processes the Orderstatus
transaction
.....
*/
int OrderStatus (char *s_buf, RTE_INPUT_DATA
"in_data, int cookie)
{
    ordstruct *bp;

```

```

int i, user_id, rtn;
char S_WORK[WORK_S];
char c_id_flag = NG;
char time_data[64], time_data0[64][64];
double deliver[15], entry;
/* Added T.Kato 03.10.20 SNAP_SHOT_TOO_OLD
retry error */
int retry_count = 0;
/* Added end */
//SvrAPL return value
#ifndef SCRTEST
HRESULT ret_value;
#endif
#ifdef
OBJECT_CNTL_INFO* ObjCntlInfo;
#endif
#ifdef SCRTEST
char* tran_errmsg;
int ret_val;
#endif
//for warning
time_data;
time_data0;
deliver;
entry;
MAC_PutFuncEntryLog("OrderStatus");
user_id = cookie - GLB_TermBase;
ObjCntlInfo = CreateSvrAplExecEnv();
if (ObjCntlInfo == 0) {
    sprintf (S_WORK, "Execution environment was not
created [ORD]\n");
    MAC_errHTML(s_buf, S_WORK, cookie );
    userlog (LOG_ERR, S_WORK);
    return (-1);
}
bp = ( ordstruct * )ObjCntlInfo->TrxData;
memset(bp, 0x00, sizeof(ordstruct));
// bp->retval = 0;
// bp->ordout.terror = 0;
/* ----- check the
Input data */
/* Replaced T.Kato 03.08.14 w-id is short range */
/* bp->ordin.w_id = (short)srv-
>m_tcctxt[user_id].w_id; */
bp->ordin.w_id = srv->m_tcctxt[user_id].w_id;
/* Replaced end */
/* check d_id data */
if ((bp->ordin.d_id == str2short (in_data->D_ID, 2) <
1) ) {
    userlog (LOG_ERR, "Input data error
[ORD] (d_id = %s)[Return_Value:%d]\n",
in_data->D_ID,
bp->ordin.d_id);
    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) {
    DBGPF(fprintf (test_fp, "check c_id=%x\n",
bp->ordin.c_id, 0, 0));
    if (bp->ordin.c_id < 0) {
        userlog (LOG_ERR, "Input data error
[ORD] (c_id = %s)[Return_Value:%d]\n",
in_data-
>C_ID, bp->ordin.c_id);
        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) {
    userlog (LOG_ERR, "c_id and c_last is
nothing [ORD]\n");
    return set_errpage(s_buf, cookie, 11, -4, 0, 0);
}
DBGPF(oder_dsp ( in_data, bp, srv-
>m_tcctxt[user_id].w_id, 0));
/* ----- Execute Orderstatus
transaction */
#ifndef SCRTEST
resend_orderstatus:
    DBGGR(tsp(0));
    if ((ret_value = ObjCntlInfo->SvrAplObj-
>ExOrderStatus((int *)&bp)) != S_OK) {
        ret_val = CreateTranErrReason(ret_value,
bp->ordout.terror, &tran_errmsg);
        if (ret_val == 1) {
            goto resend_orderstatus; /* tuxedo
busy -> retry!! */
        } else {
            /* Added T.Kato 03.10.20 SNAP_SHOT_TOO_OLD
retry error */
            retry_count++;
            if (retry_count < 5) {
                userlog (LOG_ERR,
"OrderStatus error retry %n times\n", retry_count);
                goto resend_orderstatus; /*
error -> retry!! */
            }
        }
    }
    /* Added end */
    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 );
    userlog (LOG_ERR, S_WORK);
    return (-1);
}
}
DBGGR(tsp(1));
/* Deleted 03.01.21 */
#endif

```

```

! bp = (ordstruct *)srv->m_tcctxt[user_id].trans_b;
#endif
/* Deleted end */
/* ----- Check the execution
result */
#else
dummy_orderstat( bp );
#endif
DBGP(oder_dsp ( in_data, bp, srv-
>m_tcctxt[user_id].w_id, 1));
/* Replaced T.Kato 03.04.18 Speed up */
//sprintf(S_WORK, h_order2);
strcpy(S_WORK, h_order2);
int h_order2_leng = strlen(S_WORK);
/* Replaced 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( strcmp( bp->ordout.ol_delivery_d[i],
"NOT DELIVR", 10 ) != 0 ) {
date2str
((S_WORK+i*0x3a+orderp[14]), bp-
>ordout.ol_delivery_d[i]);
}
}
/* ----- The execution result data notified RTE is make
by the HTML form */
/* Replaced T.Kato 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
int h_order1_leng;
int h_order3_leng;
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 */
return 0;
}
:
:
: TpDBDepend_Oracle/TrxPayment.cpp
:
:
/*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* (1) Payment
*
* CREATE by TSL 2002.10.01
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002
*****/
#include <windows.h>
#include "trans.h"
/* Oracle Only -----
#include "tpcc_info.h"
// -----
#include "tpccis.h"
#include "fdtbl.h"
#include "paypage.h"
#include "ObjectCntlInfo.h"
#include "TpcCntlInfo.h"
#include "Delivery.h"
#include "ExecuteThread.h"
#include "TpAplDBDependPrototype.h"
#include "log.h"
#include "tpapl.h"
#include "GlobalArea.h" // Common
#include "OracleFunction.h"
/-----
-
Payment : this function processes the Payment
transaction.
-*/
int Payment (char *s_buf, RTE_INPUT_DATA *in_data,
int cookie)
{
paystruct *bp;
int i, user_id, rtn;
float h_amount; /* For work */
char c_id_flag = NG;
char S_WORK[WORK_SIZE];
char time_data[64];
char buffer[128]; /* check HTML form */
char buffer2[128];
char buffer3[512];
int newlength;
double date, since;
//SvrAPL return value
#ifdef SCRTST
HRESULT ret_value;
#endif
}
OBJECT_CNTL_INFO* ObjCntlInfo;
#endifdef SCRTST
char* tran_errmsg;
int ret_val;
#endif
//for warning
time_data;
since;
date;
MAC_PutFuncEntryLog("Payment");
user_id = cookie - GLB_TermBase;
ObjCntlInfo = CreateSvrAplExecEnv();
if (ObjCntlInfo == 0) {
sprintf( S_WORK, "Execution environment was not
created [PAY]\n");
MAC_errHTML( s_buf, S_WORK, cookie );
userlog (LOG_ERR, S_WORK);
return (-1);
}
bp = ( paystruct * )ObjCntlInfo->TrxData;
memset(bp, 0x00, sizeof(paystruct));
// bp->retval = 0;
// bp-> payout.error = 0;
/* ----- check the
Input data */
/* Replaced T.Kato 03.08.14 w-id is short range */
/* bp->payin.w_id = (short)srv-
>m_tcctxt[user_id].w_id;
bp->payin.w_id = srv->m_tcctxt[user_id].w_id;
/* Replaced end */
/* check d_id data */
if((bp->payin.d_id = str2short (in_data->D_ID, 2)) <
1) {
userlog (LOG_ERR, "Input data error [PAY]
(d_id = %s)[Return_Value:%d]\n",
in_data->D_ID,
bp->payin.d_id);
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) {
userlog (LOG_ERR, "Input data error
[PAY] (c_id = %s)[Return_Value:%d]\n",
in_data-
>C_ID, bp->payin.c_id);
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) {
    userlog (LOG_ERR, "c_id and c_last data
is nothing [PAY]\n");
    return set_errpage(s_buf, cookie, 11, -4, 0,
0);
}
/* check c_w_id data */
/* Replaced T.Kato 03.08.20 Bug fix --effect flowing
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 */
        userlog (LOG_ERR, "Input data error [PAY]
(c_w_id = %s)[Return_Value:%d]\n",
                in_data-
>C_W_ID, bp->payin.c_w_id);
        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) {
        userlog (LOG_ERR, "Input data error [PAY]
(c_d_id = %s)[Return_Value:%d]\n",
                in_data-
>C_D_ID, bp->payin.c_d_id);
        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) {
        userlog (LOG_ERR, "Input data error [PAY]
(h_amount = %s)[Return_Value:%d]\n",
                in_data-
>H_AMOUNT, bp->payin.h_amount);
        return set_errpage(s_buf, cookie, 8, (int)bp-
>payin.h_amount, 0, 0);
    }
    DBGP(pay_dsp(in_data, bp, srv-
>m_tcctxt[user_id].w_id, 0));
/* ----- Execute Payment
transaction */
#ifdef SCRTEST
    resend_payment:
    DBGR(tsp(0));
    if ((ret_value = ObjCntlInfo->SvrAplObj-
>ExPayment((int *)&bp) != S_OK) {
        ret_val = CreateTranErrReason(ret_value,
bp->payout.terror, &tran_errmsg);
        if (ret_val == 1) {
            goto resend_payment; /* TP_base
busy. Try again */
        } else {
            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);
            userlog (LOG_ERR, S_WORK);
            return (-1);
        }
    }
}
DBGRT(tsp(1));
#else
    dummy_payment (bp);
#endif
    DBGR(pay_dsp(in_data, bp, srv-
>m_tcctxt[user_id].w_id, 1));
/* Replaced T.Kato 03.04.18 Speed up */
    //sprintf (S_WORK, h_pay2);
    strcpy(S_WORK, 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;
    newlength = checkHTMLform (&buffer2[0],
&buffer[0]);
    strcpy (&buffer3[0], &buffer[0]);
    strcat (buffer3, " ");
    alp2str (buffer2, 20, bp->payout.d_street_1);
    buffer2[20] = 0;
    newlength = checkHTMLform (&buffer2[0],
&buffer[0]);
    strcat (buffer3, &buffer[0]);
    strcat (buffer3, "\r\n");
    alp2str (buffer2, 20, bp->payout.w_street_2);
    buffer2[20] = 0;
    newlength = checkHTMLform (&buffer2[0],
&buffer[0]);
    strcat (buffer3, &buffer[0]);
    strcat (buffer3, "\r\n");
/* Replaed T.Kato 03.04.18 Speed up */
    //sprintf (buffer3, h_pay4);
    strcpy (buffer3, h_pay4);
/* 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);
    strcat (S_WORK, buffer3);
    strcpy (&buffer3[0], " ");
    alp2str (buffer2, 20, bp->payout.c_street_1);
    buffer2[20] = 0;
    newlength = checkHTMLform (&buffer2[0],
&buffer[0]);
    strcat (buffer3, &buffer[0]);
    strcat (buffer3, " Credit: ");
    alp2str (buffer2, 2, bp->payout.c_credit);
    buffer2[2] = 0;
    strcat (buffer3, &buffer2[0]);
    strcat (buffer3, "\r\n");
    strcat (buffer3, " ");
    alp2str (buffer2, 20, bp->payout.c_street_2);
    buffer2[20] = 0;
    newlength = checkHTMLform (&buffer2[0],
&buffer[0]);
    strcat (buffer3, &buffer[0]);
    strcat (buffer3, " %Disc: ");
    strcat (S_WORK, buffer3);
    dec2str (&buffer3[0], 5,
        (double)((double)(bp->payout.c_discount) *
(double)100.0));
    sprintf (&buffer3[5], "\r\n");
    strcat (S_WORK, buffer3);
/* Replaced T.Kato 03.04.18 Speed up */
    //sprintf (buffer3, h_pay5);
    strcpy (buffer3, h_pay5);
/* 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);
    strcat (S_WORK, buffer3);
    if ( (i = strlen (bp->payout.c_data)) <= 0) {
        sprintf (&buffer3[0], "\r\n\r\n\r\n\r\n");
    }
    else{
        alp2str (buffer2, 50, bp->payout.c_data);
        buffer2[50] = 0;
        newlength = checkHTMLform (&buffer2[0],
&buffer[0]);
        strcpy (&buffer3[0], &buffer[0]);
        strcat (buffer3, "\r\n");
    }
}

```



```

! 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
    int h_stock1_leng;
    int h_stock3_leng;
    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 */
return (0);
}
.....
TpDBDepend_Oracle/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)
userlog(LOG_INF, ">>>>> "func" start >>>>>");
#else
#define MAC_PutFncEntryLog(func) ;
#endif
/* Function exit point log */
#ifdef PUT_FNC_EXIT_LOG
#define MAC_PutFncExitLog(func)
userlog(LOG_INF, "<<<<< "func" end <<<<<");
#else
#define MAC_PutFncExitLog(func) ;
#endif
.....
TpDBDepend_Oracle/tpcc_info.h
.....
/*****
*
* COPYRIGHT FUJITSU LIMITED 2002
* CREATE:2000.02.14 FJH
*
*
*/
* $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 ALLTXNODEL 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 rctval;
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[20];
char h_date[20];
int retry;
};

struct paystruct {
int rctval;
struct payinstruct payin;
struct payoutstruct payout;
};

/* Order status */
struct ordinstruc {
int w_id;
int d_id;
int c_id;
int bylastname;
char c_last[17];
};

struct ordoutstruct {
int terror;
int c_id;
char c_last[17];
char c_first[17];
char c_middle[3];
double c_balance;
int o_id;
char o_entry_d[20];
int o_carrier_id;
int o_ol_cnt;
int ol_supply_w_id[15];
int ol_i_id[15];
int ol_quantity[15];
float ol_amount[15];
char ol_delivery_d[15][11];
int retry;
};

```

```

struct ordstruct {
    int retval;
    struct ordinstruct ordin;
    struct ordoutstruct ordout;
};
/* Delivery */
struct delinstruct {
    int w_id;
    int o_carrier_id;
    long startsec;
    long startusec;
    /* double qtime;
    int in_timing_int;*/
};
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

```


Appendix B: Server Source Code

```

.....
SvrAplBase/CntlExecuteThread.cpp
.....
/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* 1) InitObjectiveSvrApl : Create object
executing info.
* and initialize SvrApl object.
* 2) GetIdleSvrAplObject : Get Idling object
pointer.
* 3) ReleaseObjectCntlInfo: Release Svr Apl
object.
* 4) TermObjectiveSvrApl: Terminate SvrApl
and terminate
* control information.
*
* CREATE by TSL 2003.05.21
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
**/
#include "stdafx.h"
#include <process.h>
#include "ProcCntl.h"
#include "log.h"
#include "GlobalArea.h"
#ifdef PROC_MODE /////// Executing mode is
PRCODESS. ///////
#else /////// Executing mode is
THREAD. ///////
/*****
* Create object executing information table,
* and initialize SvrApl object.
*
* Return Value
* 0 : Normal end
* !0 : Error end
*****
**/
int InitObjectiveSvrApl() {
OBJECT_CNTL_INFO* object_cntl_info =
0;
int i;
MAC_PutFncEntryLog("InitObjectiveSvrApl"
);

```

```

/* Alloc object control information by
NumDBCntlProcesses, */
/* where the INI FILE.
*/
object_cntl_info = object_cntl_info +
GLB_NumControlProcesses;
GLB_ObjectCntlInfo =
(OBJECT_CNTL_INFO*)malloc((size_t)object_cntl_info);
if (GLB_ObjectCntlInfo == 0) {
/* Memory allocation fail */
userlog(LOG_ERR, "Object control
area allocation failed (request=%dbytes)",
object_cntl_info);
goto CREATE_FAIL;
}
memset((void*)GLB_ObjectCntlInfo, 0x00,
(size_t)object_cntl_info);
/* Create SvrApl object, and initialize SvrApl
*/
for (object_cntl_info = GLB_ObjectCntlInfo, i
= 0;
i < GLB_NumControlProcesses;
object_cntl_info++, i++) {
/* Create SvrApl object */
if ((object_cntl_info->object_pointer =
ExSvrAplCreate()) == 0) {
userlog(LOG_ERR,
"ExSvrAplCreate() failed to Create object
(%d/%d)",
GLB_NumControlProcesses,
i);
goto CREATE_FAIL;
}
/* Initialize SvrApl (Connect DB
server) */
if (ExSvrAplInitiate == NULL) {
userlog(LOG_ERR,
"ExSvrAplInitiate() not call, because of NULL
entry");
goto CREATE_FAIL;
}
ExSvrAplInitiate(object_cntl_info-
>object_pointer);
/* Create event for object idle posting */
if ((GLB_ObjectIdleEvent =
CreateEvent(NULL, FALSE, FALSE,
OBJECT_CNTL_IDLE_EVENT)) == NULL) {
userlog(LOG_ERR, "Object idle
posting event create fail, CreateEvent() <%s>
GetLastError() = %d\n",
OBJECT_CNTL_IDLE_EVENT,
GetLastError());
goto CREATE_FAIL;
}
/* Normal end */
return 0;
CREATE_FAIL:
TermObjectiveSvrApl();
return 1;
}
/*****
**/

```

```

* Get idling SvrApl object.
*
* If idling object is nothing, wait until released
object.*
*
* Return Value
* !0 : Usable object pointer
*
* 0 : Error end
*****
**/
OBJECT_CNTL_INFO* GetIdleSvrAplObject() {
DWORD ret_value;
int i;
OBJECT_CNTL_INFO* object_cntl_info =
0;
MAC_PutFncEntryLog("GetIdleSvrAplObjec
t");
if (GLB_ObjectCntlInfo == 0)
return 0;
/* Lock Thread control table, while finding
idle object */
EnterCriticalSection(&GLB_ObjectCntlCritic
al);
while(1) {
/* Search process control table for
finding idle process */
object_cntl_info =
GLB_ObjectCntlInfo;
for(i = 0; i <
GLB_NumControlProcesses; i++) {
if (object_cntl_info-
>object_status == OBJ_STAT_IDLE) {
/* Found idle object */
break;
}
object_cntl_info++;
}
if (i != GLB_NumControlProcesses) {
/* Find idle object */
break;
}
/* All object busy, waiting any object
idle */
if ((ret_value =
WaitForSingleObject(GLB_ObjectIdleEvent,
INFINITE)) != WAIT_OBJECT_0) {
LeaveCriticalSection(&GLB_ObjectCntlCritic
al);
userlog(LOG_ERR, "Illegal
object idle waiting (WaitForSingleObject()
ret_val=%d GetLastError=%d)",
ret_value, GetLastError());
return 0;
}
}
/* Find idle process, then set process busy
and unlock process control table */

```

```

    object_cntl_info->object_status =
OBJ_STAT_BUSY;
    LeaveCriticalSection(&GLB_ObjectCntlCritical);
    return object_cntl_info;
}
/*****
* Release SvrApl object.
*
* Set maker of idle object, and assert idle event.
*
* Return Value
* NONE
*****/
int ReleaseObjectCntlInfo(OBJECT_CNTL_INFO*
object_cntl_info) {
    MAC_PutFncEntryLog("ReleaseObjectCntlInfo");
    EnterCriticalSection(&GLB_ObjectCntlCritical_SetEvent);
    object_cntl_info->object_status =
OBJ_STAT_IDLE;
    if (!SetEvent(GLB_ObjectIdleEvent)) {
        userlog(LOG_ERR, "SetEvent() for
posting object idle GetLastError=%d",
GetLastError());

        LeaveCriticalSection(&GLB_ObjectCntlCritical_SetEvent);
        return -1;
    }
    LeaveCriticalSection(&GLB_ObjectCntlCritical_SetEvent);
    return 0;
}
/*****
* Terminate objective SvrApl.
*
* All SvrApl terminate, and delete object.
*
* Return Value
* NONE
*****/
void TermObjectiveSvrApl() {
    int i;
    OBJECT_CNTL_INFO* object_cntl_info =
0;
    MAC_PutFncEntryLog("TermObjectiveSvrApl");
    if (GLB_TermnateObject ==
TRMOBJ_TERMINATE)
        /* Already terminated */
        return;
    GLB_TermnateObject =
TRMOBJ_TERMINATE;

    /* Release idle posting event */
    if (GLB_ObjectIdleEvent != 0) {

```

```

        if
(CloseHandle(GLB_ObjectIdleEvent) == 0)
            userlog(LOG_ERR, "object idle
event release failed GetLastError=%d",
GetLastError());
            GLB_ObjectIdleEvent = 0;
        }
        /* Terminate SvrApl object */
        if (GLB_ObjectCntlInfo == 0)
            return;
        /* Replaced T.Kato 03.07.29 Change log output
sequence */
        #if 0
        ! for (object_cntl_info = GLB_ObjectCntlInfo, i
= 0;
        ! i < GLB_NumControlProcesses;
        ! object_cntl_info++, i++) {
        !
        ! /* Terminate SvrApl object */
        ! if (object_cntl_info->object_pointer !=
0) {
        ! /* Object existing */
        ! if (ExSvrAplTerminate ==
NULL) {
        ! userlog(LOG_ERR,
"ExSvrAplTerminate() not call, because of NULL
entry");
        ! break;
        ! }
        !
        ! ExSvrAplTerminate(object_cntl_info-
>object_pointer);
        !
        ! ExSvrAplDelete(object_cntl_info-
>object_pointer);
        !
        ! }
        !
        ! /* Output result of executing
transactions */
        ! userlog(LOG_INF, "OBJECT-%d
executes Total=%d New=%d Pay=%d Sto=%d
Odr=%d Del=%d",
        ! i,
        ! object_cntl_info->exec_new +
        ! object_cntl_info->exec_pay +
        ! object_cntl_info->exec_sto +
        ! object_cntl_info->exec_odr +
        ! object_cntl_info->exec_del,
        ! 1
        !
        ! object_cntl_info->exec_new,
        ! object_cntl_info->exec_pay,
        ! object_cntl_info->exec_sto,
        ! object_cntl_info->exec_odr,
        ! object_cntl_info->exec_del);
        ! }
        #endif

```

```

        /* Output no. of execution transaction */
        for (object_cntl_info = GLB_ObjectCntlInfo, i
= 0;
        ! i < GLB_NumControlProcesses;
        ! object_cntl_info++, i++) {
        !
        ! /* Output result of executing
transactions */
        ! userlog(LOG_INF, "OBJECT-%d
executes Total=%d New=%d Pay=%d Sto=%d
Odr=%d Del=%d",
        ! i,
        ! object_cntl_info->exec_new +
        ! object_cntl_info->exec_pay +
        ! object_cntl_info->exec_sto +
        ! object_cntl_info->exec_odr +
        ! object_cntl_info->exec_del,
        ! object_cntl_info->exec_new,
        ! object_cntl_info->exec_pay,
        ! object_cntl_info->exec_sto,
        ! object_cntl_info->exec_odr,
        ! object_cntl_info->exec_del);
        !
        ! }
        /* Output transaction executing status */
        for (object_cntl_info = GLB_ObjectCntlInfo, i
= 0;
        ! i < GLB_NumControlProcesses;
        ! object_cntl_info++, i++) {
        !
        ! /* Output result of executing
transactions */
        ! if (object_cntl_info->transaction_stat
== STAT_RESPONSE)
        ! userlog(LOG_INF, "OBJ-%d
executing status is NORMAL", i);
        ! else
        ! userlog(LOG_ERR, "OBJ-%d
executing status is waiting a response [stat=%d]",
        ! i,
        ! object_cntl_info->transaction_stat);
        ! }
        /* Terminate SvrApl object */
        for (object_cntl_info = GLB_ObjectCntlInfo, i
= 0;
        ! i < GLB_NumControlProcesses;
        ! object_cntl_info++, i++) {
        ! /* Terminate SvrApl object */
        ! if (object_cntl_info->object_pointer !=
0) {
        ! /* Object existing */
        ! if (ExSvrAplTerminate == NULL
||

```

```

        ExSvrAplDelete ==
NULL) {
    userlog(LOG_ERR,
"ExSvrAplTerminate()/ExSvrAplDelete() not call,
because of NULL entry");
    break;
}
    userlog(LOG_INF, "OBJ-%d
Terminate SvrApl stating", i);

    ExSvrAplTerminate(object_cntl_info-
>object_pointer);
    userlog(LOG_INF, "    Delete
SvrApl stating");

    ExSvrAplDelete(object_cntl_info-
>object_pointer);
    userlog(LOG_INF, "
Terminate SvrApl all ended");
}
}
/* Replaced end */

/* Release object control information */
if (GLB_ObjectCntlInfo != 0) {
    free(GLB_ObjectCntlInfo);
    GLB_ObjectCntlInfo = 0;
}
return;
}
#endif
.....
SvrAplBase/CreateCntlProcess.cpp
.....
/*****
***
*
*   TPC-C Client Application Program Source
*
*
*   Entry Functions
*   1) CreateCntlProcess    : Create DB access
cntl process*
*
*   CREATE by TSL   2003.02.19
*
*
*   All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
**/
#include "stdafx.h"
#include <process.h>
#include <io.h>
#include <stdio.h>
#include "SvrAplBase.h"
#include "ProcCntl.h"
#include "SvrApl.h"
#include "log.h"
#include "MapFileInfo.h"
#include "GlobalArea.h"

#ifdef PROC_MODE    //!!!! Executing mode is
PRCODESS.!!!!!!
/*****
***

```

```

* Create DB accessing control process.
*
* Return Value
* TRUE : Normal end
* FALSE : Abnormal end
*
*****
**/
bool CreateCntlProcess() {
    PROC_CNTL_INFO*   proc_cntl_info = 0;
    STARTUPINFO        startup_info;
    PROCESS_INFORMATION process_info;
    int                i;
    char exe_path[MAX_PATH];
    DWORD ret_value;
    char prefix[8];
    MAC_PutFuncEntryLog("CreateCntlProcess"
);
    /* Create the process control process */
    proc_cntl_info = proc_cntl_info +
GLB_NumControlProcesses;
    if ((GLB_ProcCntlInfo =
(PROC_CNTL_INFO*)CreateSharedMemory((uns
igned long)proc_cntl_info,

    PROC_CNTL_TBL_SHAREDMEM,

    &GLB_MappingHandle)) == 0) {
        userlog(LOG_ERR,
"CreateSharedMemory() for Control table fail
mapping size = %d", (long)proc_cntl_info);
        return FALSE;
    }
    memset((void*)GLB_ProcCntlInfo, 0x00,
(unsigned long)proc_cntl_info);
    /* Create event for process idle posting */
    if ((GLB_ProcIdleEvent =
CreateEvent(NULL, FALSE, FALSE,
PROC_CNTL_IDLE_EVENT)) == NULL) {
        userlog(LOG_ERR, "CreateEvent()
Idle fail, <%=s> GetLastError() = %d\n",

        PROC_CNTL_IDLE_EVENT,
GetLastError());
        return FALSE;
    }
    /* Create DB control process */
    memset(&startup_info, 0x00,
sizeof(startup_info));
    proc_cntl_info = GLB_ProcCntlInfo;
    for(i = 0; i < GLB_NumControlProcesses;
i++) {
        /* Make command line */
        sprintf(exe_path,
"%sDBprocCntl.exe %04d", GLB_DllPath, i);
        if (CreateProcess(0,

            exe_path,
            0,
            0,
            FALSE,

            NORMAL_PRIORITY_CLASS |
CREATE_NEW_PROCESS_GROUP |
DETACHED_PROCESS,

            0,
            0,

```

```

&startup_info,
&process_info)
    {
        /* Successfully create process */
        userlog(LOG_INF, "DB control
process create success [process no %d]", i);

        CloseHandle(process_info.hProcess);

        CloseHandle(process_info.hThread);
    } else {
        userlog(LOG_ERR, "DB control
process create fail [process no %d
GetLastError %d]",

            i,
GetLastError());
        return FALSE;
    }

    /* Wait for process initializing end */
    if ((ret_value =
WaitForSingleObject(GLB_ProcIdleEvent,
INFINITE)) != WAIT_OBJECT_0) {
        userlog(LOG_ERR, "Illegal
Contrpl process initializing
ended(WaitForSingleObject() ret_val=%d
GetLastError=%d)",

        ret_value, GetLastError());
        return FALSE;
    }

    /* Create shared memory for I/O area
*/
    sprintf(prefix, "%04d", i);
    if ((proc_cntl_info->shared_memory =
CreateSharedMemory(MAP_DATA_OFFSE
T, prefix, &proc_cntl_info->shared_mapping)) ==
0) {
        userlog(LOG_ERR,
"CreateSharedMemory() for Data buffer fail
mapping size = %d process no. = %s",

        MAP_DATA_OFFSET, prefix);
        return FALSE;
    }
    /* Create command/response events
*/
    if
(MakePostingEvents(&proc_cntl_info-
>command_event,

    &proc_cntl_info->response_event, prefix) != 0) {
        userlog(LOG_ERR,
"MakePostingEvents() Command/Response fail
[process no.=%s]", prefix);
        return FALSE;
    }
    /* Set next Table Pointer */
    proc_cntl_info++;
}
}
/* Normal ended */
return TRUE;
}
#endif
.....

```

```

SvrAplBase/GetCntlProcessNum.cpp
.....
/*****
***
*
* TPC-C Client Application Program Source
*
* Entry Functions
* 1) GetCntlProcessNum : Get number of
create processes *
*
* CREATE by TSL 2003.02.19
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
**/
#include "stdafx.h"
#include <process.h>
#include <io.h>
#include <stdio.h>
#include "log.h"
#include "ProcCntl.h"
#include "GlobalArea.h"
/*****
***
* Get number of create processes from INI file.
*
* Return Value
* NONE
*
*****
**/
void GetCntlProcessNum() {
    char IniPath[MAX_PATH];
    char value_buf[16];
    int i;

#define INI_FILE_NAME "TpApl.ini"
#define DEFAULT_CNTLPROCESS 30
    MAC_PutFuncEntryLog("GetCntlProcessNu
m");

    /* Make INI file path */
    strcpy(IniPath, GLB_DllPath);
    strcat(IniPath, INI_FILE_NAME);

    /* Check INI file exist */
    if (_access(IniPath, 0x00) != 0) {
        /* INI file no exist, using default value */
        userlog(LOG_WRN, "INI file nothing,
using default value");
        GLB_NumControlProcesses =
DEFAULT_CNTLPROCESS;
    } else {
        /* Get execution informations
*/
        /* If undefined key and illegal value,
using default value */
        for (i = 0; i < 3; i++) {

            GetPrivateProfileString("TPCC_PROCESS",
"NumDBCntlProcesses", "",
value_buf, sizeof(value_buf), IniPath);
            if (value_buf[0] == "") {

```

```

        /* if Key is nothing, retry
getting */
            continue;
        }
        break;
    }
    userlog(LOG_INF, "INI file information
[NumDBCntlProcesses]=[%s]", value_buf);
    if (value_buf[0] == "") {
        /* Target key was nothing, use
default value */
        GLB_NumControlProcesses =
DEFAULT_CNTLPROCESS;
    } else {
        GLB_NumControlProcesses =
atoi(value_buf);
    }
}
.....
SvrAplBase/GlobalArea.cpp
.....
/*****
***
*
* TPC-C Client Application Program Source
*
* Entry Functions
* Global area
*
* CREATE by TSL 2002.12.18
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****
**/
#include "stdafx.h"
#include "ProcCntl.h"
#include "GlobalArea.h"
/* Path informations */
char GLB_DllPath[MAX_PATH];
char GLB_LogFilePath[MAX_PATH];
/* Number of Control process or control object */
int GLB_NumControlProcesses =
0;
int GLB_InitComplete =
INIT_FIRST_COND;

#ifdef PROC_MODE //!!!! Executing mode is
PRCODESS. //!!!!
/* Control falgs */
int GLB_TerminateProcessing =
TRMPROC_NOTERMINATE;
/* Process Control Process information */
PROC_CNTL_INFO*
GLB_ProcCntlInfo = 0;
HANDLE
GLB_MappingHandle;
HANDLE
GLB_ProcIdleEvent;
CRITICAL_SECTION
GLB_ProcCntlCritical;

#else //!!!! Executing mode is THREAD. //!!!!

```

```

/* Control falgs */
int GLB_TerminateObject =
TRMOBJ_NOTERMINATE;

/* Database depended function entry */
void* (FAR* ExSvrAplCreate)();
void (FAR* ExSvrAplDelete)(void*
object_pointer);
void (FAR* ExSvrAplInitiate)(void*
object_pointer);
void (FAR* ExSvrAplTerminate)(void*
object_pointer);
int (FAR* ExTrxNewOrder)(void*
new_order_struct, void* object_pointer);
int (FAR* ExTrxPayment)(void*
payment_struct, void* object_pointer);
int (FAR* ExTrxOrderStatus)(void*
order_status_struct, void* object_pointer);
int (FAR* ExTrxStockLevel)(void*
stock_level_struct, void* object_pointer);
int (FAR* ExTrxDelivery)(void*
delivery_struct, void* object_pointer);

/* DLL instance handle */
HINSTANCE GLB_dll_inst;

/* Object Control information */
OBJECT_CNTL_INFO*
GLB_ObjectCntlInfo;
HANDLE
GLB_ObjectIdleEvent;
CRITICAL_SECTION
GLB_ObjectCntlCritical;
CRITICAL_SECTION
GLB_ObjectCntlCritical_SetEvent;
CRITICAL_SECTION
GLB_TransactionCritical;
/* Added T.Kato 03.08.24 Response data
sampling */
#ifdef SAMPLE
    SAMPLING_DATA obj_sample_data;
    SAMPLING_DATA new_sample_data;
    SAMPLING_DATA pay_sample_data;
    SAMPLING_DATA odr_sample_data;
    SAMPLING_DATA del_sample_data;
    SAMPLING_DATA sto_sample_data;
#endif
/* Added end */
#endif
.....
SvrAplBase/GlobalArea.h
.....
/*****
***
*
* TPC-C Client Application Program Source
*
* Entry Functions
* Global area externs
*
* CREATE by TSL 2002.12.18
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *

```

```

*****
**)
/* Path informations */
extern char GLB_DllPath[MAX_PATH];
extern int
    GLB_NumControlProcesses;
/* Control falgs */
extern int GLB_InitComplete;
#define INIT_FIRST_COND 0
#define INIT_COMPLETE 1
#ifdef PROC_MODE // Executing mode is
PRCODESS. //
/* Control falgs */
extern int
    GLB_TermnateProcessing;
#define TRMPROC_NOTERMINATE 0
#define TRMPROC_TERMINATE 1
/* Grobal functions */
bool InitSvrApiCntl();
/* Process Control Process information */
extern PROC_CNTL_INFO*
    GLB_ProcCntlInfo;
extern HANDLE
    GLB_MappingHandle;
extern HANDLE
    GLB_ProcIdleEvent;
extern CRITICAL_SECTION
    GLB_ProcCntlCritical;
#else // Executing mode is
THREAD. //
/* Control falgs */
extern int GLB_TermnateObject;
#define TRMOBJ_NOTERMINATE 0
#define TRMOBJ_TERMINATE 1
/* Name of DLL */
#define DLL_NAME "SvrDBDdepend.dll"

/* Database depended function entry */
extern void* (FAR* ExSvrApiCreate)();
extern void (FAR* ExSvrApiDelete)(void*
object_pointer);
extern void (FAR* ExSvrApiInitiate)(void*
object_pointer);
extern void (FAR*
ExSrvApiTerminate)(void* object_pointer);
extern int (FAR*
ExTrxNewOrder)(void* new_order_struct, void*
object_pointer);
extern int (FAR*
ExTrxPayment)(void* payment_struct, void*
object_pointer);
extern int (FAR*
ExTrxOrderStatus)(void* order_status_struct,
void* object_pointer);
extern int (FAR*
ExTrxStockLevel)(void* stock_level_struct, void*
object_pointer);
extern int (FAR*
ExTrxDelivery)(void* delivery_struct, void*
object_pointer);
/* DLL instance handle */
extern HINSTANCE GLB_dll_inst;
/* Object Control information */
extern OBJECT_CNTL_INFO*
    GLB_ObjectCntlInfo;
extern HANDLE
    GLB_ObjectIdleEvent;

```

```

extern CRITICAL_SECTION
    GLB_ObjectCntlCritical;
extern CRITICAL_SECTION
    GLB_ObjectCntlCritical_SetEvent;
extern CRITICAL_SECTION
    GLB_TransactionCritical;
/* Added T.Kato 03.08.24 Response data
sampling */
#ifdef SAMPLE
typedef struct _sample_info {
    long sample_function;
    int obj_max, obj_min, obj_ttl,
obj_cnt;
    int new_max, new_min,
new_ttl, new_cnt;
    int pay_max, pay_min,
pay_ttl, pay_cnt;
    int odr_max, odr_min, odr_ttl,
odr_cnt;
    int del_max, del_min, del_ttl,
del_cnt;
    int sto_max, sto_min, sto_ttl,
sto_cnt;
} SAMPLE_INFO;
typedef struct _sampling_data {
    DWORD max;
    DWORD min;
    DWORD total_time;
    DWORD count;
} SAMPLING_DATA;
extern SAMPLING_DATA
    obj_sample_data;
extern SAMPLING_DATA
    new_sample_data;
extern SAMPLING_DATA
    pay_sample_data;
extern SAMPLING_DATA
    odr_sample_data;
extern SAMPLING_DATA
    del_sample_data;
extern SAMPLING_DATA
    sto_sample_data;
#define MAC_SampleInit \
\
    memset(&obj_sample_data, 0x00,
sizeof(obj_sample_data)); obj_sample_data.min =
0x7fffffff; \
\
    memset(&new_sample_data, 0x00,
sizeof(new_sample_data));
new_sample_data.min = 0x7fffffff; \
\
    memset(&pay_sample_data, 0x00,
sizeof(pay_sample_data));
pay_sample_data.min = 0x7fffffff; \
\
    memset(&odr_sample_data, 0x00,
sizeof(odr_sample_data));
odr_sample_data.min = 0x7fffffff; \
\
    memset(&del_sample_data, 0x00,
sizeof(del_sample_data));
del_sample_data.min = 0x7fffffff; \
\
    memset(&sto_sample_data, 0x00,
sizeof(sto_sample_data));
sto_sample_data.min = 0x7fffffff; \
\
#define MAC_SampleWork \
\

```

```

    DWORD
    sample_start_time; \
    DWORD
    sample_proc_time;
#define MAC_SampleStartTime \
    sample_start_time =
GetTickCount(); \
    //Sleep(50);
#define
    MAC_SampleGlobalSet(data_area)
\
    sample_proc_time =
GetTickCount() - sample_start_time; \
    data_area.total_time +=
sample_proc_time; \
    data_area.count++;
\
    if (data_area.max <
sample_proc_time) { \
        data_area.max =
sample_proc_time; \
    }
\
    if (data_area.min >
sample_proc_time) { \
        data_area.min =
sample_proc_time; \
    }
#define MAC_SampleClear(data_area)
\
    data_area.count =
data_area.total_time = 0;
#define
    MAC_SampleDataSet(area_pointer)
\
    SAMPLE_INFO*
    sampl_info;
\
    sampl_info =
(SAMPLE_INFO*)area_pointer;
\
    if (sampl_info-
>sample_function != 0) { \
        sampl_info-
>obj_max = obj_sample_data.max; \
        sampl_info-
>obj_min = obj_sample_data.min == 0x7fffffff? 0:
obj_sample_data.min; \
        sampl_info->obj_ttl
= obj_sample_data.total_time; \
        sampl_info-
>obj_cnt = obj_sample_data.count; \
        sampl_info-
>new_max = new_sample_data.max;
\
        sampl_info-
>new_min = new_sample_data.min == 0x7fffffff?
0: new_sample_data.min; \
        sampl_info-
>new_ttl = new_sample_data.total_time; \
        sampl_info-
>new_cnt = new_sample_data.count; \
        sampl_info-
>pay_max = pay_sample_data.max; \

```

```

        sampl_info-
>pay_min = pay_sample_data.min == 0x7fffffff? 0:
pay_sample_data.min;\
        sampl_info->pay_ttl
= pay_sample_data.total_time; \
        sampl_info-
>pay_cnt = pay_sample_data.count; \
        sampl_info-
>odr_max = odr_sample_data.max; \
        sampl_info-
>odr_min = odr_sample_data.min == 0x7fffffff? 0:
odr_sample_data.min;\
        sampl_info->odr_ttl
= odr_sample_data.total_time; \
        sampl_info-
>odr_cnt = odr_sample_data.count; \
        sampl_info-
>del_max = del_sample_data.max; \
        sampl_info-
>del_min = del_sample_data.min == 0x7fffffff? 0:
del_sample_data.min;\
        sampl_info->del_ttl
= del_sample_data.total_time;\
        sampl_info-
>del_cnt = del_sample_data.count; \
        sampl_info-
>sto_max = sto_sample_data.max; \
        sampl_info-
>sto_min = sto_sample_data.min == 0x7fffffff? 0:
sto_sample_data.min; \
        sampl_info->sto_ttl
= sto_sample_data.total_time;\
        sampl_info-
>sto_cnt = sto_sample_data.count; \
        MAC_SampleClear(obj_sample_data);
        \
        MAC_SampleClear(new_sample_data);
        \
        MAC_SampleClear(pay_sample_data);
        \
        MAC_SampleClear(odr_sample_data);
        \
        MAC_SampleClear(del_sample_data);
        \
        MAC_SampleClear(sto_sample_data);
        \
        return S_OK;
    \
}
#else
#define MAC_SampleInit
;
#define MAC_SampleWork
;
#define MAC_SampleStartTime
;
#define MAC_SampleGlobalSet(data_area) ;
#define MAC_SampleClear(data_area)
;
#define MAC_SampleDataSet(area_pointer);

```

```

#endif
/* Added end */
#endif
.....
SvrAplBase/InitSvrAplCntl.cpp
.....
/******
***
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* 1) InitSvrAplCntl : Initialize SvrApl
Controller *
*
* CREATE by TSL 2003.02.19
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
**/
#include "stdafx.h"
#include <process.h>
#include "SvrAplBase.h"
#include "ProcCntl.h"
#include "SvrApl.h"
#include "log.h"
#include "MapFileInfo.h"
#include "GlobalArea.h"
#ifdef PROC_MODE /////// Executing mode is
PRCODESS. ///////
/******
***
* Initialize SvrApl DB control process
*
* Return Value *
* None *
*****
**/
void CSvrApl::InitSvrAplCntl() {
    MAC_PutFuncEntryLog("CSvrApl::InitSvrApl
Cntl");
    /* Create DB Accessing process */
    if (GLB_FirstProcessing ==
FSTPROC_NOCREATE) {
        /* Control Process is not created */
        EnterCriticalSection(&GLB_ProcCntlCritical)
;
        if (GLB_FirstProcessing ==
FSTPROC_NOCREATE) {
            /* Only one executing */
            if (!CreateCntlProcess()) {
                userlog(LOG_ERR,
"Control Process creating failed");
                return;
            }
            GLB_FirstProcessing =
FSTPROC_CREATE;
        }
        LeaveCriticalSection(&GLB_ProcCntlCritical
);
    }
}

```

```

/* Set Initialize end */
m_InitComplete = INIT_COMPLETE;
return;
}
#endif ///////
.....
SvrAplBase/LoadSvrAplDll.cpp
.....
/******
***
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Globla area
*
*
* CREATE by TSL 2003.05.12
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
**/
#include "stdafx.h"
#include "ProcCntl.h"
#include "log.h"
#include "GlobalArea.h"
#ifdef PROC_MODE /////// Executing mode is
PRCODESS. ///////
#else /////// Executing mode is
THREAD. ///////
/******
***
* Load database depended DLL and set
function entry *
*
* Return *
*
* 0 : Normal end *
* !0 : Error end *
*****
int LoadSvrAplDll() {
    char load_dll_path[_MAX_PATH];
    // Load SvrDBDepended.dll
    strcpy(load_dll_path, GLB_DllPath);
    strcat(load_dll_path, DLL_NAME);
    GLB_dll_inst = LoadLibrary(load_dll_path);
    if (GLB_dll_inst == NULL)
    {
        userlog(LOG_ERR, "LoadLibrary()
failed GetLastError=%d", GetLastError());
        userlog(LOG_ERR, "DLL Path [%s]",
load_dll_path);
        return 1;
    }
    // Get calling function entry point
    ExSvrAplCreate = (void*(FAR*))
GetProcAddress((HMODULE)GLB_dll_inst,
"ExSvrAplCreate");
}

```

```

        if (ExSvrAplCreate == NULL) {
            userlog(LOG_ERR,
"GetProcAddress() fail <ExSvrAplCreate>
GetLastError=%d",
GetLastError());
        }
        ExSvrAplDelete = (void(FAR*)(void*
object_pointer))

        GetProcAddress((HMODULE)GLB_dll_inst,
"ExSvrAplDelete");
        if (ExSvrAplDelete == NULL) {
            userlog(LOG_ERR,
"GetProcAddress() fail <ExSvrAplDelete>
GetLastError=%d",
GetLastError());
        }
        ExSvrAplInitiate = (void(FAR*)(void*
object_pointer))

        GetProcAddress((HMODULE)GLB_dll_inst,
"ExSvrAplInitiate");
        if (ExSvrAplInitiate == NULL) {
            userlog(LOG_ERR,
"GetProcAddress() fail <ExSvrAplInitiate>
GetLastError=%d",
GetLastError());
        }
        ExSvrAplTerminate = (void(FAR*)(void*
object_pointer))

        GetProcAddress((HMODULE)GLB_dll_inst,
"ExSvrAplTerminate");
        if (ExSvrAplTerminate == NULL) {
            userlog(LOG_ERR,
"GetProcAddress() fail <ExSvrAplTerminate>
GetLastError=%d",
GetLastError());
        }
        ExTrxNewOrder = (int(FAR*)(void*
new_order_struct, void* object_pointer))

        GetProcAddress((HMODULE)GLB_dll_inst,
"ExTrxNewOrder");
        if (ExTrxNewOrder == NULL) {
            userlog(LOG_ERR,
"GetProcAddress() fail <ExTrxNewOrder>
GetLastError=%d",
GetLastError());
        }
        ExTrxPayment = (int(FAR*)(void*
payment_struct, void* object_pointer))

        GetProcAddress((HMODULE)GLB_dll_inst,
"ExTrxPayment");
        if (ExTrxPayment == NULL) {
            userlog(LOG_ERR,
"GetProcAddress() fail <ExTrxPayment>
GetLastError=%d",
GetLastError());
        }
        ExTrxOrderStatus = (int(FAR*)(void*
order_status_struct, void* object_pointer))

```

```

GetProcAddress((HMODULE)GLB_dll_inst,
"ExTrxOrderStatus");
        if (ExTrxOrderStatus == NULL) {
            userlog(LOG_ERR,
"GetProcAddress() fail <ExTrxOrderStatus>
GetLastError=%d",
GetLastError());
        }
        ExTrxStockLevel = (int(FAR*)(void*
stock_level_struct, void* object_pointer))

        GetProcAddress((HMODULE)GLB_dll_inst,
"ExTrxStockLevel");
        if (ExTrxStockLevel == NULL) {
            userlog(LOG_ERR,
"GetProcAddress() fail <ExTrxStockLevel>
GetLastError=%d",
GetLastError());
        }
        ExTrxDelivery = (int(FAR*)(void*
delivery_struct, void* object_pointer))

        GetProcAddress((HMODULE)GLB_dll_inst,
"ExTrxDelivery");
        if (ExTrxDelivery == NULL) {
            userlog(LOG_ERR,
"GetProcAddress() fail <ExTrxDelivery>
GetLastError=%d",
GetLastError());
        }
        if (ExSvrAplCreate == 0 ||
            ExSvrAplDelete == 0 ||
            ExSvrAplInitiate == 0 ||
            ExSvrAplTerminate == 0 ||
            ExTrxNewOrder == 0 ||
            ExTrxPayment == 0 ||
            ExTrxOrderStatus == 0 ||
            ExTrxStockLevel == 0 ||
            ExTrxDelivery == 0)
            return 2;
        /* Normal end */
        return 0;
    }
}
#endif
.....
SvrAplBase/ProcCntl.h
.....
*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Control process informations
*
*
* CREATE by TSL 2003.02.19
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *

```

```

*****
**/
#define STAT_FILE_NAME
    "SvrAplBase_status.dat"
#ifdef PROC_MODE //!!!! Executing mode is
PRCODESS. //!!!!
/* Mutex lock information */
#define PROC_CNTL_MUTEX_NAME
    "TPCC-PROC-CNTL-MUTEX"
/* Process cntrol information */
typedef struct _proc_cntl_info {
    unsigned long process_status; /*
Process status */
#define PROC_STAT_IDLE 0
    /* Process idle */
#define PROC_STAT_BUSY 1
    /* Process busy */
#define PROC_STAT_TERM 2
    /* Process Terminate */
    HANDLE command_event;
    /* Command sending event*/
    HANDLE response_event;
    /* Response sending event*/
    void* shared_memory; /*
Shared memory pointer*/
    HANDLE shared_mapping;
    /* Mapping handle */
    int new_struct_size; /*
New Order data size */
    int pay_struct_size; /*
Payment data size */
    int odr_struct_size; /*
Orderstatus data size*/
    int del_struct_size; /*
Delivery data size */
    int sto_struct_size; /*
Stock Level data size*/
    /* Executing transactions */
    int exec_new; /*
Number of NewOrder executing */
    int exec_pay; /*
Number of Payment executing */
    int exec_sto; /*
Number of StockLevel executing */
    int exec_odr; /*
Number of OrderStatus executing */
    int exec_del; /*
Number of Delivery Executing */
} PROC_CNTL_INFO;
#define PROC_CNTL_TBL_SHARED MEM
    "ProcCntlTblSharedMem"
#define PROC_CNTL_IDLE_EVENT
    "TPCC_ProcCntlIdleEvent"
/* Global functions */
bool CreateCntlProcess();
void TermSvrAplCntl();
void GetCntlProcessNum();
int RequestProc(int command, void*
req_data, unsigned long req_len,
PROC_CNTL_INFO* proc_cntl_ifo);
int WaitProcessEnd(void* proc_result,
unsigned long proc_len, PROC_CNTL_INFO*
proc_cntl_ifo);
#else //!!!! Executing mode is
THREAD. //!!!!
/* Object cntrol information */
typedef struct _object_cntl_info {

```

```

int    object_status;
/* Object control status */
#define OBJ_STAT_IDLE    0
/* Object IDLE */
#define OBJ_STAT_BUSY    1
/* Object BUSY */

void*  object_pointer; /*
SvrApl object pointer */

/* Executing transactions */
int    exec_new;
/* Number of NewOrder executing */
int    exec_pay;
/* Number of Payment executing */
int    exec_sto;
/* Number of StockLevel executing */
int    exec_odr;
/* Number of OrderStatus executing */
int    exec_del;
/* Number of Delivery Executing */

/* Added T.Kato 03.07.29 Research information
add */
int    transaction_stat; /*
Transaction execution status */
#define STAT_REQUEST_NEW    1
/* Reauest NewOrder */
#define STAT_REQUEST_PAY    2
/* Reauest Payment */
#define STAT_REQUEST_ODR    3
/* Reauest OrderStatus */

#define STAT_REQUEST_DEL    4
/* Reauest Delivery */
#define STAT_REQUEST_STO    5
/* Reauest StockLevel */
#define STAT_RESPONSE    0
/* Turn to response */

} OBJECT_CNTL_INFO;

/* Event name */
#define OBJECT_CNTL_IDLE_EVENT
"TPCC_ObjectCntlIdleEvent"

/* Global functions */
int
InitObjectiveSvrApl();
OBJECT_CNTL_INFO* GetIdleSvrAplObject();
int
ReleaseObjectCntlInfo(OBJECT_CNTL_IN
FO* object_cntl_info);
void    TermObjectiveSvrApl();
int    LoadSvrAplDll();
void    GetCntlProcessNum();
#endif
.....
SvrAplBase/RequestProc.cpp
.....
/*
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* 1) GetDBAccessInfo : Get DB access process
information */

```

```

* 2) RequestProc : Request to control
process
*
* CREATE by TSL 2002.12.18
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****
**/
#include "stdafx.h"
#include "SvrAplBase.h"
#include "ProcCntl.h"
#include "SvrApl.h"
#include "log.h"
#include "MapFileInfo.h"
#include "GlobalArea.h"

#ifdef PROC_MODE /////// Executing mode is
PRCODESS. ///////
/*
* Get DB access process information.
*
* If idle process nothing, then wait until process
idle.
* Return Value
* !0 : Normal end (control table pointer)
* 0 : Abnormal end
*****
**/
PROC_CNTL_INFO*
CSvrApl::GetDBAccessInfo() {
    DWORD ret_value;
    PROC_CNTL_INFO* proc_cntl_info = 0;
    int i;
    MAC_PutFncEntryLog("CSvrApl::GetDBAcc
essInfo");
    /* Lock process control table, while finding
idle process */
    EnterCriticalSection(&GLB_ProcCntlCritical)
;

    while(1) {
        /* Search process control table for
finding idle process */
        proc_cntl_info = GLB_ProcCntlInfo;

        for(i = 0; i <
GLB_NumControlProcesses; i++) {
            if (proc_cntl_info->
process_status == PROC_STAT_IDLE) {
                /* Found idle process */
                break;
            }
            proc_cntl_info++;
        }
        if (i != GLB_NumControlProcesses) {
            /* Find idle process */
            break;
        }
        /* All process busy, waiting any
process idle */

```

```

        if ((ret_value =
WaitForSingleObject(GLB_ProcIdleEvent,
INFINITE)) != WAIT_OBJECT_0) {
            userlog(LOG_ERR, "Illegal
process idle waiting (WaitForSingleObject()
ret_val=%d GetLastError=%d)",
ret_value, GetLastError());
            return 0;
        }
    }
    /* Find idle process, then set process busy
and unlock process control table */
    proc_cntl_info->process_status =
PROC_STAT_BUSY;
    LeaveCriticalSection(&GLB_ProcCntlCritical
);
    return proc_cntl_info;
}
/*
*****
* Release DB access process information.
*
* Return Value
* 0 : Normal end
* !0 : Abnormal end
*****
**/
int
CSvrApl::ReleaseDBAccessInfo(PROC_CNTL_IN
FO* proc_cntl_info) {
    proc_cntl_info->process_status =
PROC_STAT_IDLE;
    if (!SetEvent(GLB_ProcIdleEvent)) {
        userlog(LOG_ERR, "SetEvent() for
posting process idle GetLastError=%d",
GetLastError());
        return -1;
    }
    return 0;
}
/*
*****
* Request DB access to control process
*
* Return Value
* 0 : Normal end
* !0 : Abnormal end
*****
**/
int RequestProc(int command, void* req_data,
unsigned long req_len, PROC_CNTL_INFO*
proc_cntl_info) {
    MAC_PutFncEntryLog("RequestProc");
    /* Set Command code */
    *(int*)(proc_cntl_info->shared_memory) =
command;
    /* Set Processing data */
    if (req_data != 0) {
        /* set processing data to shared
memory */
        memcpy((char*)(proc_cntl_info->
shared_memory) + MAP_DATA_OFFSET,
req_data, req_len);
    }

    /* Post request to control process */

```

```

        if (!SetEvent(proc_cntl_info->command_event)) {
            userlog(LOG_ERR, "SetEvent() for
request command executing fail
GetLastError=%d",
                GetLastError());
            return -1;
        }
        return 0;
    }
}

#endif

.....
SvrAplBase/StdAfx.cpp
.....
// stdafx.cpp :
// stdafx.pch
// stdafx.obj

#include "stdafx.h"

#ifdef _ATL_STATIC_REGISTRY
#include <statreg.h>
#include <statreg.cpp>
#endif

#include <atimpl.cpp>

.....
SvrAplBase/StdAfx.h
.....
// stdafx.h :
//
//

#ifdef AFX_STDAFX_H_9C25EB91_5B09_47E8_ACFD_97B60ADF13D0__INCLUDED_
#define
AFX_STDAFX_H_9C25EB91_5B09_47E8_ACFD_97B60ADF13D0__INCLUDED_
#endif

#ifdef _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
#define STRICT
#ifdef _WIN32_WINNT
#define _WIN32_WINNT 0x0400
#endif
#define _ATL_APARTMENT_THREADED
#include <atlbase.h>
// CComModule
// _Module
extern CComModule _Module;
#include <atcom.h>
//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++
#endif
// !defined(AFX_STDAFX_H_9C25EB91_5B09_47E8_ACFD_97B60ADF13D0__INCLUDED)
.....
SvrAplBase/SvrApl.cpp
.....
// SvrApl.cpp : CSvrApl
#include "stdafx.h"
#include <mtx.h>
#include "SvrAplBase.h"

```

```

#include "ProcCntl.h"
#include "SvrApl.h"
#include "SvrAplErrorCode.h"
#include "log.h"
#include "ProcCntlCmd.h"
#include "SvrAplErrorcode.h"
#include "GlobalArea.h"
// C SvrApl

/*****
* Constructor of CSvrApl Class
*
* This function execute initializing module.
*****/

CSvrApl::CSvrApl()
{
    userlog(LOG_INF, "CSvrApl constructor
executing, nothin to do");
}

/*****
* Destructor of CSvrApl Class
*
* This function call terminating module.
*****/

CSvrApl::~CSvrApl()
{
    //
    // /* Terminate control process */
    // /* TermSvrAplCntl(); Done
CSvrApl::TermAvrApl() function */
    // /*
    Nothing to do in this function */
}

/*****
* New Order Transaction Executing
*
* This function call New Order transaction
executing module.
*****/

STDMETHODIMP CSvrApl::ExNewOrder(int
new_order_struct)
{
    // TODO:
#ifdef PROC_MODE // Executing mode is
PRCODESS. //
    PROC_CNTL_INFO* proc_cntl_info;
    int ret_code;

    MAC_PutFncEntryLog("CSvrApl::ExNewOr
der");
    if (GLB_InitComplete != INIT_COMPLETE) {
        userlog(LOG_ERR, "New Order not
execute, because of initialize incomplete");
        return ERR_INIT_INCOMPLETE;
    }
    proc_cntl_info = GetDBAccessInfo();

```

```

    ret_code =
RequestProc(COMMAND_NEWORDER,
(void*)new_order_struct,

    proc_cntl_info->new_struct_size,
proc_cntl_info);
    if (ret_code != 0) {
        userlog(LOG_ERR, "New Order
Processing request fail");
        return ERR_NO_EXEC;
    }
    ret_code =
WaitProcessEnd((void*)new_order_struct,

    proc_cntl_info->new_struct_size,
proc_cntl_info);
    if (ret_code != 0) {
        userlog(LOG_ERR, "New Order
Processing wait fail");
        return ERR_CNTL;
    }
    if (*(int*)proc_cntl_info->shared_memory !=
0) {
        /* Transaction process error */
        userlog(LOG_ERR, "New Order
processing error end ret_code = %d",

            *(int*)proc_cntl_info->shared_memory);
        return ERR_TRANS_PROC;
    }
    if (ReleaseDBAccessInfo(proc_cntl_info) !=
0) {
        userlog(LOG_ERR, "New Order
release process DB access information");
        return ERR_CNTL;
    }
    return S_OK;
}
#else // Executing mode is
THREAD. //
int ret_code;
OBJECT_CNTL_INFO* object_cntl_info;
MAC_SampleWork;
MAC_PutFncEntryLog("CSvrApl::ExNewOr
der");
MAC_SampleDataSet(*new_order_struct);
if (GLB_InitComplete != INIT_COMPLETE) {
    userlog(LOG_ERR, "New Order not
execute, because of initialize incomplete");
    return ERR_INIT_INCOMPLETE;
}

    MAC_SampleStartTime;
    if ((object_cntl_info = GetIdleSvrAplObject())
== 0) {
        userlog(LOG_ERR,
"ExTrxNewOrder() fail to get the object control
information");
        return ERR_NO_EXEC;
    }
    MAC_SampleGlobalSet(obj_sample_data);

    object_cntl_info->transaction_stat =
STAT_REQUEST_NEW;
    MAC_SampleStartTime;
    ret_code =
ExTrxNewOrder((void*)new_order_struct,
object_cntl_info->object_pointer);
    if (ret_code != 0) {

```

```

        userlog(LOG_ERR,
"ExTrxNewOrder() error return rc=%d", ret_code);
    }
    MAC_SampleGlobalSet(new_sample_data)
;
    object_cntl_info->transaction_stat =
STAT_RESPONSE;

    object_cntl_info->exec_new++;
    if
(RelaseObjectCntlInfo(object_cntl_info) != 0) {
        userlog(LOG_ERR,
"ExTrxNewOrder() fail to release the object control
information");
        return ERR_CNTL;
    }

/* Added T.Kato 03.07.25 Error information can't
post to RTE */
    if (ret_code != 0)
        return ERR_TRANS_PROC;
/* Added end */

    return S_OK;
#endif
}

/*****
*****
*   Payment Transaction Executing
*
*   This function call Payment transaction
executing module.
*****
******/
STDMETHODIMP CSvrApl::ExPayment(int
*payment_struct)
{
    // TODO:

#ifdef PROC_MODE    //!!!!!! executing mode is
PRCODESS. //!!!!

    PROC_CNTL_INFO*   proc_cntl_info;
    int                ret_code;
    MAC_PutFuncEntryLog("CSvrApl::ExPayme
nt");
    if (GLB_InitComplete != INIT_COMPLETE) {
        userlog(LOG_ERR, "Payment not
execute, because of initialize incomplete");
        return ERR_INIT_INCOMPLETE;
    }
    proc_cntl_info = GetDBAccessInfo();
    ret_code =
RequestProc(COMMAND_PAYMENT,
(void)*payment_struct,

    proc_cntl_info->pay_struct_size,
proc_cntl_info);
    if (ret_code != 0) {
        userlog(LOG_ERR, "Payment
Processing request fail");
        return ERR_NO_EXEC;
    }
    ret_code =
WaitProcessEnd((void)*payment_struct,

```

```

    proc_cntl_info->pay_struct_size,
proc_cntl_info);
    if (ret_code != 0) {
        userlog(LOG_ERR, "Payment
Processing wait fail fail");
        return ERR_CNTL;
    }
    if (*(int*)proc_cntl_info->shared_memory !=
0) {
        /* Transaction process error */
        userlog(LOG_ERR, "Payment
processing error end ret_code = %d",

        *(int*)proc_cntl_info->shared_memory);
        return ERR_TRANS_PROC;
    }
    if (ReleaseDBAccessInfo(proc_cntl_info) !=
0) {
        userlog(LOG_ERR, "Payment release
process DB access information");
        return ERR_CNTL;
    }
    return S_OK;
//!!!!!! Executing mode is
THREAD. //!!!!
OBJECT_CNTL_INFO* object_cntl_info;
int                ret_code;
MAC_SampleWork;
MAC_PutFuncEntryLog("CSvrApl::ExPayme
nt");
    if (GLB_InitComplete != INIT_COMPLETE) {
        userlog(LOG_ERR, "Payment not
execute, because of initialize incomplete");
        return ERR_INIT_INCOMPLETE;
    }
    MAC_SampleStartTime;
    if ((object_cntl_info = GetIdleSvrAplObject())
== 0) {
        userlog(LOG_ERR, "ExTrxPayment()
fail to get the object control information");
        return ERR_NO_EXEC;
    }
    MAC_SampleGlobalSet(obj_sample_data);
    object_cntl_info->transaction_stat =
STAT_REQUEST_PAY;
    MAC_SampleStartTime;
    ret_code =
ExTrxPayment((void)*payment_struct,
object_cntl_info->object_pointer);
    if (ret_code != 0) {
        userlog(LOG_ERR, "ExTrxPayment()
error return rc=%d", ret_code);
    }
    MAC_SampleGlobalSet(pay_sample_data);
    object_cntl_info->transaction_stat =
STAT_RESPONSE;
    object_cntl_info->exec_pay++;
    if
(RelaseObjectCntlInfo(object_cntl_info) != 0) {
        userlog(LOG_ERR, "ExTrxPayment()
fail to release the object control information");
        return ERR_CNTL;
    }
/* Added T.Kato 03.07.25 Error information can't
post to RTE */
    if (ret_code != 0)
        return ERR_TRANS_PROC;
/* Added end */

```

```

    return S_OK;
#endif
}

/*****
*****
*   Order Status Transaction Executing
*
*   This function call Order Status transaction
executing module.
*****
******/
STDMETHODIMP CSvrApl::ExOrderStatus(int
*order_status_struct)
{
    // TODO:
#ifdef PROC_MODE    //!!!!!! Executing mode is
PRCODESS. //!!!!

    PROC_CNTL_INFO*   proc_cntl_info;
    int                ret_code;
    MAC_PutFuncEntryLog("CSvrApl::ExOrderSt
atus");
    if (GLB_InitComplete != INIT_COMPLETE) {
        userlog(LOG_ERR, "Order Status not
execute, because of initialize incomplete");
        return ERR_INIT_INCOMPLETE;
    }
    proc_cntl_info = GetDBAccessInfo();
    ret_code =
RequestProc(COMMAND_ORDERSTATUS,
(void)*order_status_struct,

    proc_cntl_info->odr_struct_size,
proc_cntl_info);
    if (ret_code != 0) {
        userlog(LOG_ERR, "Order Status
Processing request fail");
        return ERR_NO_EXEC;
    }

    ret_code =
WaitProcessEnd((void)*order_status_struct,

    proc_cntl_info->odr_struct_size,
proc_cntl_info);
    if (ret_code != 0) {
        userlog(LOG_ERR, "Order Status
Processing wait fail fail");
        return ERR_CNTL;
    }

    if (*(int*)proc_cntl_info->shared_memory !=
0) {
        /* Transaction process error */
        userlog(LOG_ERR, "Order Status
processing error end ret_code = %d",

        *(int*)proc_cntl_info->shared_memory);
        return ERR_TRANS_PROC;
    }
    if (ReleaseDBAccessInfo(proc_cntl_info) !=
0) {
        userlog(LOG_ERR, "Order Status
release process DB access information");
        return ERR_CNTL;
    }
}

```

```

return S_OK;

#else          /////// Executing mode is
THREAD. ///////

OBJECT_CNTL_INFO* object_cntl_info;
int      ret_code;

MAC_SampleWork;

MAC_PutFncEntryLog("CSvrApl::ExOrderSt
atus");

if (GLB_InitComplete != INIT_COMPLETE) {
    userlog(LOG_ERR, "Order Status not
execute, because of initialize incomplete");
    return ERR_INIT_INCOMPLETE;
}

MAC_SampleStartTime;
if ((object_cntl_info = GetIdleSvrAplObject())
== 0) {
    userlog(LOG_ERR,
"ExTrxOrderStatus() fail to get the object control
information");
    return ERR_NO_EXEC;
}
MAC_SampleGlobalSet(obj_sample_data);

object_cntl_info->transaction_stat =
STAT_REQUEST_ODR;
MAC_SampleStartTime;
ret_code =
ExTrxOrderStatus((void*)"order_status_struct,
object_cntl_info->object_pointer);
if (ret_code != 0) {
    userlog(LOG_ERR,
"ExTrxOrderStatus() error return rc=%d",
ret_code);
}
MAC_SampleGlobalSet(odr_sample_data);
object_cntl_info->transaction_stat =
STAT_RESPONSE;
object_cntl_info->exec_odr++;
if
(RelaseObjectCntlInfo(object_cntl_info) != 0) {
    userlog(LOG_ERR,
"ExTrxOrderStatus() fail to release the object
control information");
    return ERR_CNTL;
}
/* Added T.Kato 03.07.25 Error information can't
post to RTE */
if (ret_code != 0)
    return ERR_TRANS_PROC;
/* Added end */

return S_OK;
#endif
}
/*****
* Stock Level Transaction Executing
*
* This function call Stock Level transaction
executing module.
*****/

```

```

STDMETHODIMP CSvrApl::ExStockLevel(int
*stock_level_struct)
{
    // TODO:
#ifdef PROC_MODE /////// Executing mode is
PRCODESS. ///////
    PROC_CNTL_INFO* proc_cntl_info;
    int      ret_code;
    MAC_PutFncEntryLog("CSvrApl::ExStockL
evel");
    if (GLB_InitComplete != INIT_COMPLETE) {
        userlog(LOG_ERR, "Stock Level not
execute, because of initialize incomplete");
        return ERR_INIT_INCOMPLETE;
    }
    proc_cntl_info = GetDBAccessInfo();
    ret_code =
RequestProc(COMMAND_STOCKLEVEL,
(void*)"stock_level_struct,

    proc_cntl_info->sto_struct_size,
proc_cntl_info);
    if (ret_code != 0) {
        userlog(LOG_ERR, "Stock Level
Processing request fail");
        return ERR_NO_EXEC;
    }
    ret_code =
WaitProcessEnd((void*)"stock_level_struct,

    proc_cntl_info->sto_struct_size,
proc_cntl_info);
    if (ret_code != 0) {
        userlog(LOG_ERR, "Stock Level
Processing wait fail fail");
        return ERR_CNTL;
    }
    if (*(int*)proc_cntl_info->shared_memory !=
0) {
        /* Transaction process error */
        userlog(LOG_ERR, "Stock Level
processing error end ret_code = %d",

        *(int*)proc_cntl_info->shared_memory);
        return ERR_TRANS_PROC;
    }
    if (RelaseDBAccessInfo(proc_cntl_info) !=
0) {
        userlog(LOG_ERR, "Stock Level
release process DB access information");
        return ERR_CNTL;
    }
    return S_OK;
#else          /////// Executing mode is
THREAD. ///////
    OBJECT_CNTL_INFO* object_cntl_info;
    int      ret_code;
    MAC_SampleWork;
    MAC_PutFncEntryLog("CSvrApl::ExStockL
evel");
    if (GLB_InitComplete != INIT_COMPLETE) {
        userlog(LOG_ERR, "Stock Level not
execute, because of initialize incomplete");
        return ERR_INIT_INCOMPLETE;
    }
    MAC_SampleStartTime;
    if ((object_cntl_info = GetIdleSvrAplObject())
== 0) {

```

```

        userlog(LOG_ERR,
"ExTrxStockLevel() fail to get the object control
information");
        return ERR_NO_EXEC;
    }
    MAC_SampleGlobalSet(obj_sample_data);

    object_cntl_info->transaction_stat =
STAT_REQUEST_STO;
    MAC_SampleStartTime;
    ret_code =
ExTrxStockLevel((void*)"stock_level_struct,
object_cntl_info->object_pointer);
    if (ret_code != 0) {
        userlog(LOG_ERR,
"ExTrxStockLevel() error return rc=%d", ret_code);
    }
    MAC_SampleGlobalSet(sto_sample_data);
    object_cntl_info->transaction_stat =
STAT_RESPONSE;

    object_cntl_info->exec_sto++;
    if
(RelaseObjectCntlInfo(object_cntl_info) != 0) {
        userlog(LOG_ERR,
"ExTrxStockLevel() fail to release the object
control information");
        return ERR_CNTL;
    }
    /* Added T.Kato 03.07.25 Error information can't
post to RTE */
    if (ret_code != 0)
        return ERR_TRANS_PROC;
    /* Added end */

    return S_OK;
#endif
}
/*****
* Delivery Transaction Executing
*
* This function call Delivery transaction
executing module.
*****/
STDMETHODIMP CSvrApl::ExDelivery(int
*delivery_struct)
{
    // TODO:
#ifdef PROC_MODE /////// Executing mode is
PRCODESS. ///////
    PROC_CNTL_INFO* proc_cntl_info;
    int      ret_code;

    MAC_PutFncEntryLog("CSvrApl::ExDeliver
y");

    if (GLB_InitComplete != INIT_COMPLETE) {
        userlog(LOG_ERR, "Delivery not
execute, because of initialize incomplete");
        return ERR_INIT_INCOMPLETE;
    }

```

```

proc_cntl_info = GetDBAccessInfo();

ret_code =
RequestProc(COMMAND_DELIVERY,
(void*)"delivery_struct,

proc_cntl_info->del_struct_size,
proc_cntl_info);
if (ret_code != 0) {
userlog(LOG_ERR, "Stock Level
Processing request fail");
return ERR_NO_EXEC;
}

ret_code =
WaitProcessEnd((void*)"delivery_struct,

proc_cntl_info->del_struct_size,
proc_cntl_info);
if (ret_code != 0) {
userlog(LOG_ERR, "Delivery
Processing wait fail fail");
return ERR_CNTL;
}

if (*(int*)proc_cntl_info->shared_memory !=
0) {
/* Transaction process error */
userlog(LOG_ERR, "Delivery
processing error end ret_code = %d",

*(int*)proc_cntl_info->shared_memory);
return ERR_TRANS_PROC;
}

if (ReleaseDBAccessInfo(proc_cntl_info) !=
0) {
userlog(LOG_ERR, "Delevery release
process DB access information");
return ERR_CNTL;
}
return S_OK;
#else // Executing mode is
THREAD. //
OBJECT_CNTL_INFO* object_cntl_info;
int ret_code;
MAC_SampleWork;
MAC_PutFncEntryLog("CSvrApl::ExDeliver
y");
if (GLB_InitComplete != INIT_COMPLETE) {
userlog(LOG_ERR, "Delivery not
execute, because of initialize incomplete");
return ERR_INIT_INCOMPLETE;
}
MAC_SampleStartTime;
if ((object_cntl_info = GetIdleSvrAplObject())
== 0) {
userlog(LOG_ERR, "ExTrxDelivery()
fail to get the object control information");
return ERR_NO_EXEC;
}
MAC_SampleGlobalSet(obj_sample_data);

object_cntl_info->transaction_stat =
STAT_REQUEST_DEL;
MAC_SampleStartTime;

```

```

ret_code =
ExTrxDelivery((void*)"delivery_struct,
object_cntl_info->object_pointer);
if (ret_code != 0) {
userlog(LOG_ERR, "ExTrxDelivery()
error return rc=%d", ret_code);
}
MAC_SampleGlobalSet(del_sample_data);
object_cntl_info->transaction_stat =
STAT_RESPONSE;
object_cntl_info->exec_del++;
if
(RelaseObjectCntlInfo(object_cntl_info) != 0) {
userlog(LOG_ERR, "ExTrxDelivery()
fail to release the object control information");
return ERR_CNTL;
}
/* Added T.Kato 03.07.25 Error information can't
post to RTE */
if (ret_code != 0)
return ERR_TRANS_PROC;
/* Added end */

return S_OK;
#endif
}
/* *****
* Object executing with in the scope of
transaction.
* *****
*/
STDMETHODIMP CSvrApl::EntryTrans()
{
// TODO:
MAC_PutFncEntryLog("CSvrApl::EntryTran
s");
IObjectContext* pIObject = NULL;
HRESULT hr =
GetObjectContext(&pIObject);
pIObject->SetComplete();
pIObject->Release();
pIObject = NULL;
return hr;
}
/* *****
* Terminate SvrApl
* *****
*/
STDMETHODIMP CSvrApl::TermSvrApl()
{
// TODO:
/* Terminate SvrApl */
#ifdef PROC_MODE // Executing mode is
PRCODESS. //
MAC_PutFncEntryLog("CSvrApl::TermSvrA
pl");
userlog(LOG_INF, "Request of terminate
SvrApl accept & execute");
TermSvrAplCntl();
return S_OK;
#else // Executing mode is
THREAD. //

```

```

MAC_PutFncEntryLog("CSvrApl::TermSvrA
pl");
userlog(LOG_INF, "Request of terminate
SvrApl accept & execute");
TermObjectiveSvrApl();
return S_OK;
#endif
}
/* *****
* Initialize SvrApl
* *****
*/
STDMETHODIMP CSvrApl::InitSvrApl()
{
// TODO:
/* Initialize SvrApl */
#ifdef PROC_MODE // Executing mode is
PRCODESS. //
MAC_PutFncEntryLog("CSvrApl::InitSvrApl"
);
userlog(LOG_INF, "Request of initialize
SvrApl accept & execute");
#endif
#if 0
! /* Initialize control process and control
information */
! InitSvrAplCntl();
#endif
return S_OK;
#else // Executing mode is
THREAD. //
MAC_PutFncEntryLog("CSvrApl::InitSvrApl"
);
userlog(LOG_INF, "Request of initialize
SvrApl (Nothing to do)");
return S_OK;
#endif
}
/* *****
SvrAplBase/SvrApl.h
*****
// SvrApl.h : CSvrApl
*/
#ifdef __SVR_APL_H_
#define __SVR_APL_H_

#include "resource.h" //
#include <mtx.h>

// *****
// CSvrApl
class ATL_NO_VTABLE CSvrApl :
public
CComObjectRootEx<CComMultiThreadModel>,
public CComCoClass<CSvrApl,
&CLSID_SvrApl>,

```

```

    public ISvrApl
//    public ISvrApl,
//    public IObjectControl

{
public:
    CSvrApl() {};

#ifdef PROC_MODE    // Executing mode is
PRCODESS. //
/* Functions */
PROC_CNTL_INFO* GetDBAccessInfo();
int
    ReleaseDBAccessInfo(PROC_CNTL_INFO
* proc_cntl_info);

#else                // Executing mode is
THREAD. //
#endif

DECLARE_REGISTRY_RESOURCEID(IDR_SV
RAPL)

DECLARE_PROTECT_FINAL_CONSTRUCT()

BEGIN_COM_MAP(CSvrApl)
    COM_INTERFACE_ENTRY(ISvrApl)
//    COM_INTERFACE_ENTRY(IObjectControl)

END_COM_MAP()

//public:
//    STDMETHODCALLTYPE { return S_OK; }
//    STDMETHODCALLTYPE (BOOL, CanBePooled)
{ return TRUE; }
//    STDMETHODCALLTYPE (void, Deactivate) {}

// ISvrApl
public:
    STDMETHODCALLTYPE (InitSvrApl)();
    STDMETHODCALLTYPE (TermSvrApl)();
    STDMETHODCALLTYPE (EntryTrans)();
    STDMETHODCALLTYPE (ExDelivery)("[in,out]"/ int*
delivery_struct);
    STDMETHODCALLTYPE (ExStockLevel)("[in,out]"/ int*
stock_level_struct);
    STDMETHODCALLTYPE (ExOrderStatus)("[in,out]"/
int* order_status_struct);
    STDMETHODCALLTYPE (ExPayment)("[in,out]"/ int*
payment_struct);
    STDMETHODCALLTYPE (ExNewOrder)("[in,out]"/ int*
new_order_struct);
};
#endif // __SVRAPL_H
.....
SvrAplBase/SvrAplBase.cpp
.....
// SvrAplBase.cpp : DLL
// : Proxy/Stub
// proxy/stub DLL
// nmake -f SvrAplBaseps.mak
#include "stdafx.h"
#include "resource.h"
#include <initguid.h>
#include <io.h>
#include "SvrAplBase.h"
#include "SvrAplBase_i.c"
#include "ProcCntl.h"
#include "SvrApl.h"

```

```

#include "GlobalArea.h"
#include "log.h"
CComModule _Module;
BEGIN_OBJECT_MAP(ObjectMap)
OBJECT_ENTRY(CLSID_SvrApl, CSvrApl)
END_OBJECT_MAP()
// DLL
void GetOwnDllPath(HINSTANCE hInstance);
int GetModuleExecStat();
extern "C"
BOOL WINAPI DllMain(HINSTANCE hInstance,
DWORD dwReason, LPVOID /*lpReserved*/)
{
    if (dwReason == DLL_PROCESS_ATTACH)
    {
        _Module.Init(ObjectMap, hInstance,
&LIBID_SVRAPLBASELib);
        DisableThreadLibraryCalls(hInstance);
        GetOwnDllPath(hInstance);
        userlog(LOG_INF, "===
SvrAplBase.dll process attache path [%s] ===",
GLB_DllPath);
        if (GetModuleExecStat() == 0) return
TRUE;
#ifdef PROC_MODE    // Executing mode is
PRCODESS. //
        userlog(LOG_INF, "=== Executing
mode is PROCESS ===");
        GetCntlProcessNum();
        /* Initialize CritiicalSection for process
contoroling */
        InitializeCriticalSection(&GLB_ProcCntlCritica
l);
        /* Create DB Accessing process */
        if (!CreateCntlProcess()) {
            userlog(LOG_ERR, "Control
Process creating failed");
            return FALSE;
        }
        GLB_InitComplete =
INIT_COMPLETE;
#else                // Executing mode is
THREAD. //
        userlog(LOG_INF, "=== Executing
mode is THREAD ===");
        GetCntlProcessNum();
        /* Load DLL & Get process address
form DLL */
        if (LoadSvrAplDll() != 0) {
            userlog(LOG_ERR,
"SvrDBDepend.dll loading failed");
            return FALSE;
        }
        /* initialize object control */
        InitializeCriticalSection(&GLB_ObjectCntlCri
tical);
        InitializeCriticalSection(&GLB_ObjectCntlCri
tical_SetEvent);
        InitializeCriticalSection(&GLB_TransactionC
ritical);
        if (InitObjectiveSrvApl() != 0) {
            /* Initialize failed */
            userlog(LOG_ERR, "Initialize
object control failed");

```

```

        return FALSE;
    }
    GLB_InitComplete =
INIT_COMPLETE;
/* Added T.Kato 03.08.25 Sampling paformance
data */
    MAC_SampleInit;
/* Added end */
#endif
}
else if (dwReason ==
DLL_PROCESS_DETACH) {
    userlog(LOG_INF, "===
SvrAplBase.dll process detache ===");
    if (GetModuleExecStat() == 0) return
TRUE;

#ifdef PROC_MODE    // Executing mode is
PRCODESS. //
    /* Delete CriticalSection */
    DeleteCriticalSection(&GLB_ProcCntlCritica
l);
    TermSvrAplCntl();
#else                // Executing mode is
THREAD. //
    /* Delete CriticalSection */
    DeleteCriticalSection(&GLB_ObjectCntlCriti
cal);
    DeleteCriticalSection(&GLB_ObjectCntlCriti
cal_SetEvent);
    DeleteCriticalSection(&GLB_TransactionCriti
cal);

    /* Terminate SvrApl object */
    //TermObjectiveSvrApl();

    /* Unload DLL of DB depended */
    if (!FreeLibrary(GLB_dll_inst)) {
        userlog(LOG_ERR,
"FreeLibrary() failed");
    }
}
#endif
_Module.Term();
}
return TRUE; // ok
}

/* Get Own DLL path */
void GetOwnDllPath(HINSTANCE hInstance) {
    int i;

    GetModuleFileName(hInstance,
GLB_DllPath, sizeof(GLB_DllPath)-1);
    for(i = strlen(GLB_DllPath) - 1;
GLB_DllPath[i] != '\\'; i--);
    GLB_DllPath[i+1] = '\0';

    strcpy(GLB_LogFilePath, GLB_DllPath);
    strcat(GLB_LogFilePath,
LOG_FILE_NAME);
}

```

```

/* Get Module executing status */
int GetModuleExecStat() {
    char stat_file[MAX_PATH];

    strcpy(stat_file, GLB_DllPath);
    strcat(stat_file, STAT_FILE_NAME);

    if (_access(stat_file, 0x00) != -1)
        /* Exist status file */
        return 0;
    else
        /* No exist status file */
        return 1;
}

////////////////////////////////////////////////////
// DLL OLE

STDAPI DllCanUnloadNow(void)
{
    return (_Module.GetLockCount()==0) ? S_OK :
    S_FALSE;
}

////////////////////////////////////////////////////
//

STDAPI DllGetObject(REFCLSID rclsid,
REFIID riid, LPVOID*ppv)
{
    return _Module.GetObject(rclsid, riid,
    ppv);
}

////////////////////////////////////////////////////
// DllRegisterServer -

STDAPI DllRegisterServer(void)
{
    //
    return _Module.RegisterServer(TRUE);
}

////////////////////////////////////////////////////
// DllUnregisterServer -
STDAPI DllUnregisterServer(void)
{
    return _Module.UnregisterServer(TRUE);
}

:-----:
SvrAplBase/SvrAplBase.def
:-----:
; SvrAplBase.def :
LIBRARY "SvrAplBase.DLL"
EXPORTS
    DllCanUnloadNow @1 PRIVATE
    DllGetObject @2 PRIVATE
    DllRegisterServer @3 PRIVATE
    DllUnregisterServer @4 PRIVATE
:-----:
SvrAplBase/SvrAplBase.h
:-----:
/* this ALWAYS GENERATED file contains the
definitions for the interfaces */
/* File created by MIDL compiler version
5.01.0164 */
/* at Thu Sep 04 19:19:40 2003
*/
/* Compiler settings for
F:\SvrAplBase\SvrAplBase.idl:

    Oicf (OptLev=i2), W1, Zp8, env=Win32, ms_ext,
c_ext
    error checks: allocation ref bounds_check enum
stub_data
*/
//@@@MIDL_FILE_HEADERING( )
/* verify that the <rpcndr.h> version is high enough
to compile this file*/
#ifdef __REQUIRED_RPCNDR_H_VERSION__
#define
__REQUIRED_RPCNDR_H_VERSION__ 440
#endif
#include "rpc.h"
#include "rpcndr.h"
#ifdef __RPCNDR_H_VERSION__
#error this stub requires an updated version of
<rpcndr.h>
#endif // __RPCNDR_H_VERSION__
#ifdef COM_NO_WINDOWS_H
#include "windows.h"
#include "ole2.h"
#endif /*COM_NO_WINDOWS_H*/
#ifdef __SvrAplBase_h__
#define __SvrAplBase_h__
#endif /*__cplusplus*/
extern "C"{
#ifdef
/* Forward Declarations */
#ifdef __ISvrApl_FWD_DEFINED__
#define __ISvrApl_FWD_DEFINED__
typedef interface ISvrApl ISvrApl;
#ifdef __ISvrApl_FWD_DEFINED__
#ifdef __SvrApl_FWD_DEFINED__
#define __SvrApl_FWD_DEFINED__
#endif /*__cplusplus*/
typedef class SvrApl SvrApl;
#else
typedef struct SvrApl SvrApl;
#endif /* __cplusplus */
#ifdef __ISvrApl_FWD_DEFINED__
/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"
void __RPC_FAR * __RPC_USER
MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void
__RPC_FAR * );
#endif /* __ISvrApl_INTERFACE_DEFINED__
#define __ISvrApl_INTERFACE_DEFINED__
/* interface ISvrApl */
/* [unique][helpstring][uuid][object] */
EXTERN_C const IID IID_ISvrApl;

#if defined(__cplusplus)
&& Idefined(CINTERFACE)

    MIDL_INTERFACE("CB255680-B8B2-48AD-
996B-6D2E826B03F7")
    ISvrApl : public IUnknown
    {
    public:
        virtual /* [helpstring] */ HRESULT
STDMETHODCALLTYPE ExNewOrder(
        /* [out][in] */ int __RPC_FAR
*new_order_struct) = 0;

        virtual /* [helpstring] */ HRESULT
STDMETHODCALLTYPE ExPayment(
        /* [out][in] */ int __RPC_FAR
*payment_struct) = 0;

        virtual /* [helpstring] */ HRESULT
STDMETHODCALLTYPE ExOrderStatus(
        /* [out][in] */ int __RPC_FAR
*order_status_struct) = 0;

        virtual /* [helpstring] */ HRESULT
STDMETHODCALLTYPE ExStockLevel(
        /* [out][in] */ int __RPC_FAR
*stock_level_struct) = 0;

        virtual /* [helpstring] */ HRESULT
STDMETHODCALLTYPE ExDelivery(
        /* [out][in] */ int __RPC_FAR
*delivery_struct) = 0;

        virtual /* [helpstring] */ HRESULT
STDMETHODCALLTYPE EntryTrans( void ) = 0;

        virtual /* [helpstring] */ HRESULT
STDMETHODCALLTYPE TermSvrApl( void ) = 0;

        virtual /* [helpstring] */ HRESULT
STDMETHODCALLTYPE InitSvrApl( void ) = 0;
};

#else /* C style interface */

typedef struct ISvrAplVtbl
{
    BEGIN_INTERFACE

        HRESULT ( STDMETHODCALLTYPE
__RPC_FAR *QueryInterface )(
        ISvrApl __RPC_FAR * This,
        /* [in] */ REFIID riid,
        /* [iid_is][out] */ void __RPC_FAR
*__RPC_FAR *ppvObject);

        ULONG ( STDMETHODCALLTYPE
__RPC_FAR *AddRef )(
        ISvrApl __RPC_FAR * This);

        ULONG ( STDMETHODCALLTYPE
__RPC_FAR *Release )(
        ISvrApl __RPC_FAR * This);

        /* [helpstring] */ HRESULT
( STDMETHODCALLTYPE __RPC_FAR
*ExNewOrder )(
        ISvrApl __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR
*new_order_struct);

        /* [helpstring] */ HRESULT
( STDMETHODCALLTYPE __RPC_FAR
*ExPayment )(
        ISvrApl __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR
*payment_struct);

        /* [helpstring] */ HRESULT
( STDMETHODCALLTYPE __RPC_FAR
*ExOrderStatus )(
        ISvrApl __RPC_FAR * This,

```

```

    /* [out][in] */ int __RPC_FAR
    *order_status_struct);

    /* [helpstring] */ HRESULT
    ( STDMETHODCALLTYPE __RPC_FAR
    *ExStockLevel )(
        ISvrApl __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR
        *stock_level_struct);

    /* [helpstring] */ HRESULT
    ( STDMETHODCALLTYPE __RPC_FAR
    *ExDelivery )(
        ISvrApl __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR
        *delivery_struct);

    /* [helpstring] */ HRESULT
    ( STDMETHODCALLTYPE __RPC_FAR
    *EntryTrans )(
        ISvrApl __RPC_FAR * This);

    /* [helpstring] */ HRESULT
    ( STDMETHODCALLTYPE __RPC_FAR
    *TermSvrApl )(
        ISvrApl __RPC_FAR * This);

    /* [helpstring] */ HRESULT
    ( STDMETHODCALLTYPE __RPC_FAR
    *InitSvrApl )(
        ISvrApl __RPC_FAR * This);

    END_INTERFACE
    } ISvrAplVtbl;
    interface ISvrApl
    {
        CONST_VTBL struct ISvrAplVtbl
        __RPC_FAR *lpVtbl;
    };
    #ifdef COBJMACROS
    define ISvrApl_QueryInterface(This,riid,ppvObject)
        \
        (This)->lpVtbl ->
        QueryInterface(This,riid,ppvObject)
    #define ISvrApl_AddRef(This) \
        (This)->lpVtbl -> AddRef(This)
    #define ISvrApl_Release(This) \
        (This)->lpVtbl -> Release(This)
    #define
    ISvrApl_ExNewOrder(This,new_order_struct) \
        (This)->lpVtbl ->
        ExNewOrder(This,new_order_struct)
    #define ISvrApl_ExPayment(This,payment_struct)
        \
        (This)->lpVtbl ->
        ExPayment(This,payment_struct)
    #define
    ISvrApl_ExOrderStatus(This,order_status_struct)
        \
        (This)->lpVtbl ->
        ExOrderStatus(This,order_status_struct)
    #define
    ISvrApl_ExStockLevel(This,stock_level_struct) \
        (This)->lpVtbl ->
        ExStockLevel(This,stock_level_struct)
    #define ISvrApl_ExDelivery(This,delivery_struct) \
        (This)->lpVtbl ->
        ExDelivery(This,delivery_struct)
    #define ISvrApl_EntryTrans(This) \

```

```

        (This)->lpVtbl -> EntryTrans(This)
    #define ISvrApl_TermSvrApl(This) \
        (This)->lpVtbl -> TermSvrApl(This)
    #define ISvrApl_InitSvrApl(This) \
        (This)->lpVtbl -> InitSvrApl(This)
    #endif /* COBJMACROS */
    #endif /* C style interface */
    /* [helpstring] */ HRESULT
    STDMETHODCALLTYPE
    ISvrApl_ExNewOrder_Proxy(
        ISvrApl __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR
        *new_order_struct);
    void __RPC_STUB ISvrApl_ExNewOrder_Stub(
        IRpcStubBuffer *This,
        IRpcChannelBuffer *_pRpcChannelBuffer,
        PRPC_MESSAGE _pRpcMessage,
        DWORD *_pdwStubPhase);
    /* [helpstring] */ HRESULT
    STDMETHODCALLTYPE
    ISvrApl_ExPayment_Proxy(
        ISvrApl __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR *payment_struct);
    void __RPC_STUB ISvrApl_ExPayment_Stub(
        IRpcStubBuffer *This,
        IRpcChannelBuffer *_pRpcChannelBuffer,
        PRPC_MESSAGE _pRpcMessage,
        DWORD *_pdwStubPhase);
    /* [helpstring] */ HRESULT
    STDMETHODCALLTYPE
    ISvrApl_ExOrderStatus_Proxy(
        ISvrApl __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR
        *order_status_struct);
    void __RPC_STUB ISvrApl_ExOrderStatus_Stub(
        IRpcStubBuffer *This,
        IRpcChannelBuffer *_pRpcChannelBuffer,
        PRPC_MESSAGE _pRpcMessage,
        DWORD *_pdwStubPhase);
    /* [helpstring] */ HRESULT
    STDMETHODCALLTYPE
    ISvrApl_ExStockLevel_Proxy(
        ISvrApl __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR
        *stock_level_struct);

    void __RPC_STUB ISvrApl_ExStockLevel_Stub(
        IRpcStubBuffer *This,
        IRpcChannelBuffer *_pRpcChannelBuffer,
        PRPC_MESSAGE _pRpcMessage,
        DWORD *_pdwStubPhase);

    /* [helpstring] */ HRESULT
    STDMETHODCALLTYPE
    ISvrApl_ExDelivery_Proxy(
        ISvrApl __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR *delivery_struct);

    void __RPC_STUB ISvrApl_ExDelivery_Stub(
        IRpcStubBuffer *This,
        IRpcChannelBuffer *_pRpcChannelBuffer,
        PRPC_MESSAGE _pRpcMessage,
        DWORD *_pdwStubPhase);

```

```

    /* [helpstring] */ HRESULT
    STDMETHODCALLTYPE
    ISvrApl_EntryTrans_Proxy(
        ISvrApl __RPC_FAR * This);

    void __RPC_STUB ISvrApl_EntryTrans_Stub(
        IRpcStubBuffer *This,
        IRpcChannelBuffer *_pRpcChannelBuffer,
        PRPC_MESSAGE _pRpcMessage,
        DWORD *_pdwStubPhase);

    /* [helpstring] */ HRESULT
    STDMETHODCALLTYPE
    ISvrApl_TermSvrApl_Proxy(
        ISvrApl __RPC_FAR * This);

    void __RPC_STUB ISvrApl_TermSvrApl_Stub(
        IRpcStubBuffer *This,
        IRpcChannelBuffer *_pRpcChannelBuffer,
        PRPC_MESSAGE _pRpcMessage,
        DWORD *_pdwStubPhase);

    /* [helpstring] */ HRESULT
    STDMETHODCALLTYPE
    ISvrApl_InitSvrApl_Proxy(
        ISvrApl __RPC_FAR * This);

    void __RPC_STUB ISvrApl_InitSvrApl_Stub(
        IRpcStubBuffer *This,
        IRpcChannelBuffer *_pRpcChannelBuffer,
        PRPC_MESSAGE _pRpcMessage,
        DWORD *_pdwStubPhase);

    #endif /*
    __ISvrApl_INTERFACE_DEFINED__ */

    #ifndef
    __SVRAPLBASELib_LIBRARY_DEFINED__
    #define
    __SVRAPLBASELib_LIBRARY_DEFINED__

    /* library SVRAPLBASELib */
    /* [helpstring][version][uuid] */

    EXTERN_C const IID LIBID_SVRAPLBASELib;

    EXTERN_C const CLSID CLSID_SvrApl;

    #ifdef __cplusplus

    class DECLSPEC_UUID("1924BC6E-7A8C-
    427D-81DA-E002F2894BE7")
    SvrApl;
    #endif
    #endif /*
    __SVRAPLBASELib_LIBRARY_DEFINED__ */

    /* Additional Prototypes for ALL interfaces */

```

```

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#ifdef

.....
SvrAplBase/SvrAplBase.idl
.....
// SvrAplBase.idl : SvrAplBase.dll IDL
//

// (SvrAplBase.tlb)
// MIDL

import "oidl.idl";
import "ocidl.idl";
{
    object,
    uuid(CB255680-B8B2-48AD-996B-
6D2E826B03F7),

        helpstring("ISvrApl Interface"),
        pointer_default(unique)
    }
    interface ISvrApl : IUnknown
    {
        [helpstring(" ExNewOrder")]
        HRESULT ExNewOrder([in,out] int*
new_order_struct);
        [helpstring(" ExPayment")] HRESULT
ExPayment([in,out] int* payment_struct);
        [helpstring(" ExOrderStatus")]
        HRESULT ExOrderStatus([in,out] int*
order_status_struct);
        [helpstring(" ExStockLevel")]
        HRESULT ExStockLevel([in,out] int*
stock_level_struct);
        [helpstring(" ExDelivery")] HRESULT
ExDelivery([in,out] int* delivery_struct);
        [helpstring(" EntryTrans")] HRESULT
EntryTrans();
        [helpstring(" TermSvrApl")] HRESULT
TermSvrApl();
        [helpstring(" InitSvrApl")] HRESULT
InitSvrApl();
    };
}
{
    uuid(3348A83E-DDB2-46AC-9FD7-
591CE59AE63A),
    version(1.0),
    helpstring("SvrAplBase 1.0 ")
}
library SVRAPLBASELib
{
    importlib("stdole32.tlb");
    importlib("stdole2.tlb");
    {
        uuid(1924BC6E-7A8C-427D-81DA-
E002F2894BE7),
        helpstring("SvrApl Class")
    }
    coclass SvrApl
    {
        [default] interface ISvrApl;
    };
};

```

```

.....
SvrAplBase/SvrAplBase.mak
.....
# Microsoft Developer Studio Generated NMAKE
File, Based on SvrAplBase.dsp
!!IF "$(CFG)" == ""
CFG=SvrAplBase - Win32 Debug
!MESSAGE SvrAplBase - Win32 Debug
!ENDIF
!!IF "$(CFG)" != "SvrAplBase - Win32 Debug" &&
"$(CFG)" != "SvrAplBase - Win32 Unicode
Debug" && "$(CFG)" != "SvrAplBase - Win32
Release MinSize" && "$(CFG)" != "SvrAplBase -
Win32 Release MinDependency" && "$(CFG)" !=
"SvrAplBase - Win32 Unicode Release MinSize"
&& "$(CFG)" != "SvrAplBase - Win32 Unicode
Release MinDependency"
!MESSAGE "$(CFG)"
!MESSAGE NMAKE
!MESSAGE :
!MESSAGE
!MESSAGE NMAKE /f "SvrAplBase.mak"
CFG="SvrAplBase - Win32 Debug"
!MESSAGE
!MESSAGE :
!MESSAGE
!MESSAGE "SvrAplBase - Win32 Debug" ("Win32
(x86) Dynamic-Link Library")
!MESSAGE "SvrAplBase - Win32 Unicode
Debug" ("Win32 (x86) Dynamic-Link Library")
!MESSAGE "SvrAplBase - Win32 Release
MinSize" ("Win32 (x86) Dynamic-Link Library")
!MESSAGE "SvrAplBase - Win32 Release
MinDependency" ("Win32 (x86) Dynamic-Link
Library")
!MESSAGE "SvrAplBase - Win32 Unicode
Release MinSize" ("Win32 (x86) Dynamic-Link
Library")
!MESSAGE "SvrAplBase - Win32 Unicode
Release MinDependency" ("Win32 (x86)
Dynamic-Link Library")
!MESSAGE
!ERROR
!ENDIF

!!IF "$(OS)" == "Windows_NT"
NULL=
!ELSE
NULL=nul
!ENDIF

!!IF "$(CFG)" == "SvrAplBase - Win32 Debug"

OUTDIR=.\\Debug
INTDIR=.\\Debug
# Begin Custom Macros
OutDir=.\\Debug
# End Custom Macros

ALL : "$(OUTDIR)\SvrAplBase.dll"
"\SvrAplBase.tlb" ".\SvrAplBase.h"
"\SvrAplBase_i.c" ".\Debugregsvr32.trg"

CLEAN :
    -@erase
    "$(INTDIR)\CntlExecuteThread.obj"
    -@erase
    "$(INTDIR)\CreateCntlProcess.obj"

```

```

-@erase
"$$(INTDIR)\GetCntlProcessNum.obj"
-@erase "$(INTDIR)\GlobalArea.obj"
-@erase "$(INTDIR)\LoadSvrAplDll.obj"
-@erase "$(INTDIR)\RequestProc.obj"
-@erase "$(INTDIR)\StdAfx.obj"
-@erase "$(INTDIR)\SvrApl.obj"
-@erase "$(INTDIR)\SvrAplBase.obj"
-@erase "$(INTDIR)\SvrAplBase.pch"
-@erase "$(INTDIR)\SvrAplBase.res"
-@erase "$(INTDIR)\TermSvrAplCntl.obj"
-@erase "$(INTDIR)\vc60.idb"
-@erase "$(OUTDIR)\vc60.pdb"
-@erase "$(INTDIR)\WaitProcessEnd.obj"
-@erase "$(OUTDIR)\SvrAplBase.dll"
-@erase "$(OUTDIR)\SvrAplBase.exp"
-@erase "$(OUTDIR)\SvrAplBase.ilk"
-@erase "$(OUTDIR)\SvrAplBase.lib"
-@erase "$(OUTDIR)\SvrAplBase.pdb"
-@erase ".\SvrAplBase.h"
-@erase ".\SvrAplBase.tlb"
-@erase ".\SvrAplBase_i.c"
-@erase ".\Debugregsvr32.trg"

"$$(OUTDIR)" :
    if not exist "$$(OUTDIR)\$(NULL)" mkdir
"$$(OUTDIR)"

CPP=cl.exe
CPP_PROJ=/nologo /MTd /W4 /Gm /ZI /Od /I
".\SvrAplSharedMem" /I ".\TPCCPublicFunc" /I
".\DBProcCntl" /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /D "_MBCS" /D "_USRDL"
/Fp"$$(INTDIR)\SvrAplBase.pch" /Yu"stdafx.h"
/Fo"$$(INTDIR)\\" /Fd"$$(INTDIR)\\" /D /GZ /c

.c{$$(INTDIR)}.obj:
    $(CPP) @<<
    $(CPP_PROJ) <
<<

.cpp{$$(INTDIR)}.obj:
    $(CPP) @<<
    $(CPP_PROJ) <
<<

.cxx{$$(INTDIR)}.obj:
    $(CPP) @<<
    $(CPP_PROJ) <
<<

.c{$$(INTDIR)}.sbr:
    $(CPP) @<<
    $(CPP_PROJ) <
<<

.cpp{$$(INTDIR)}.sbr:
    $(CPP) @<<
    $(CPP_PROJ) <
<<

.cxx{$$(INTDIR)}.sbr:
    $(CPP) @<<
    $(CPP_PROJ) <
<<

MTL=midl.exe
MTL_PROJ=
RSC=rc.exe

```

```
RSC_PROJ=/I 0x411
/fo"$(INTDIR)\SvrAplBase.res" /d "_DEBUG"
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/fo"$(OUTDIR)\SvrAplBase.bsc"
BSC32_SBRS= \

LINK32=link.exe
LINK32_FLAGS=SvrAplSharedMem.lib
TPCCPublicFunc.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib
shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib
odbccp32.lib /nologo /subsystem:windows /dll
/incremental:yes
/pdb:"$(OUTDIR)\SvrAplBase.pdb" /debug
/machine:i386 /def:".SvrAplBase.def"
/out:"$(OUTDIR)\SvrAplBase.dll"
/implib:"$(OUTDIR)\SvrAplBase.lib" /pdbtype:sept
/libpath:".SvrAplSharedMem\Debug"
/libpath:".TPCCPublicFunc\Debug"
DEF_FILE= \
    ".SvrAplBase.def"
LINK32_OBJS= \
    "$(INTDIR)\CntlExecuteThread.obj" \
    "$(INTDIR)\CreateCntlProcess.obj" \
    "$(INTDIR)\GetCntlProcessNum.obj" \
    "$(INTDIR)\GlobalArea.obj" \
    "$(INTDIR)\LoadSvrAplDll.obj" \
    "$(INTDIR)\RequestProc.obj" \
    "$(INTDIR)\StdAfx.obj" \
    "$(INTDIR)\SvrApl.obj" \
    "$(INTDIR)\SvrAplBase.obj" \
    "$(INTDIR)\TermSvrAplCntl.obj" \
    "$(INTDIR)\WaitProcessEnd.obj" \
    "$(INTDIR)\SvrAplBase.res"
"$(OUTDIR)\SvrAplBase.dll" : "$(OUTDIR)"
$(DEF_FILE) $(LINK32_OBJS)
$(LINK32) @<<
$(LINK32_FLAGS) $(LINK32_OBJS)
<<
OutDir=.Debug
TargetPath=.Debug\SvrAplBase.dll
InputPath=.Debug\SvrAplBase.dll
SOURCE="$(InputPath)"
"$(OUTDIR)\regsvr32.trg" : $(SOURCE)
"$(INTDIR)" "$(OUTDIR)"
<<tempfile.bat
@echo off
echo regsvr32 /s /c "$(TargetPath)"
echo regsvr32 exec. time >
"$(OutDir)\regsvr32.trg"
<<

SOURCE="$(InputPath)"
PostBuild_Desc=bin
DS_POSTBUILD_DEP=$(INTDIR)\postbld.dep
ALL : $(DS_POSTBUILD_DEP)
# Begin Custom Macros
OutDir=.Debug
# End Custom Macros
$(DS_POSTBUILD_DEP) :
"$(OUTDIR)\SvrAplBase.dll" ".SvrAplBase.tlb"
".SvrAplBase.h" ".SvrAplBase_i.c"
".Debug\regsvr32.trg"

copy .Debug\SvrAplBase.dll .bin\Debug\SvrAplB
ase.dll
echo Helper for Post-build step >
"$(DS_POSTBUILD_DEP)"
```

```
!ELSEIF "$(CFG)" == "SvrAplBase - Win32
Unicode Debug"
OUTDIR=.DebugU
INTDIR=.DebugU
# Begin Custom Macros
OutDir=.DebugU
# End Custom Macros
ALL : "$(OUTDIR)\SvrAplBase.dll"
".SvrAplBase.tlb" ".SvrAplBase.h"
".SvrAplBase_i.c" ".DebugU\regsvr32.trg"
CLEAN :
    -@erase
"$(INTDIR)\CntlExecuteThread.obj"
    -@erase
"$(INTDIR)\CreateCntlProcess.obj"
    -@erase
"$(INTDIR)\GetCntlProcessNum.obj"
    -@erase "$(INTDIR)\GlobalArea.obj"
    -@erase "$(INTDIR)\LoadSvrAplDll.obj"
    -@erase "$(INTDIR)\RequestProc.obj"
    -@erase "$(INTDIR)\StdAfx.obj"
    -@erase "$(INTDIR)\SvrApl.obj"
    -@erase "$(INTDIR)\SvrAplBase.obj"
    -@erase "$(INTDIR)\SvrAplBase.pch"
    -@erase "$(INTDIR)\SvrAplBase.res"
    -@erase "$(INTDIR)\TermSvrAplCntl.obj"
    -@erase "$(INTDIR)\vc60.idb"
    -@erase "$(INTDIR)\vc60.pdb"
    -@erase "$(INTDIR)\WaitProcessEnd.obj"
    -@erase "$(OUTDIR)\SvrAplBase.dll"
    -@erase "$(OUTDIR)\SvrAplBase.exp"
    -@erase "$(OUTDIR)\SvrAplBase.ilk"
    -@erase "$(OUTDIR)\SvrAplBase.lib"
    -@erase "$(OUTDIR)\SvrAplBase.pdb"
    -@erase ".SvrAplBase.h"
    -@erase ".SvrAplBase.tlb"
    -@erase ".SvrAplBase_i.c"
    -@erase ".DebugU\regsvr32.trg"

"$(OUTDIR)" :
    if not exist "$(OUTDIR)\$(NULL)" mkdir
"$(OUTDIR)"

CPP=c.exe
CPP_PROJ=/nologo /MTd /W3 /Gm /ZI /Od /D
"WIN32" /D "_DEBUG" /D "_WINDOWS" /D
"_USRDLL" /D "_UNICODE"
/Fp"$(INTDIR)\SvrAplBase.pch" /Yu"stdafx.h"
/Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /GZ /c

.c{$(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<

.cpp{$(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<

.cxx{$(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<

.c{$(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
```

```
.cpp{$(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<

.cxx{$(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<

MTL=midl.exe
MTL_PROJ=
RSC=rc.exe
RSC_PROJ=/I 0x411
/fo"$(INTDIR)\SvrAplBase.res" /d "_DEBUG"
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/fo"$(OUTDIR)\SvrAplBase.bsc"
BSC32_SBRS= \

LINK32=link.exe
LINK32_FLAGS=kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbcc32.lib
odbccp32.lib /nologo /subsystem:windows /dll
/incremental:yes
/pdb:"$(OUTDIR)\SvrAplBase.pdb" /debug
/machine:i386 /def:".SvrAplBase.def"
/out:"$(OUTDIR)\SvrAplBase.dll"
/implib:"$(OUTDIR)\SvrAplBase.lib" /pdbtype:sept
DEF_FILE= \
    ".SvrAplBase.def"
LINK32_OBJS= \
    "$(INTDIR)\CntlExecuteThread.obj" \
    "$(INTDIR)\CreateCntlProcess.obj" \
    "$(INTDIR)\GetCntlProcessNum.obj" \
    "$(INTDIR)\GlobalArea.obj" \
    "$(INTDIR)\LoadSvrAplDll.obj" \
    "$(INTDIR)\RequestProc.obj" \
    "$(INTDIR)\StdAfx.obj" \
    "$(INTDIR)\SvrApl.obj" \
    "$(INTDIR)\SvrAplBase.obj" \
    "$(INTDIR)\TermSvrAplCntl.obj" \
    "$(INTDIR)\WaitProcessEnd.obj" \
    "$(INTDIR)\SvrAplBase.res"
"$(OUTDIR)\SvrAplBase.dll" : "$(OUTDIR)"
$(DEF_FILE) $(LINK32_OBJS)
$(LINK32) @<<
$(LINK32_FLAGS) $(LINK32_OBJS)
<<

OutDir=.DebugU
TargetPath=.DebugU\SvrAplBase.dll
InputPath=.DebugU\SvrAplBase.dll
SOURCE="$(InputPath)"

"$(OUTDIR)\regsvr32.trg" : $(SOURCE)
"$(INTDIR)" "$(OUTDIR)"
<<tempfile.bat
@echo off
if "%OS%"==" " goto NOTNT
if not "%OS%"=="Windows_NT" goto
NOTNT
regsvr32 /s /c "$(TargetPath)"
echo regsvr32 exec. time >
"$(OutDir)\regsvr32.trg"
goto end
```

```

:NOTNT
echo : Windows 95 Unicode DLL
:end
<<

!ELSEIF "$(CFG)" == "SvrAplBase - Win32
Release MinSize"

OUTDIR=. \ReleaseMinSize
INTDIR=. \ReleaseMinSize
# Begin Custom Macros
OutDir=. \ReleaseMinSize
# End Custom Macros

ALL : "$(OUTDIR)\SvrAplBase.dll"
".\ReleaseMinSize\regsvr32.trg"

CLEAN :
-@erase
"$(INTDIR)\CntlExecuteThread.obj"
-@erase
"$(INTDIR)\CreateCntlProcess.obj"
-@erase
"$(INTDIR)\GetCntlProcessNum.obj"
-@erase "$(INTDIR)\GlobalArea.obj"
-@erase "$(INTDIR)\LoadSvrAplDll.obj"
-@erase "$(INTDIR)\RequestProc.obj"
-@erase "$(INTDIR)\StdAfx.obj"
-@erase "$(INTDIR)\SvrApl.obj"
-@erase "$(INTDIR)\SvrAplBase.obj"
-@erase "$(INTDIR)\SvrAplBase.pch"
-@erase "$(INTDIR)\SvrAplBase.res"
-@erase "$(INTDIR)\TermSvrAplCntl.obj"
-@erase "$(INTDIR)\vc60.idb"
-@erase "$(INTDIR)\WaitProcessEnd.obj"
-@erase "$(OUTDIR)\SvrAplBase.dll"
-@erase "$(OUTDIR)\SvrAplBase.exp"
-@erase "$(OUTDIR)\SvrAplBase.lib"
-@erase ".\SvrAplBase.h"
-@erase ".\SvrAplBase.tlb"
-@erase ".\SvrAplBase_i.c"
-@erase ".\ReleaseMinSize\regsvr32.trg"
"$(OUTDIR)" :
if not exist "$(OUTDIR)\$(NULL)" mkdir
"$(OUTDIR)"
CPP=cl.exe
CPP_PROJ=/nologo /MT /W4 /O1 /I
".. \SvrAplSharedMem" /I ".. \TPCCPublicFunc" /I
".. \DBProcCntl" /D "SAMPLE" /D
"XPROC_MODE" /D "WIN32" /D "NDEBUG" /D
"_WINDOWS" /D "_MBCS" /D "_USRDLL" /D
"_ATL_DLL" /Fp "$(INTDIR)\SvrAplBase.pch"
/Yu "stdafx.h" /Fo "$(INTDIR)\\" /Fd "$(INTDIR)\\"
/FD /c
.c $(INTDIR).obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cpp $(INTDIR).obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cxx $(INTDIR).obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.c $(INTDIR).sbr:

```

```

$(CPP) @<<
$(CPP_PROJ) $<
<<
.cpp $(INTDIR).sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cxx $(INTDIR).sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
MTL=midl.exe
MTL_PROJ=
RSC=rc.exe
RSC_PROJ=/I 0x411
/fo "$(INTDIR)\SvrAplBase.res" /d "NDEBUG"
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/o "$(OUTDIR)\SvrAplBase.bsc"
BSC32_SBRS= \

LINK32=link.exe
LINK32_FLAGS=SvrAplSharedMem.lib
TPCCPublicFunc.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib
shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib
odbccp32.lib /nologo /subsystem:windows /dll
/incremental:no
/pdb:"$(OUTDIR)\SvrAplBase.pdb" /machine:I386
/def:". \SvrAplBase.def"
/out:"$(OUTDIR)\SvrAplBase.dll"
/implib:"$(OUTDIR)\SvrAplBase.lib"
/libpath:". \SvrAplSharedMem\Release"
/libpath:". \TPCCPublicFunc\Release"
DEF_FILE= \
". \SvrAplBase.def"
LINK32_OBJS= \
"$(INTDIR)\CntlExecuteThread.obj" \
"$(INTDIR)\CreateCntlProcess.obj" \
"$(INTDIR)\GetCntlProcessNum.obj" \
"$(INTDIR)\GlobalArea.obj" \
"$(INTDIR)\LoadSvrAplDll.obj" \
"$(INTDIR)\RequestProc.obj" \
"$(INTDIR)\StdAfx.obj" \
"$(INTDIR)\SvrApl.obj" \
"$(INTDIR)\SvrAplBase.obj" \
"$(INTDIR)\TermSvrAplCntl.obj" \
"$(INTDIR)\WaitProcessEnd.obj" \
"$(INTDIR)\SvrAplBase.res"

"$(OUTDIR)\SvrAplBase.dll" : "$(OUTDIR)"
$(DEF_FILE) $(LINK32_OBJS)
$(LINK32) @<<
$(LINK32_FLAGS) $(LINK32_OBJS)
<<

OutDir=. \ReleaseMinSize
TargetPath=. \ReleaseMinSize\SvrAplBase.dll
InputPath=. \ReleaseMinSize\SvrAplBase.dll
SOURCE="$(InputPath)"

"$(OUTDIR)\regsvr32.trg" : $(SOURCE)
"$(INTDIR)" "$(OUTDIR)"
<<tempfile.bat
@echo off
regsvr32 /s /c "$(TargetPath)"
echo regsvr32 exec. time >
"$(OutDir)\regsvr32.trg"
<<

```

```

SOURCE="$(InputPath)"
PostBuild_Desc=bin
DS_POSTBUILD_DEP=$(INTDIR)\postbld.dep

ALL : $(DS_POSTBUILD_DEP)

# Begin Custom Macros
OutDir=. \ReleaseMinSize
# End Custom Macros

$(DS_POSTBUILD_DEP) :
"$(OUTDIR)\SvrAplBase.dll"
". \ReleaseMinSize\regsvr32.trg"

copy .\ReleaseMinSize\SvrAplBase.dll ..\bin\Relea
se\SvrAplBase.dll
echo Helper for Post-build step >
"$(DS_POSTBUILD_DEP)"

!ELSEIF "$(CFG)" == "SvrAplBase - Win32
Release MinDependency"

OUTDIR=. \ReleaseMinDependency
INTDIR=. \ReleaseMinDependency
# Begin Custom Macros
OutDir=. \ReleaseMinDependency
# End Custom Macros

ALL : "$(OUTDIR)\SvrAplBase.dll"
".\SvrAplBase.tlb" ".\SvrAplBase.h"
".\SvrAplBase_i.c"
".\ReleaseMinDependency\regsvr32.trg"

CLEAN :
-@erase
"$(INTDIR)\CntlExecuteThread.obj"
-@erase
"$(INTDIR)\CreateCntlProcess.obj"
-@erase
"$(INTDIR)\GetCntlProcessNum.obj"
-@erase "$(INTDIR)\GlobalArea.obj"
-@erase "$(INTDIR)\LoadSvrAplDll.obj"
-@erase "$(INTDIR)\RequestProc.obj"
-@erase "$(INTDIR)\StdAfx.obj"
-@erase "$(INTDIR)\SvrApl.obj"
-@erase "$(INTDIR)\SvrAplBase.obj"
-@erase "$(INTDIR)\SvrAplBase.pch"
-@erase "$(INTDIR)\SvrAplBase.res"
-@erase "$(INTDIR)\TermSvrAplCntl.obj"
-@erase "$(INTDIR)\vc60.idb"
-@erase "$(INTDIR)\WaitProcessEnd.obj"
-@erase "$(OUTDIR)\SvrAplBase.dll"
-@erase "$(OUTDIR)\SvrAplBase.exp"
-@erase "$(OUTDIR)\SvrAplBase.lib"
-@erase ".\SvrAplBase.h"
-@erase ".\SvrAplBase.tlb"
-@erase ".\SvrAplBase_i.c"
-@erase
". \ReleaseMinDependency\regsvr32.trg"

"$(OUTDIR)" :
if not exist "$(OUTDIR)\$(NULL)" mkdir
"$(OUTDIR)"

CPP=d.exe
CPP_PROJ=/nologo /MT /W3 /O2 /I
".. \SvrAplSharedMem" /I ".. \TPCCPublicFunc" /D

```

```

"WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"_MBCS" /D "_USRDLL" /D
"_ATL_STATIC_REGISTRY"
/Fp"${(INTDIR)\SvrAplBase.pch" /Yu"stdafx.h"
/Fo"${(INTDIR)\\" /Fd"${(INTDIR)\\" /FD /c
.c${(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) <
<<
.cpp${(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) <
<<
.cxx${(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) <
<<
.c${(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) <
<<
.cpp${(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) <
<<
.cxx${(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) <
<<
MTL=midl.exe
MTL_PROJ=
RSC=rc.exe
RSC_PROJ=/I 0x411
/fo"${(INTDIR)\SvrAplBase.res" /d "NDEBUG"
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/o"${(OUTDIR)\SvrAplBase.bsc"
BSC32_SBRS= \

LINK32=link.exe
LINK32_FLAGS=SvrAplSharedMem.lib
TPCCPublicFunc.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib
shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib
odbccp32.lib /nologo /subsystem:windows /dll
/incremental:no
/pdb:"$(OUTDIR)\SvrAplBase.pdb" /machine:i386
/def:".SvrAplBase.def"
/out:"$(OUTDIR)\SvrAplBase.dll"
/implib:"$(OUTDIR)\SvrAplBase.lib"
/libpath:".SvrAplSharedMem\Release"
/libpath:".TPCCPublicFunc\Release"
DEF_FILE= \
    ".SvrAplBase.def"
LINK32_OBJS= \
    "$(INTDIR)\CntlExecuteThread.obj" \
    "$(INTDIR)\CreateCntlProcess.obj" \
    "$(INTDIR)\GetCntlProcessNum.obj" \
    "$(INTDIR)\GlobalArea.obj" \
    "$(INTDIR)\LoadSvrAplDll.obj" \
    "$(INTDIR)\RequestProc.obj" \
    "$(INTDIR)\StdAfx.obj" \
    "$(INTDIR)\SvrApl.obj" \
    "$(INTDIR)\SvrAplBase.obj" \
    "$(INTDIR)\TermSvrAplCntl.obj" \
    "$(INTDIR)\WaitProcessEnd.obj" \
    "$(INTDIR)\SvrAplBase.res"
"$(OUTDIR)\SvrAplBase.dll" : "$(OUTDIR)
$(DEF_FILE) $(LINK32_OBJS)

```

```

$(LINK32) @<<
$(LINK32_FLAGS) $(LINK32_OBJS)
<<
OutDir=.ReleaseMinDependency
TargetPath=.ReleaseMinDependency\SvrAplBase
.dll
InputPath=.ReleaseMinDependency\SvrAplBase.
.dll
SOURCE="$(InputPath)"
"$(OUTDIR)\regsvr32.trg" : $(SOURCE)
"$(INTDIR)" "$(OUTDIR)"
<<tempfile.bat
@echo off
echo regsvr32 /s /c "$(TargetPath)"
echo regsvr32 exec. time >
"$(OutDir)\regsvr32.trg"
<<

SOURCE="$(InputPath)"
PostBuild_Desc=bin
DS_POSTBUILD_DEP=$(INTDIR)\postbld.dep
ALL : $(DS_POSTBUILD_DEP)
# Begin Custom Macros
OutDir=.ReleaseMinDependency
# End Custom Macros
$(DS_POSTBUILD_DEP) :
"$(OUTDIR)\SvrAplBase.dll" ".SvrAplBase.tlb"
".SvrAplBase.h" ".SvrAplBase_i.c"
".ReleaseMinDependency\regsvr32.trg"

copy .ReleaseMinDependency\SvrAplBase.dll ..lb
in\Release\SvrAplBase.dll
echo Helper for Post-build step >
"$(DS_POSTBUILD_DEP)"

!ELSEIF "$(CFG) == "SvrAplBase - Win32
Unicode Release MinSize"

OUTDIR=.ReleaseUMinSize
INTDIR=.ReleaseUMinSize
# Begin Custom Macros
OutDir=.ReleaseUMinSize
# End Custom Macros

ALL : "$(OUTDIR)\SvrAplBase.dll"
".SvrAplBase.tlb" ".SvrAplBase.h"
".SvrAplBase_i.c"
".ReleaseUMinSize\regsvr32.trg"

CLEAN :
-@erase
"$(INTDIR)\CntlExecuteThread.obj"
-@erase
"$(INTDIR)\CreateCntlProcess.obj"
-@erase
"$(INTDIR)\GetCntlProcessNum.obj"
-@erase "$(INTDIR)\GlobalArea.obj"
-@erase "$(INTDIR)\LoadSvrAplDll.obj"
-@erase "$(INTDIR)\RequestProc.obj"
-@erase "$(INTDIR)\StdAfx.obj"
-@erase "$(INTDIR)\SvrApl.obj"
-@erase "$(INTDIR)\SvrAplBase.obj"
-@erase "$(INTDIR)\SvrAplBase.pch"
-@erase "$(INTDIR)\SvrAplBase.res"
-@erase "$(INTDIR)\TermSvrAplCntl.obj"
-@erase "$(INTDIR)\vc60.idb"
-@erase "$(INTDIR)\WaitProcessEnd.obj"
-@erase "$(OUTDIR)\SvrAplBase.dll"

```

```

-@erase "$(OUTDIR)\SvrAplBase.exp"
-@erase "$(OUTDIR)\SvrAplBase.lib"
-@erase ".SvrAplBase.h"
-@erase ".SvrAplBase.tlb"
-@erase ".SvrAplBase_i.c"
-@erase ".ReleaseUMinSize\regsvr32.trg"

"$(OUTDIR)" :
if not exist "$(OUTDIR)\$(NULL)" mkdir
"$(OUTDIR)"

CPP=cl.exe
CPP_PROJ=/nologo /MT /W3 /O1 /D "WIN32" /D
"NDEBUG" /D "_WINDOWS" /D "_USRDLL" /D
"_UNICODE" /D "_ATL_DLL" /D
"_ATL_MIN_CRT"
/Fp"${(INTDIR)\SvrAplBase.pch" /Yu"stdafx.h"
/Fo"${(INTDIR)\\" /Fd"${(INTDIR)\\" /FD /c
.c${(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) <
<<
.cpp${(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) <
<<
.cxx${(INTDIR)}.obj:
$(CPP) @<<
$(CPP_PROJ) <
<<
.c${(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) <
<<
.cpp${(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) <
<<
.cxx${(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) <
<<
.c${(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) <
<<
.cpp${(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) <
<<
.cxx${(INTDIR)}.sbr:
$(CPP) @<<
$(CPP_PROJ) <
<<
MTL=midl.exe
MTL_PROJ=
RSC=rc.exe
RSC_PROJ=/I 0x411
/fo"${(INTDIR)\SvrAplBase.res" /d "NDEBUG"
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/o"${(OUTDIR)\SvrAplBase.bsc"
BSC32_SBRS= \

LINK32=link.exe
LINK32_FLAGS=kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbcc32.lib
odbccp32.lib /nologo /subsystem:windows /dll
/incremental:no
/pdb:"$(OUTDIR)\SvrAplBase.pdb" /machine:i386
/def:".SvrAplBase.def"
/out:"$(OUTDIR)\SvrAplBase.dll"
/implib:"$(OUTDIR)\SvrAplBase.lib"
DEF_FILE= \
    ".SvrAplBase.def"
LINK32_OBJS= \
    "$(INTDIR)\CntlExecuteThread.obj" \
    "$(INTDIR)\CreateCntlProcess.obj" \
    "$(INTDIR)\GetCntlProcessNum.obj" \
    "$(INTDIR)\GlobalArea.obj" \

```

```

$(INTDIR)\LoadSvrAplDll.obj" \
$(INTDIR)\RequestProc.obj" \
$(INTDIR)\StdAfx.obj" \
$(INTDIR)\SvrApl.obj" \
$(INTDIR)\SvrAplBase.obj" \
$(INTDIR)\TermSvrAplCntl.obj" \
$(INTDIR)\WaitProcessEnd.obj" \
$(INTDIR)\SvrAplBase.res"
$(OUTDIR)\SvrAplBase.dll" : "$(OUTDIR)"
$(DEF_FILE) $(LINK32_OBJS)
$(LINK32) @<<
$(LINK32_FLAGS) $(LINK32_OBJS)
<<
OutDir=. \ReleaseUMinSize
TargetPath=. \ReleaseUMinSize\SvrAplBase.dll
InputPath=. \ReleaseUMinSize\SvrAplBase.dll
SOURCE="$(InputPath)"
$(OUTDIR)regsvr32.trg" : $(SOURCE)
$(INTDIR) "$(OUTDIR)"
<<tempfile.bat
@echo off
if "%OS%"==" " goto NOTNT
if not "%OS%"=="Windows_NT" goto
NOTNT
regsvr32 /s /c "$(TargetPath)"
echo regsvr32 exec. time >
$(OutDir)regsvr32.trg"
goto end
:NOTNT
echo : Windows 95 Unicode DLL
:end
<<

!ELSEIF "$(CFG)" == "SvrAplBase - Win32
Unicode Release MinDependency"
OUTDIR=. \ReleaseUMinDependency
INTDIR=. \ReleaseUMinDependency
# Begin Custom Macros
OutDir=. \ReleaseUMinDependency
# End Custom Macros
ALL : "$(OUTDIR)\SvrAplBase.dll"
".\SvrAplBase.lib" ".\SvrAplBase.h"
".\SvrAplBase_i.c"
". \ReleaseUMinDependency\regsvr32.trg"
CLEAN :
-@erase
$(INTDIR)\CntlExecuteThread.obj"
-@erase
$(INTDIR)\CreateCntlProcess.obj"
-@erase
$(INTDIR)\GetCntlProcessNum.obj"
-@erase "$(INTDIR)\GlobalArea.obj"
-@erase "$(INTDIR)\LoadSvrAplDll.obj"
-@erase "$(INTDIR)\RequestProc.obj"
-@erase "$(INTDIR)\StdAfx.obj"
-@erase "$(INTDIR)\SvrApl.obj"
-@erase "$(INTDIR)\SvrAplBase.obj"
-@erase "$(INTDIR)\SvrAplBase.pch"
-@erase "$(INTDIR)\SvrAplBase.res"
-@erase "$(INTDIR)\TermSvrAplCntl.obj"
-@erase "$(INTDIR)\vc60.idb"
-@erase "$(INTDIR)\WaitProcessEnd.obj"
-@erase "$(OUTDIR)\SvrAplBase.dll"
-@erase "$(OUTDIR)\SvrAplBase.exp"
-@erase "$(OUTDIR)\SvrAplBase.lib"
-@erase ".\SvrAplBase.h"
-@erase ".\SvrAplBase.lib"
-@erase ".\SvrAplBase_i.c"

```

```

-@erase
". \ReleaseUMinDependency\regsvr32.trg"
$(OUTDIR) :
if not exist "$(OUTDIR)\$(NULL)" mkdir
$(OUTDIR)
CPP=cl.exe
CPP_PROJ=/nologo /MT /W3 /O1 /D "WIN32" /D
"NDEBUG" /D "_WINDOWS" /D "_USRDLL" /D
"_UNICODE" /D "_ATL_STATIC_REGISTRY" /D
"_ATL_MIN_CRT"
/Fp "$(INTDIR)\SvrAplBase.pch" /Yu"stdafx.h"
/Fo "$(INTDIR)\\" /Fd "$(INTDIR)\\" /FD /c
.c $(INTDIR).obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cxx $(INTDIR).obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.c $(INTDIR).sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cpp $(INTDIR).sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cxx $(INTDIR).sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<

MTL=midl.exe
MTL_PROJ=
RSC=rc.exe
RSC_PROJ=/I 0x411
/fo "$(INTDIR)\SvrAplBase.res" /d "NDEBUG"
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/fo "$(OUTDIR)\SvrAplBase.bsc"
BSC32_SBRS= \

LINK32=link.exe
LINK32_FLAGS=kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbcc32.lib
/obccpp32.lib /nologo /subsystem:windows /dll
/incremental:no
/pdb:"$(OUTDIR)\SvrAplBase.pdb" /machine:I386
/def:". \SvrAplBase.def"
/out:"$(OUTDIR)\SvrAplBase.dll"
/implib:"$(OUTDIR)\SvrAplBase.lib"
DEF_FILE= \
". \SvrAplBase.def"
LINK32_OBJS= \
$(INTDIR)\CntlExecuteThread.obj" \
$(INTDIR)\CreateCntlProcess.obj" \
$(INTDIR)\GetCntlProcessNum.obj" \
$(INTDIR)\GlobalArea.obj" \
$(INTDIR)\LoadSvrAplDll.obj" \

```

```

$(INTDIR)\RequestProc.obj" \
$(INTDIR)\StdAfx.obj" \
$(INTDIR)\SvrApl.obj" \
$(INTDIR)\SvrAplBase.obj" \
$(INTDIR)\TermSvrAplCntl.obj" \
$(INTDIR)\WaitProcessEnd.obj" \
$(INTDIR)\SvrAplBase.res"
$(OUTDIR)\SvrAplBase.dll" : "$(OUTDIR)"
$(DEF_FILE) $(LINK32_OBJS)
$(LINK32) @<<
$(LINK32_FLAGS) $(LINK32_OBJS)
<<
OutDir=. \ReleaseUMinDependency
TargetPath=. \ReleaseUMinDependency\SvrAplBase.dll
InputPath=. \ReleaseUMinDependency\SvrAplBase.dll
SOURCE="$(InputPath)"
$(OUTDIR)regsvr32.trg" : $(SOURCE)
$(INTDIR) "$(OUTDIR)"
<<tempfile.bat
@echo off
if "%OS%"==" " goto NOTNT
if not "%OS%"=="Windows_NT" goto
NOTNT
regsvr32 /s /c "$(TargetPath)"
echo regsvr32 exec. time >
$(OutDir)regsvr32.trg"
goto end
:NOTNT
echo : Windows 95 Unicode DLL
:end
<<

!ENDIF
!IF "$(NO_EXTERNAL_DEPS)" != "1"
!IF EXISTS("SvrAplBase.dep")
!INCLUDE "SvrAplBase.dep"
!ELSE
!MESSAGE Warning: cannot find
"SvrAplBase.dep"
!ENDIF
!ENDIF
!IF "$(CFG)" == "SvrAplBase - Win32 Debug" ||
$(CFG) == "SvrAplBase - Win32 Unicode
Debug" || $(CFG) == "SvrAplBase - Win32
Release MinSize" || $(CFG) == "SvrAplBase -
Win32 Release MinDependency" || $(CFG) ==
"SvrAplBase - Win32 Unicode Release MinSize" ||
$(CFG) == "SvrAplBase - Win32 Unicode
Release MinDependency"
SOURCE=. \CntlExecuteThread.cpp
$(INTDIR)\CntlExecuteThread.obj" : $(SOURCE)
$(INTDIR) "$(INTDIR)\SvrAplBase.pch"
SOURCE=. \CreateCntlProcess.cpp
$(INTDIR)\CreateCntlProcess.obj" : $(SOURCE)
$(INTDIR) "$(INTDIR)\SvrAplBase.pch"
SOURCE=. \GetCntlProcessNum.cpp
$(INTDIR)\GetCntlProcessNum.obj" :
$(SOURCE) "$(INTDIR)"
$(INTDIR)\SvrAplBase.pch"
SOURCE=. \GlobalArea.cpp
$(INTDIR)\GlobalArea.obj" : $(SOURCE)
$(INTDIR) "$(INTDIR)\SvrAplBase.pch"
SOURCE=. \LoadSvrAplDll.cpp
$(INTDIR)\LoadSvrAplDll.obj" : $(SOURCE)
$(INTDIR) "$(INTDIR)\SvrAplBase.pch"

```

```

SOURCE=.RequestProc.cpp
"$(INTDIR)\RequestProc.obj" : $(SOURCE)
"$(INTDIR)" "$(INTDIR)\SvrAplBase.pch"
SOURCE=.StdAfx.cpp
!IF "$(CFG)" == "SvrAplBase - Win32 Debug"
CPP_SWITCHES=/nologo /MTd /W4 /Gm /ZI /Od
/I ".\SvrAplSharedMem" /I ".\TPCCPublicFunc" /I
"\.DBProcCntl" /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /D "_MBCS" /D "_USRDLL"
/Fp"$(INTDIR)\SvrAplBase.pch" /Yc"stdafx.h"
/Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /GZ /c
"$(INTDIR)\StdAfx.obj"
"$(INTDIR)\SvrAplBase.pch" : $(SOURCE)
"$(INTDIR)"
$(CPP) @<<
$(CPP_SWITCHES) $(SOURCE)
<<
!ELSEIF "$(CFG)" == "SvrAplBase - Win32
Unicode Debug"
CPP_SWITCHES=/nologo /MTd /W3 /Gm /ZI /Od
/D "WIN32" /D "_DEBUG" /D "_WINDOWS" /D
"_USRDLL" /D "_UNICODE"
/Fp"$(INTDIR)\SvrAplBase.pch" /Yc"stdafx.h"
/Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /GZ /c
"$(INTDIR)\StdAfx.obj"
"$(INTDIR)\SvrAplBase.pch" : $(SOURCE)
"$(INTDIR)"
$(CPP) @<<
$(CPP_SWITCHES) $(SOURCE)
<<
!ELSEIF "$(CFG)" == "SvrAplBase - Win32
Release MinSize"
CPP_SWITCHES=/nologo /MT /W4 /O1 /I
".\SvrAplSharedMem" /I ".\TPCCPublicFunc" /I
"\.DBProcCntl" /D "SAMPLE" /D
"XPROC_MODE" /D "WIN32" /D "NDEBUG" /D
"_WINDOWS" /D "_MBCS" /D "_USRDLL" /D
"_ATL_DLL" /Fp"$(INTDIR)\SvrAplBase.pch"
/Yc"stdafx.h" /Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\"
/FD /c
"$(INTDIR)\StdAfx.obj"
"$(INTDIR)\SvrAplBase.pch" : $(SOURCE)
"$(INTDIR)"
$(CPP) @<<
$(CPP_SWITCHES) $(SOURCE)
<<
!ELSEIF "$(CFG)" == "SvrAplBase - Win32
Release MinDependency"
CPP_SWITCHES=/nologo /MT /W3 /O2 /I
".\SvrAplSharedMem" /I ".\TPCCPublicFunc" /D
"WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"_MBCS" /D "_USRDLL" /D
"_ATL_STATIC_REGISTRY"
/Fp"$(INTDIR)\SvrAplBase.pch" /Yc"stdafx.h"
/Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /c
"$(INTDIR)\StdAfx.obj"
"$(INTDIR)\SvrAplBase.pch" : $(SOURCE)
"$(INTDIR)"
$(CPP) @<<
$(CPP_SWITCHES) $(SOURCE)
<<
!ELSEIF "$(CFG)" == "SvrAplBase - Win32
Unicode Release MinSize"
CPP_SWITCHES=/nologo /MT /W3 /O1 /D
"WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"_USRDLL" /D "_UNICODE" /D "_ATL_DLL" /D
"_ATL_MIN_CRT"

```

```

/Fp"$(INTDIR)\SvrAplBase.pch" /Yc"stdafx.h"
/Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /c
"$(INTDIR)\StdAfx.obj"
"$(INTDIR)\SvrAplBase.pch" : $(SOURCE)
"$(INTDIR)"
$(CPP) @<<
$(CPP_SWITCHES) $(SOURCE)
<<
!ELSEIF "$(CFG)" == "SvrAplBase - Win32
Unicode Release MinDependency"
CPP_SWITCHES=/nologo /MT /W3 /O1 /D
"WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"_USRDLL" /D "_UNICODE" /D
"_ATL_STATIC_REGISTRY" /D
"_ATL_MIN_CRT"
/Fp"$(INTDIR)\SvrAplBase.pch" /Yc"stdafx.h"
/Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /c
"$(INTDIR)\StdAfx.obj"
"$(INTDIR)\SvrAplBase.pch" : $(SOURCE)
"$(INTDIR)"
$(CPP) @<<
$(CPP_SWITCHES) $(SOURCE)
<<
!ENDIF
SOURCE=.SvrApl.cpp
"$(INTDIR)\SvrApl.obj" : $(SOURCE) "$(INTDIR)"
"$(INTDIR)\SvrAplBase.pch"
SOURCE=.SvrAplBase.cpp
"$(INTDIR)\SvrAplBase.obj" : $(SOURCE)
"$(INTDIR)" "$(INTDIR)\SvrAplBase.pch"
SOURCE=.SvrAplBase.idl
!IF "$(CFG)" == "SvrAplBase - Win32 Debug"
MTL_SWITCHES=/tlb ".\SvrAplBase.tlb" /h
"SvrAplBase.h" /iid "SvrAplBase_i.c" /Oicf
".\SvrAplBase.tlb" ".\SvrAplBase.h"
".\SvrAplBase_i.c" : $(SOURCE)
"$(INTDIR)"
$(MTL) @<<
$(MTL_SWITCHES) $(SOURCE)
<<
!ELSEIF "$(CFG)" == "SvrAplBase - Win32
Unicode Debug"
MTL_SWITCHES=/tlb ".\SvrAplBase.tlb" /h
"SvrAplBase.h" /iid "SvrAplBase_i.c" /Oicf
".\SvrAplBase.tlb" ".\SvrAplBase.h"
".\SvrAplBase_i.c" : $(SOURCE)
"$(INTDIR)"
$(MTL) @<<

```

```

$(MTL_SWITCHES) $(SOURCE)
<<
!ELSEIF "$(CFG)" == "SvrAplBase - Win32
Release MinSize"
MTL_SWITCHES=/tlb ".\SvrAplBase.tlb" /h
"SvrAplBase.h" /iid "SvrAplBase_i.c" /Oicf
".\SvrAplBase.tlb" ".\SvrAplBase.h"
".\SvrAplBase_i.c" : $(SOURCE)
"$(INTDIR)"
$(MTL) @<<
$(MTL_SWITCHES) $(SOURCE)
<<
!ELSEIF "$(CFG)" == "SvrAplBase - Win32
Unicode Release MinDependency"
MTL_SWITCHES=/tlb ".\SvrAplBase.tlb" /h
"SvrAplBase.h" /iid "SvrAplBase_i.c" /Oicf
".\SvrAplBase.tlb" ".\SvrAplBase.h"
".\SvrAplBase_i.c" : $(SOURCE)
"$(INTDIR)"
$(MTL) @<<
$(MTL_SWITCHES) $(SOURCE)
<<
!ELSEIF "$(CFG)" == "SvrAplBase - Win32
Unicode Release MinDependency"
MTL_SWITCHES=/tlb ".\SvrAplBase.tlb" /h
"SvrAplBase.h" /iid "SvrAplBase_i.c" /Oicf
".\SvrAplBase.tlb" ".\SvrAplBase.h"
".\SvrAplBase_i.c" : $(SOURCE)
"$(INTDIR)"
$(MTL) @<<
$(MTL_SWITCHES) $(SOURCE)
<<
!ENDIF
SOURCE=.SvrAplBase.rc
"$(INTDIR)\SvrAplBase.res" : $(SOURCE)
"$(INTDIR)"
$(RSC) $(RSC_PROJ) $(SOURCE)
SOURCE=.TermSvrAplCntl.cpp
"$(INTDIR)\TermSvrAplCntl.obj" : $(SOURCE)
"$(INTDIR)" "$(INTDIR)\SvrAplBase.pch"
SOURCE=.WaitProcessEnd.cpp
"$(INTDIR)\WaitProcessEnd.obj" : $(SOURCE)
"$(INTDIR)" "$(INTDIR)\SvrAplBase.pch"
!ENDIF
.....
SvrAplBase/SvrAplBase_i.c
.....
/* this file contains the actual definitions of */
/* the IIDs and CLSIDs */
/* link this file in with the server and any clients */
/* File created by MIDL compiler version
5.01.0164 */
/* at Thu Sep 04 19:19:40 2003
*/

```

```

/* Compiler settings for
F:\SvrAplBase\SvrAplBase.idl:
  Oicf (OptLev=i2), W1, Zp8, env=Win32, ms_ext,
c_ext
  error checks: allocation ref bounds_check enum
stub_data
*/
//@@@MIDL_FILE_HEADING( )
#ifdef __cplusplus
extern "C"{
#endif
#ifndef __IID_DEFINED__
#define __IID_DEFINED__
typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
} IID;
#endif // __IID_DEFINED__
#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED
const IID IID_ISvrApl =
{0xCB255680,0xB8B2,0x48AD,{0x99,0x6B,0x6D,
0x2E,0x82,0x6B,0x03,0xF7}};
const IID LIBID_SVRAPLBASELib =
{0x3348A83E,0xDDDB2,0x46AC,{0x9F,0xD7,0x59,
0x1C,0xE5,0x9A,0xE6,0x3A}};
const CLSID CLSID_SvrApl =
{0x1924BC6E,0x7A8C,0x427D,{0x81,0xDA,0xE0,
0x02,0xF2,0x89,0x4B,0xE7}};
#ifdef __cplusplus
}
#endif
.....
SvrAplBase\SvrAplBase.p.c
.....
/* this ALWAYS GENERATED file contains the
proxy stub code */
/* File created by MIDL compiler version
5.01.0164 */
/* at Thu Sep 04 19:19:40 2003
*/
/* Compiler settings for
F:\SvrAplBase\SvrAplBase.idl:
  Oicf (OptLev=i2), W1, Zp8, env=Win32, ms_ext,
c_ext
  error checks: allocation ref bounds_check enum
stub_data
*/
//@@@MIDL_FILE_HEADING( )

#define USE_STUBLESS_PROXY

/* verify that the <rpcproxy.h> version is high
enough to compile this file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define
__REQUIRED_RPCPROXY_H_VERSION__ 440
#endif

#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__

```

```

#error this stub requires an updated version of
<rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__

#include "SvrAplBase.h"

#define TYPE_FORMAT_STRING_SIZE 7
#define PROC_FORMAT_STRING_SIZE 207

typedef struct _MIDL_TYPE_FORMAT_STRING
{
    short Pad;
    unsigned char
Format[ TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
{
    short Pad;
    unsigned char
Format[ PROC_FORMAT_STRING_SIZE ];
} MIDL_PROC_FORMAT_STRING;

extern const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString;

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,
0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: ISvrApl, ver. 0.0,
GUID={0xCB255680,0xB8B2,0x48AD,{0x99,0x6B
,0x6D,0x2E,0x82,0x6B,0x03,0xF7}} */

extern const MIDL_STUB_DESC
Object_StubDesc;

extern const MIDL_SERVER_INFO
ISvrApl_ServerInfo;
#pragma code_seg(".orpc")
static const MIDL_STUB_DESC Object_StubDesc
=
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x50100a4, /* MIDL Version 5.1.164 */
    0,
    0,
    0,
    0, /* notify & notify_flag routine table */
    1, /* Flags */

```

```

0, /* Reserved3 */
0, /* Reserved4 */
0, /* Reserved5 */
};

static const unsigned short
ISvrApl_FormatStringOffsetTable =
{
    0,
    28,
    56,
    84,
    112,
    140,
    162,
    184
};

static const MIDL_SERVER_INFO
ISvrApl_ServerInfo =
{
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &ISvrApl_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0,
    0
};

static const MIDL_STUBLESS_PROXY_INFO
ISvrApl_ProxyInfo =
{
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &ISvrApl_FormatStringOffsetTable[-3],
    0,
    0,
    0
};

CINTERFACE_PROXY_VTABLE(11)
_ISvrAplProxyVtbl =
{
    &ISvrApl_ProxyInfo,
    &IID_ISvrApl,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *)-1 /* ISvrApl::ExNewOrder */ ,
    (void *)-1 /* ISvrApl::ExPayment */ ,
    (void *)-1 /* ISvrApl::ExOrderStatus */ ,
    (void *)-1 /* ISvrApl::ExStockLevel */ ,
    (void *)-1 /* ISvrApl::ExDelivery */ ,
    (void *)-1 /* ISvrApl::EntryTrans */ ,
    (void *)-1 /* ISvrApl::TermSvrApl */ ,
    (void *)-1 /* ISvrApl::InitSvrApl */
};

const CInterfaceStubVtbl _ISvrAplStubVtbl =
{
    &IID_ISvrApl,
    &ISvrApl_ServerInfo,
    11,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

#pragma data_seg(".rdata")
#if !defined(__RPC_WIN32__)
#error Invalid build platform for this stub.
#endif
#if !(TARGET_IS_NT40_OR_LATER)

```

```

#error You need a Windows NT 4.0 or later to run
this stub because it uses these features:
#error -Oif or -Oicf, more than 32 methods in the
interface.
#error However, your C/C++ compilation flags
indicate you intend to run this app on earlier
systems.
#error This app will die there with the
RPC_X_WRONG_STUB_VERSION error.
#endif
static const MIDL_PROC_FORMAT_STRING
_MIDL_ProcFormatString =
{
    0,
    {
        /* Procedure ExNewOrder */
        0x33, /*
FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object,
Oi2 */
        /* 2 */ NdrFcLong(0x0), /* 0 */
        /* 6 */ NdrFcShort(0x3), /* 3 */
#ifdef _ALPHA_
        /* 8 */ NdrFcShort(0xc), /* x86, MIPS,
PPC Stack size/offset = 12 */
#else
        NdrFcShort(0x18), /*
Alpha Stack size/offset = 24 */
#endif
        /* 10 */ NdrFcShort(0x8), /* 8 */
        /* 12 */ NdrFcShort(0x10), /* 16 */
        /* 14 */ 0x4, /* Oi2 Flags: has return,
*/
        0x2, /* 2 */
        /* Parameter new_order_struct */
        /* 16 */ NdrFcShort(0x158), /* Flags: in,
out, base type, simple ref, */
#ifdef _ALPHA_
        /* 18 */ NdrFcShort(0x4), /* x86, MIPS,
PPC Stack size/offset = 4 */
#else
        NdrFcShort(0x8), /*
Alpha Stack size/offset = 8 */
#endif
        /* 20 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        /* Return value */

        /* 22 */ NdrFcShort(0x70), /* Flags: out,
return, base type, */
#ifdef _ALPHA_
        /* 24 */ NdrFcShort(0x8), /* x86, MIPS,
PPC Stack size/offset = 8 */
#else
        NdrFcShort(0x10), /*
Alpha Stack size/offset = 16 */
#endif
        /* 26 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        /* Procedure ExPayment */

        /* 28 */ 0x33, /* FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object,
Oi2 */
        /* 30 */ NdrFcLong(0x0), /* 0 */
        /* 34 */ NdrFcShort(0x4), /* 4 */
#ifdef _ALPHA_

```

```

        /* 36 */ NdrFcShort(0xc), /* x86, MIPS,
PPC Stack size/offset = 12 */
#else
        NdrFcShort(0x18), /*
Alpha Stack size/offset = 24 */
#endif
        /* 38 */ NdrFcShort(0x8), /* 8 */
        /* 40 */ NdrFcShort(0x10), /* 16 */
        /* 42 */ 0x4, /* Oi2 Flags: has return,
*/
        0x2, /* 2 */

        /* Parameter payment_struct */

        /* 44 */ NdrFcShort(0x158), /* Flags: in,
out, base type, simple ref, */
#ifdef _ALPHA_
        /* 46 */ NdrFcShort(0x4), /* x86, MIPS,
PPC Stack size/offset = 4 */
#else
        NdrFcShort(0x8), /*
Alpha Stack size/offset = 8 */
#endif
        /* 48 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        /* Return value */

        /* 50 */ NdrFcShort(0x70), /* Flags: out,
return, base type, */
#ifdef _ALPHA_
        /* 52 */ NdrFcShort(0x8), /* x86, MIPS,
PPC Stack size/offset = 8 */
#else
        NdrFcShort(0x10), /*
Alpha Stack size/offset = 16 */
#endif
        /* 54 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        /* Procedure ExOrderStatus */

        /* 56 */ 0x33, /* FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object,
Oi2 */
        /* 58 */ NdrFcLong(0x0), /* 0 */
        /* 62 */ NdrFcShort(0x5), /* 5 */
#ifdef _ALPHA_
        /* 64 */ NdrFcShort(0xc), /* x86, MIPS,
PPC Stack size/offset = 12 */
#else
        NdrFcShort(0x18), /*
Alpha Stack size/offset = 24 */
#endif
        /* 66 */ NdrFcShort(0x8), /* 8 */
        /* 68 */ NdrFcShort(0x10), /* 16 */
        /* 70 */ 0x4, /* Oi2 Flags: has return,
*/
        0x2, /* 2 */

        /* Parameter order_status_struct */

        /* 72 */ NdrFcShort(0x158), /* Flags: in,
out, base type, simple ref, */
#ifdef _ALPHA_
        /* 74 */ NdrFcShort(0x4), /* x86, MIPS,
PPC Stack size/offset = 4 */
#else

```

```

        NdrFcShort(0x8), /*
Alpha Stack size/offset = 8 */
#endif
        /* 76 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        /* Return value */

        /* 78 */ NdrFcShort(0x70), /* Flags: out,
return, base type, */
#ifdef _ALPHA_
        /* 80 */ NdrFcShort(0x8), /* x86, MIPS,
PPC Stack size/offset = 8 */
#else
        NdrFcShort(0x10), /*
Alpha Stack size/offset = 16 */
#endif
        /* 82 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        /* Procedure ExStockLevel */
        /* 84 */ 0x33, /* FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object,
Oi2 */
        /* 86 */ NdrFcLong(0x0), /* 0 */
        /* 90 */ NdrFcShort(0x6), /* 6 */
#ifdef _ALPHA_
        /* 92 */ NdrFcShort(0xc), /* x86, MIPS,
PPC Stack size/offset = 12 */
#else
        NdrFcShort(0x18), /*
Alpha Stack size/offset = 24 */
#endif
        /* 94 */ NdrFcShort(0x8), /* 8 */
        /* 96 */ NdrFcShort(0x10), /* 16 */
        /* 98 */ 0x4, /* Oi2 Flags: has return,
*/
        0x2, /* 2 */

        /* Parameter stock_level_struct */
        /* 100 */ NdrFcShort(0x158), /* Flags: in,
out, base type, simple ref, */
#ifdef _ALPHA_
        /* 102 */ NdrFcShort(0x4), /* x86, MIPS,
PPC Stack size/offset = 4 */
#else
        NdrFcShort(0x8), /*
Alpha Stack size/offset = 8 */
#endif
        /* 104 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        /* Return value */
        /* 106 */ NdrFcShort(0x70), /* Flags: out,
return, base type, */
#ifdef _ALPHA_
        /* 108 */ NdrFcShort(0x8), /* x86, MIPS,
PPC Stack size/offset = 8 */
#else
        NdrFcShort(0x10), /*
Alpha Stack size/offset = 16 */
#endif
        /* 110 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        /* Procedure ExDelivery */
        /* 112 */ 0x33, /* FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object,
Oi2 */
        /* 114 */ NdrFcLong(0x0), /* 0 */
        /* 118 */ NdrFcShort(0x7), /* 7 */
#ifdef _ALPHA_

```

```

/* 120 */ NdrFcShort(0xc), /* x86, MIPS,
PPC Stack size/offset = 12 */
#else
    NdrFcShort(0x18), /*
Alpha Stack size/offset = 24 */
#endif
/* 122 */ NdrFcShort(0x8), /* 8 */
/* 124 */ NdrFcShort(0x10), /* 16 */
/* 126 */ 0x4, /* Oi2 Flags: has return,
*/
    0x2, /* 2 */
/* Parameter delivery_struct */
/* 128 */ NdrFcShort(0x158), /* Flags: in,
out, base type, simple ref, */
#ifdef _ALPHA_
/* 130 */ NdrFcShort(0x4), /* x86, MIPS,
PPC Stack size/offset = 4 */
#else
    NdrFcShort(0x8), /*
Alpha Stack size/offset = 8 */
#endif
/* 132 */ 0x8, /* FC_LONG */
    0x0, /* 0 */
/* Return value */
/* 134 */ NdrFcShort(0x70), /* Flags: out,
return, base type, */
#ifdef _ALPHA_
/* 136 */ NdrFcShort(0x8), /* x86, MIPS,
PPC Stack size/offset = 8 */
#else
    NdrFcShort(0x10), /*
Alpha Stack size/offset = 16 */
#endif
/* 138 */ 0x8, /* FC_LONG */
    0x0, /* 0 */

/* Procedure EntryTrans */

/* 140 */ 0x33, /* FC_AUTO_HANDLE */
    0x6c, /* Old Flags: object,
Oi2 */
/* 142 */ NdrFcLong(0x0), /* 0 */
/* 146 */ NdrFcShort(0x8), /* 8 */
#ifdef _ALPHA_
/* 148 */ NdrFcShort(0x8), /* x86, MIPS,
PPC Stack size/offset = 8 */
#else
    NdrFcShort(0x10), /*
Alpha Stack size/offset = 16 */
#endif
/* 150 */ NdrFcShort(0x0), /* 0 */
/* 152 */ NdrFcShort(0x8), /* 8 */
/* 154 */ 0x4, /* Oi2 Flags: has return,
*/
    0x1, /* 1 */

/* Return value */

/* 156 */ NdrFcShort(0x70), /* Flags: out,
return, base type, */
#ifdef _ALPHA_
/* 158 */ NdrFcShort(0x4), /* x86, MIPS,
PPC Stack size/offset = 4 */
#else
    NdrFcShort(0x8), /*
Alpha Stack size/offset = 8 */
#endif
/* 160 */ 0x8, /* FC_LONG */
    0x0, /* 0 */

```

```

/* Procedure TermSvrApl */

/* 162 */ 0x33, /* FC_AUTO_HANDLE */
    0x6c, /* Old Flags: object,
Oi2 */
/* 164 */ NdrFcLong(0x0), /* 0 */
/* 168 */ NdrFcShort(0x9), /* 9 */
#ifdef _ALPHA_
/* 170 */ NdrFcShort(0x8), /* x86, MIPS,
PPC Stack size/offset = 8 */
#else
    NdrFcShort(0x10), /*
Alpha Stack size/offset = 16 */
#endif
/* 172 */ NdrFcShort(0x0), /* 0 */
/* 174 */ NdrFcShort(0x8), /* 8 */
/* 176 */ 0x4, /* Oi2 Flags: has return,
*/
    0x1, /* 1 */

/* Return value */

/* 178 */ NdrFcShort(0x70), /* Flags: out,
return, base type, */
#ifdef _ALPHA_
/* 180 */ NdrFcShort(0x4), /* x86, MIPS,
PPC Stack size/offset = 4 */
#else
    NdrFcShort(0x8), /*
Alpha Stack size/offset = 8 */
#endif
/* 182 */ 0x8, /* FC_LONG */
    0x0, /* 0 */

/* Procedure InitSvrApl */

/* 184 */ 0x33, /* FC_AUTO_HANDLE */
    0x6c, /* Old Flags: object,
Oi2 */
/* 186 */ NdrFcLong(0x0), /* 0 */
/* 190 */ NdrFcShort(0xa), /* 10 */
#ifdef _ALPHA_
/* 192 */ NdrFcShort(0x8), /* x86, MIPS,
PPC Stack size/offset = 8 */
#else
    NdrFcShort(0x10), /*
Alpha Stack size/offset = 16 */
#endif
/* 194 */ NdrFcShort(0x0), /* 0 */
/* 196 */ NdrFcShort(0x8), /* 8 */
/* 198 */ 0x4, /* Oi2 Flags: has return,
*/
    0x1, /* 1 */

/* Return value */

/* 200 */ NdrFcShort(0x70), /* Flags: out,
return, base type, */
#ifdef _ALPHA_
/* 202 */ NdrFcShort(0x4), /* x86, MIPS,
PPC Stack size/offset = 4 */
#else
    NdrFcShort(0x8), /*
Alpha Stack size/offset = 8 */
#endif
/* 204 */ 0x8, /* FC_LONG */
    0x0, /* 0 */

```

```

}
};

static const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString =
{
    0,
    {
        NdrFcShort(0x0), /* 0 */
/* 2 */
        0x11, 0x8, /* FC_RP
[simple_pointer] */
/* 4 */ 0x8, /* FC_LONG */
        0x5c, /* FC_PAD */
        0x0
    }
};

const CInterfaceProxyVtbl *
_SvrAplBase_ProxyVtblList[] =
{
    (CInterfaceProxyVtbl *) &_ISvrAplProxyVtbl,
    0
};

const CInterfaceStubVtbl *
_SvrAplBase_StubVtblList[] =
{
    (CInterfaceStubVtbl *) &_ISvrAplStubVtbl,
    0
};

PCInterfaceName const
_SvrAplBase_InterfaceNamesList[] =
{
    "ISvrApl",
    0
};

#define _SvrAplBase_CHECK_IID(n)
IID_GENERIC_CHECK_IID(_SvrAplBase,
pIID, n)
int __stdcall _SvrAplBase_IID_Lookup( const IID *
pIID, int * pIndex )
{
    if(!_SvrAplBase_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }
    return 0;
}

const ExtendedProxyFileInfo
SvrAplBase_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) &
_SvrAplBase_ProxyVtblList,
    (PCInterfaceStubVtblList *) &
_SvrAplBase_StubVtblList,
    (const PCInterfaceName *) &
_SvrAplBase_InterfaceNamesList,
    0, // no delegation
    &_SvrAplBase_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
};
.....

```

```

SvrAplBase/SvrAplBaseps.def
.....
LIBRARY "SvrAplBasePS"
DESCRIPTION 'Proxy/Stub DLL'
EXPORTS
    DllGetClassObject @1 PRIVATE
    DllCanUnloadNow @2 PRIVATE
    GetProxyDllInfo @3 PRIVATE
    DllRegisterServer @4 PRIVATE
    DllUnregisterServer @5 PRIVATE
.....
SvrAplBase/SvrAplBaseCode.h
.....
/*****
*
* TPC-C Client Application Program Source
*
* CREATE by TSL 2002.12.18
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****
**/
#define ERR_INIT_INCOMPLETE 0x80040200
/* Initialize incomplete */
#define ERR_NO_EXEC 0x80040201 /* Transaction no execute */
#define ERR_CNTL 0x80040202 /* Control error */
#define ERR_NO_ENTRY 0x80040210 /* Nothing DLL entry point */
#define ERR_TRANS_PROC 0x800402ff /* Transaction proc. error */

.....
SvrAplBase/TermSvrAplCntl.cpp
.....
/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* TermSvrAplCntl : Terminae control
process
*
* CREATE by TSL 2002.12.18
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****
**/
#include "stdafx.h"
#include <process.h>
#include "SvrAplBase.h"
#include "ProcCntl.h"
#include "SvrApl.h"
#include "log.h"

```

```

#include "MapFileInfo.h"
#include "ProcCntlCmd.h"
#include "GlobalArea.h"

#ifndef PROC_MODE /////// Executing mode is
PRCODESS. ///////

void TermSvrAplCntl() {
    int ret_val;
    int i, j;
    int num_term_proc;
    PROC_CNTL_INFO* proc_cntl_info;

    MAC_PutFuncEntryLog("CSvrApl::TermSvrA
plCntl");
    if (GLB_TermnateProcessing !=
TRMPROC_NOTERMINATE) {
        /* Not first terminate */
        return;
    }

    /* First time, terminate all control process */

    /* All process terminate */
    proc_cntl_info = GLB_ProcCntlInfo;
    for(i = 0; i < GLB_NumControlProcesses;
i++) {

        /* Send terminate command to control
process */
        if ((ret_val =
RequestProc(COMMAND_TERMINATE, (void*)0,
0, proc_cntl_info) != 0) {
            userlog(LOG_ERR,
"RequestProc() faild rc=%d", ret_val);
            /* continue next process
terminating */
        }
        proc_cntl_info++;

        /* Wait all process ended */
        j = 0;

        while (1) {
            num_term_proc = 0;
            proc_cntl_info = GLB_ProcCntlInfo;
            for(i = 0; i <
GLB_NumControlProcesses; i++) {
                if (proc_cntl_info-
>process_status == PROC_STAT_TERM)
                    num_term_proc++;
                proc_cntl_info++;
            }
            if (num_term_proc ==
GLB_NumControlProcesses)
                break; /* All process
terminated */

            Sleep(100);
            if (++j >= 6000) {
                /* Timeout = 10min */

```

```

                userlog(LOG_ERR, "Control
process terminating timeout (%d/%d ended)",
num_term_proc, GLB_NumControlProcesses);
                break;
            }
        }
        /* Close shared mapping (Contorl table) */
        if (GLB_ProcCntlInfo != 0) {
            if
(!UnmapViewOfFile(GLB_ProcCntlInfo))
                userlog(LOG_ERR,
"Unmapping failed GetLastError=%d",
GetLastError());
            GLB_ProcCntlInfo = 0;
        }

        if (GLB_MappingHandle != 0) {
            if
(!CloseHandle(GLB_MappingHandle))
                userlog(LOG_ERR, "Mapping
handle close failed GetLastError=%d",
GetLastError());
            GLB_MappingHandle = 0;
        }
        /* Close event for process idle posting */
        if (GLB_ProcIdleEvent != 0) {
            if (CloseHandle(GLB_ProcIdleEvent)
== 0)
                userlog(LOG_ERR, "Process
idle event release failed GetLastError=%d",
GetLastError());
            GLB_ProcIdleEvent = 0;
        }
        /* Set terminate end, only one time
executing */
        GLB_TermnateProcessing =
TRMPROC_TERMINATE;
        userlog(LOG_INF, "Control process
terminate ended");
        return;
    }
}
#endif ///////
.....
SvrAplBase/WaitProcessEnd.cpp
.....
/*****
*
* TPC-C Client Application Program Source
*
* Entry Functions
* WaitProcessEnd : Waiting data processing
end
*
* CREATE by TSL 2002.12.18
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2002 *
*****
**/
#include "stdafx.h"
#include "SvrAplBase.h"
#include "ProcCntl.h"
#include "SvrApl.h"
#include "log.h"

```

```
#include "MapFileInfo.h"
#ifdef PROC_MODE //!!!! Executing mode is
PRCODESS. //!!!!
int WaitProcessEnd(void* proc_result,
unsigned long proc_len, PROC_CNTL_INFO*
proc_cntl_ifo) {
    DWORD ret_val;
    MAC_PutFncEntryLog("WaitProcessEnd");
    /* Wait control process proceeding end */
    if ((ret_val =
WaitForSingleObject(proc_cntl_ifo-
>response_event, INFINITE)) !=
WAIT_OBJECT_0) {
        userlog(LOG_ERR, "Illegal restart by
WaitForSingleObject() ret_val=%d
GetLastError=%d",
ret_val,
GetLastError());
        return -1;
    }
    /* Set Processing data */
    if (proc_result != 0) {
        /* set processing data to shared
memory */
        memcpy(proc_result,
(char*)proc_cntl_ifo->shared_memory +
MAP_DATA_OFFSET, proc_len);
    }

    return 0;
}

#endif

.....
SvrAplBase/dlldata.c
.....
/*****
DllData file -- generated by MIDL compiler

DO NOT ALTER THIS FILE

This file is regenerated by MIDL on every IDL
file compile.

To completely reconstruct this file, delete it and
rerun MIDL
on all the IDL files in this DLL, specifying this file
for the
/dlldata command line option

*****/

#include <rpcproxy.h>

#ifdef __cplusplus
extern "C" {
#endif

EXTERN_PROXY_FILE(SvrAplBase)

PROXYFILE_LIST_START
/* Start of list */
REFERENCE_PROXY_FILE(SvrAplBase),
/* End of list */
PROXYFILE_LIST_END
```

```
DLldata_ROUTINES( aProxyFileList,
GET_DLL_CLSID )

#ifdef __cplusplus
} /*extern "C" */
#endif

/* end of generated dlldata file */

.....
SvrAplBase/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)
userlog(LOG_INF, ">>>>> "func" start
>>>>>");
#else
#define MAC_PutFncEntryLog(func) ;
#endif

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

.....
SvrAplBase/resource.h
.....
/!(NO_DEPENDENCIES)
// Microsoft Developer Studio generated include
file.
// Used by SvrAplBase.rc
//
#define IDS_PROJNAME 100
#define IDR_SVRAPL 101

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
```

```
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE
201
#define _APS_NEXT_COMMAND_VALUE
32768
#define _APS_NEXT_CONTROL_VALUE
201
#define _APS_NEXT_SYMED_VALUE 102
#endif
.....
SvrDBDepend_Oracle/DbgLog.cpp
.....
/* Copyright FUJITSU LIMITED (c) 2003 */
/* Logger for Debug */
#include <Windows.h>
#include "tpcc.h"
#include "GlobalArea.h"
void DbgLog(char* form_data, int arg)
{
    char log_form[256];
#ifdef DGLCOM
    FILE* log_file;
    char file_name[MAX_PATH];
    char file_path[MAX_PATH];
#endif
    /* Make format string */
    strcpy(log_form, "THREADID[%06d] ");
    strcat(log_form, form_data);
    strcat(log_form, "\n");
#ifdef DGLCOM
    /* output user log on the TUXLOG */
    userlog (LOG_FILE_LINE, log_form,
GetCurrentThreadId(), arg);
#else
    sprintf(file_path,
LOG_FILE_NAME_THREAD,
GetCurrentThreadId());
    strcpy(file_name, GLB_DllPath);
    strcat(file_name, file_path);
    log_file = fopen(file_name, "a");
    fprintf(log_file, log_form,
GetCurrentThreadId(), arg);
    fclose(log_file);
#endif
    return;
}
.....
SvrDBDepend_Oracle/DllMain.cpp
.....
/*****
***
* TPC-C Client Application Program Source
*
* Entry Functions
* DLLMain
*
* CREATE by TSL 2003.05.16
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
**/
#include <windows.h>
#include "tpcc.h"
```

```

#include "tpcc_info.h"
#include "log.h"
#include "tpccsvr_Obj.h"
//extern "C"
BOOL WINAPI DIIMain(HINSTANCE hInstance,
DWORD dwReason, LPVOID lpReserved)
{
    int         retval;
    lpReserved;
    if (dwReason == DLL_PROCESS_ATTACH)
    {
        GetOwnDllPath(hInstance);
        userlog(LOG_FILE_LINE, "===
SvrDBDepend.dll Attached dll path [%s] ===\n",
GLB_DllPath);
        if
(SetCurrentDirectory(GLB_DllPath)) {
            /* Changing current path failed
*/
            userlog(LOG_FILE_LINE,
"SvrDBDepend.dll Changing current path
failed[%s]\n", GLB_DllPath);
            userlog(LOG_FILE_LINE, "
GetLastError[%d]\n", GetLastError());
            return FALSE;
        }
        /* Allocate TLS index */
        GLB_TlsIndex = TlsAlloc();
        if(GLB_TlsIndex == 0xFFFFFFFF){
            /* Failed TlsAlloc */
            userlog(LOG_FILE_LINE,
"Failed TlsAlloc\n");
            return FALSE;
        }
    }
    if (dwReason == DLL_PROCESS_DETACH)
    {
        /* Free TLS index */
        retval = TlsFree(GLB_TlsIndex);
        if (retval == 0) {
            userlog(LOG_FILE_LINE,
"Failed TlsFree\n");
            return FALSE;
        }
    }
    return TRUE; // ok
}
/* Get Own DLL path */
void GetOwnDllPath(HINSTANCE hInstance) {
    int         i;
    GetModuleFileName(hInstance,
GLB_DllPath, sizeof(GLB_DllPath)-1);
    for(i = strlen(GLB_DllPath) - 1;
GLB_DllPath[i] != '\\; i--);
    GLB_DllPath[i+1] = '\\0';
    strcpy(GLB_LogFilePath, GLB_DllPath);
    strcat(GLB_LogFilePath,
LOG_FILE_NAME);
}
.....
SvrDBDepend_Oracle/GlobalArea.cpp
.....
/*
*
* 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 *
.....
**/
#include <windows.h>
/* Thread Local Storage key */
DWORD GLB_TlsIndex;
char GLB_LogFilePath[MAX_PATH];
char GLB_DllPath[MAX_PATH];
.....
SvrDBDepend_Oracle/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 *
.....
**/
/* Thread Local Storage key */
extern DWORD GLB_TlsIndex;
extern char GLB_LogFilePath[MAX_PATH];
extern char GLB_DllPath[MAX_PATH];
#define LOG_FILE_NAME_THREAD
"log\\SvrThread%05d.log"
extern void GetOwnDllPath(HINSTANCE
hInstance);
.....
SvrDBDepend_Oracle/SvrApiObject.cpp
.....
/*
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* SvrApi object operation.
*
* CREATE by TSL 2003.05.16
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
.....
**/
#include <windows.h>
#include "tpcc.h"
#include "log.h"
#include "tpccsvr_Obj.h"
CDBAccess::CDBAccess(void)
{

```

```

    proc_no = 0;
    logon = 0;
    new_init = 0;
    pay_init = 0;
    ord_init = 0;
#ifdef DEL_ORA81
    del_init = 0;
#else
    del_init_oci = 0;
    del_init_plsql = 0;
#endif
    sto_init = 0;
    res_init = 0;
    tracelevel = 0;
    lfp = NULL;
}
CDBAccess::~CDBAccess(void)
{
}
// -----
// Create CDBAccess class
// -----
void* ExSvrApiCreate()
{
    CDBAccess* CIsPt;
    int         retval;
    CIsPt = new CDBAccess;
    if (CIsPt == 0) {
        userlog(LOG_FILE_LINE, "Object
creation failure.");
        return (void *)0;
    }
    retval = TlsSetValue(GLB_TlsIndex, (void
*)CIsPt);
    if (retval == 0) {
        userlog(LOG_FILE_LINE, "Failed
TlsGetValue.");
        return (void *)0;
    }
    /* Check Test Log */
    userlog(LOG_FILE_LINE, "Success to
create SvrApi Object (pointer=%08x)\n", CIsPt);
    return (void *)CIsPt;
}
// -----
// Delete CDBAccess class
// -----
void ExSvrApiDelete(void* prmPt)
{
    delete prmPt;
    return;
}
.....
SvrDBDepend_Oracle/SvrDBDepend.mak
.....
# Microsoft Developer Studio Generated NMAKE
File, Based on SvrDBDepend.dsp
!IF "$(CFG)" == ""
CFG=SvrDBDepend - Win32 Debug
MESSAGE SvrDBDepend - Win32 Debug
!ENDIF
!IF "$(CFG)" != "SvrDBDepend - Win32 Release"
&& "$(CFG)" != "SvrDBDepend - Win32 Debug"
MESSAGE "$(CFG)"
MESSAGE NMAKE
MESSAGE :
MESSAGE
MESSAGE NMAKE /f "SvrDBDepend.mak"
CFG="SvrDBDepend - Win32 Debug"

```

```

!MESSAGE
!MESSAGE :
!MESSAGE
!MESSAGE "SvrDBDepend - Win32 Release"
("Win32 (x86) Dynamic-Link Library")
!MESSAGE "SvrDBDepend - Win32 Debug"
("Win32 (x86) Dynamic-Link Library")
!MESSAGE
!ERROR
!ENDIF
!IF "$(OS)" == "Windows_NT"
NULL=
!ELSE
NULL=nul
!ENDIF
!IF "$(CFG)" == "SvrDBDepend - Win32 Release"
OUTDIR=.Release
INTDIR=.Release
# Begin Custom Macros
OutDir=.Release
# End Custom Macros
ALL : "$(OUTDIR)\SvrDBDepend.dll"
"$$(OUTDIR)\SvrDBDepend.bsc"
CLEAN :
-@erase "$(INTDIR)\DbgLog.obj"
-@erase "$(INTDIR)\DbgLog.sbr"
-@erase "$(INTDIR)\DIIMain.obj"
-@erase "$(INTDIR)\DIIMain.sbr"
-@erase "$(INTDIR)\GlobalArea.obj"
-@erase "$(INTDIR)\GlobalArea.sbr"
-@erase "$(INTDIR)\log.obj"
-@erase "$(INTDIR)\log.sbr"
-@erase "$(INTDIR)\pldel.obj"
-@erase "$(INTDIR)\pldel.sbr"
-@erase "$(INTDIR)\plnew.obj"
-@erase "$(INTDIR)\plnew.sbr"
-@erase "$(INTDIR)\plord.obj"
-@erase "$(INTDIR)\plord.sbr"
-@erase "$(INTDIR)\plpay.obj"
-@erase "$(INTDIR)\plpay.sbr"
-@erase "$(INTDIR)\plsto.obj"
-@erase "$(INTDIR)\plsto.sbr"
-@erase "$(INTDIR)\SvrApiObject.obj"
-@erase "$(INTDIR)\SvrApiObject.sbr"
-@erase "$(INTDIR)\tpccpl.obj"
-@erase "$(INTDIR)\tpccpl.sbr"
-@erase "$(INTDIR)\tpccsvr.obj"
-@erase "$(INTDIR)\tpccsvr.sbr"
-@erase "$(INTDIR)\vc60.idb"
-@erase "$(OUTDIR)\SvrDBDepend.bsc"
-@erase "$(OUTDIR)\SvrDBDepend.dll"
-@erase "$(OUTDIR)\SvrDBDepend.exp"
-@erase "$(OUTDIR)\SvrDBDepend.lib"
"$(OUTDIR)" :
if not exist "$(OUTDIR)\$(NULL)" mkdir
"$(OUTDIR)"
CPP=cl.exe
CPP_PROJ=/nologo /MD /W4 /GX /O2 /I
"../oracle/oci/include" /D "XTSL" /D
PLSQLFLAG=1 /D "WIN32" /D "_WINDOWS" /D
"_MBCS" /D "_USRDL" /D
"SVRDBDEPEND_EXPORTS" /FR "$(INTDIR)\
\Fp "$(INTDIR)\SvrDBDepend.pch" /YX
/Fo "$(INTDIR)\F" /Fd "$(INTDIR)\FD /c
.c $(INTDIR).obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cpp $(INTDIR).obj:

```

```

$(CPP) @<<
$(CPP_PROJ) $<
<<
.cxx $(INTDIR).obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.c $(INTDIR).sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cpp $(INTDIR).sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cxx $(INTDIR).sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
MTL=midl.exe
MTL_PROJ=/nologo /D "NDEBUG" /mktylib203
/win32
RSC=rc.exe
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/o "$(OUTDIR)\SvrDBDepend.bsc"
BSC32_SBRS= \
"$(INTDIR)\DbgLog.sbr" \
"$(INTDIR)\DIIMain.sbr" \
"$(INTDIR)\GlobalArea.sbr" \
"$(INTDIR)\log.sbr" \
"$(INTDIR)\pldel.sbr" \
"$(INTDIR)\plnew.sbr" \
"$(INTDIR)\plord.sbr" \
"$(INTDIR)\plpay.sbr" \
"$(INTDIR)\plsto.sbr" \
"$(INTDIR)\SvrApiObject.sbr" \
"$(INTDIR)\tpccpl.sbr" \
"$(INTDIR)\tpccsvr.sbr"
"$(OUTDIR)\SvrDBDepend.bsc" : "$(OUTDIR)"
$(BSC32)_SBRS
$(BSC32) @<<
$(BSC32_FLAGS) $(BSC32_SBRS)
<<
LINK32=link.exe
LINK32_FLAGS=oci.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib
shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib
odbccp32.lib /nologo /dll /incremental:no
/pdb:"$(OUTDIR)\SvrDBDepend.pdb"
/machine:i386 /def:"\svrapl.def"
/out:"$(OUTDIR)\SvrDBDepend.dll"
/implib:"$(OUTDIR)\SvrDBDepend.lib"
/libpath:"../oracle/OCI/LIB/MSVC"
DEF_FILE= \
"\svrapl.def"
LINK32_OBJS= \
"$(INTDIR)\DbgLog.obj" \
"$(INTDIR)\DIIMain.obj" \
"$(INTDIR)\GlobalArea.obj" \
"$(INTDIR)\log.obj" \
"$(INTDIR)\pldel.obj" \
"$(INTDIR)\plnew.obj" \
"$(INTDIR)\plord.obj" \
"$(INTDIR)\plpay.obj" \
"$(INTDIR)\plsto.obj" \
"$(INTDIR)\SvrApiObject.obj" \
"$(INTDIR)\tpccpl.obj" \
"$(INTDIR)\tpccsvr.obj"

```

```

"$(OUTDIR)\SvrDBDepend.dll" : "$(OUTDIR)"
$(DEF_FILE) $(LINK32_OBJS)
$(LINK32) @<<
$(LINK32_FLAGS) $(LINK32_OBJS)
<<
SOURCE="$(InputPath)"
PostBuild_Desc=bin
DS_POSTBUILD_DEP=$(INTDIR)\postbld.dep
ALL : $(DS_POSTBUILD_DEP)
# Begin Custom Macros
OutDir=.Release
# End Custom Macros
$(DS_POSTBUILD_DEP) :
"$$(OUTDIR)\SvrDBDepend.dll"
"$$(OUTDIR)\SvrDBDepend.bsc"

copy .Release\SvrDBDepend.dll ..\..\bin\Releas
e\THREAD_SvrDBDepend.dll_3tier
echo Helper for Post-build step >
"$$(DS_POSTBUILD_DEP)"
!ELSEIF "$(CFG)" == "SvrDBDepend - Win32
Debug"
OUTDIR=.Debug
INTDIR=.Debug
# Begin Custom Macros
OutDir=.Debug
# End Custom Macros
ALL : "$(OUTDIR)\SvrDBDepend.dll"
"$$(OUTDIR)\SvrDBDepend.bsc"
CLEAN :
-@erase "$(INTDIR)\DbgLog.obj"
-@erase "$(INTDIR)\DbgLog.sbr"
-@erase "$(INTDIR)\DIIMain.obj"
-@erase "$(INTDIR)\DIIMain.sbr"
-@erase "$(INTDIR)\GlobalArea.obj"
-@erase "$(INTDIR)\GlobalArea.sbr"
-@erase "$(INTDIR)\log.obj"
-@erase "$(INTDIR)\log.sbr"
-@erase "$(INTDIR)\pldel.obj"
-@erase "$(INTDIR)\pldel.sbr"
-@erase "$(INTDIR)\plnew.obj"
-@erase "$(INTDIR)\plnew.sbr"
-@erase "$(INTDIR)\plord.obj"
-@erase "$(INTDIR)\plord.sbr"
-@erase "$(INTDIR)\plpay.obj"
-@erase "$(INTDIR)\plpay.sbr"
-@erase "$(INTDIR)\plsto.obj"
-@erase "$(INTDIR)\plsto.sbr"
-@erase "$(INTDIR)\SvrApiObject.obj"
-@erase "$(INTDIR)\SvrApiObject.sbr"
-@erase "$(INTDIR)\tpccpl.obj"
-@erase "$(INTDIR)\tpccpl.sbr"
-@erase "$(INTDIR)\tpccsvr.obj"
-@erase "$(INTDIR)\tpccsvr.sbr"
-@erase "$(INTDIR)\vc60.idb"
-@erase "$(OUTDIR)\SvrDBDepend.bsc"
-@erase "$(OUTDIR)\SvrDBDepend.dll"
-@erase "$(OUTDIR)\SvrDBDepend.exp"
-@erase "$(OUTDIR)\SvrDBDepend.lib"
-@erase "$(OUTDIR)\SvrDBDepend.pdb"
"$(OUTDIR)" :
if not exist "$(OUTDIR)\$(NULL)" mkdir
"$(OUTDIR)"
CPP=cl.exe
CPP_PROJ=/nologo /MTd /W4 /Gm /GX /ZI /Od /I
"../oracle/oci/include" /D "_DEBUG" /D
PLSQLFLAG=1 /D "WIN32" /D "_WINDOWS" /D

```

```

"_MBCS" /D "_USRDL" /D
"SVRDBDEPEND_EXPORTS" /Fr"${(INTDIR)}\
\Fp"${(INTDIR)\SvrDBDepend.pch" /YX
\Fo"${(INTDIR)}\ /Fd"${(INTDIR)}\ /FD /GZ /c
.c("${(INTDIR)}.obj":
$(CPP) @<<
$(CPP_PROJ) <<
<<
.cpp("${(INTDIR)}.obj":
$(CPP) @<<
$(CPP_PROJ) <<
<<
.cxx("${(INTDIR)}.obj":
$(CPP) @<<
$(CPP_PROJ) <<
<<
.cpp("${(INTDIR)}.sbr":
$(CPP) @<<
$(CPP_PROJ) <<
<<
.cxx("${(INTDIR)}.sbr":
$(CPP) @<<
$(CPP_PROJ) <<
<<
MTL=midl.exe
MTL_PROJ=/nologo /D "_DEBUG" /mktyplib203
/win32
RSC=rc.exe
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/o"${(OUTDIR)\SvrDBDepend.bsc"
BSC32_SBRS= \
    "${(INTDIR)\DbgLog.sbr" \
    "${(INTDIR)\DIIMain.sbr" \
    "${(INTDIR)\GlobalArea.sbr" \
    "${(INTDIR)\log.sbr" \
    "${(INTDIR)\pldel.sbr" \
    "${(INTDIR)\plnew.sbr" \
    "${(INTDIR)\plord.sbr" \
    "${(INTDIR)\plpay.sbr" \
    "${(INTDIR)\plsto.sbr" \
    "${(INTDIR)\SvrAplObject.sbr" \
    "${(INTDIR)\tpccpl.sbr" \
    "${(INTDIR)\tpccsvr.sbr"
"${(OUTDIR)\SvrDBDepend.bsc" : "${(OUTDIR)"
$(BSC32_SBRS)
$(BSC32) @<<
$(BSC32_FLAGS) $(BSC32_SBRS)
<<
LINK32=link.exe
LINK32_FLAGS=oci.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib
shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib
odbccp32.lib /nologo /dll /incremental:yes
/pdb:"${(OUTDIR)\SvrDBDepend.pdb" /debug
/machine:i386 /def:".svrapl.def"
/out:"${(OUTDIR)\SvrDBDepend.dll"
/implib:"${(OUTDIR)\SvrDBDepend.lib"
/pdbtype:sept /libpath:"../oracle/OCI/LIB/MSVC"
DEF_FILE= \
    ".svrapl.def"
LINK32_OBJS= \
    "${(INTDIR)\DbgLog.obj" \
    "${(INTDIR)\DIIMain.obj" \
    "${(INTDIR)\GlobalArea.obj" \

```

```

"${(INTDIR)\log.obj" \
"${(INTDIR)\pldel.obj" \
"${(INTDIR)\plnew.obj" \
"${(INTDIR)\plord.obj" \
"${(INTDIR)\plpay.obj" \
"${(INTDIR)\plsto.obj" \
"${(INTDIR)\SvrAplObject.obj" \
"${(INTDIR)\tpccpl.obj" \
"${(INTDIR)\tpccsvr.obj"
"${(OUTDIR)\SvrDBDepend.dll" : "${(OUTDIR)"
$(DEF_FILE) $(LINK32_OBJS)
$(LINK32) @<<
$(LINK32_FLAGS) $(LINK32_OBJS)
<<
SOURCE="$(InputPath)"
PostBuild_Desc=bin
DS_POSTBUILD_DEP=${(INTDIR)\postbld.dep
ALL : $(DS_POSTBUILD_DEP)
# Begin Custom Macros
OutDir=.Debug
# End Custom Macros
$(DS_POSTBUILD_DEP) :
"${(OUTDIR)\SvrDBDepend.dll"
"${(OUTDIR)\SvrDBDepend.bsc"
copy .Debug\SvrDBDepend.dll ..\..\bin\Debug\S
vrDBDepend.dll_3tier
echo Helper for Post-build step >
"${(DS_POSTBUILD_DEP)"
!ENDIF
!!IF "$(NO_EXTERNAL_DEPS)" != "1"
!!IF EXISTS("SvrDBDepend.dep")
!!INCLUDE "SvrDBDepend.dep"
!!ELSE
!!MESSAGE Warning: cannot find
"SvrDBDepend.dep"
!ENDIF
!ENDIF
!!IF "$(CFG)" == "SvrDBDepend - Win32 Release"
|| "$(CFG)" == "SvrDBDepend - Win32 Debug"
SOURCE=.DbgLog.cpp
"${(INTDIR)\DbgLog.obj" "${(INTDIR)\DbgLog.sbr" :
$(SOURCE) "${(INTDIR)"
SOURCE=.DIIMain.cpp
"${(INTDIR)\DIIMain.obj" "${(INTDIR)\DIIMain.sbr" :
$(SOURCE) "${(INTDIR)"
SOURCE=.GlobalArea.cpp
"${(INTDIR)\GlobalArea.obj"
"${(INTDIR)\GlobalArea.sbr" : $(SOURCE)
"${(INTDIR)"
SOURCE=.log.cpp
"${(INTDIR)\log.obj" "${(INTDIR)\log.sbr" :
$(SOURCE) "${(INTDIR)"
SOURCE=.pldel.cpp
"${(INTDIR)\pldel.obj" "${(INTDIR)\pldel.sbr" :
$(SOURCE) "${(INTDIR)"
SOURCE=.plnew.cpp
"${(INTDIR)\plnew.obj" "${(INTDIR)\plnew.sbr" :
$(SOURCE) "${(INTDIR)"
SOURCE=.plord.cpp
"${(INTDIR)\plord.obj" "${(INTDIR)\plord.sbr" :
$(SOURCE) "${(INTDIR)"
SOURCE=.plpay.cpp
"${(INTDIR)\plpay.obj" "${(INTDIR)\plpay.sbr" :
$(SOURCE) "${(INTDIR)"
SOURCE=.plsto.cpp
"${(INTDIR)\plsto.obj" "${(INTDIR)\plsto.sbr" :
$(SOURCE) "${(INTDIR)"
SOURCE=.SvrAplObject.cpp

```

```

"${(INTDIR)\SvrAplObject.obj"
"${(INTDIR)\SvrAplObject.sbr" : $(SOURCE)
"${(INTDIR)"
SOURCE=.tpccpl.cpp
"${(INTDIR)\tpccpl.obj" "${(INTDIR)\tpccpl.sbr" :
$(SOURCE) "${(INTDIR)"
SOURCE=.tpccsvr.cpp
"${(INTDIR)\tpccsvr.obj" "${(INTDIR)\tpccsvr.sbr" :
$(SOURCE) "${(INTDIR)"
!ENDIF
.....
SvrDBDepend_Oracle\TrnCntrlInfo.h
.....
/*****
***
*
* TPC-C Client Application Program Source
*
*
* Entry Functions
* Transaction structure object definition.
*
*
* CREATE by TSL 2003.05.16
*
*
* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
**/
/* ----- */
/* Delivery Struct */
/* ----- */
struct delctx {
    sb2 del_o_id_ind[NDISTS];
    sb2 d_id_ind[NDISTS];
    sb2 c_id_ind[NDISTS];
    sb2 del_date_ind[NDISTS];
    sb2 carrier_id_ind[NDISTS];
    sb2 amt_ind[NDISTS];
    ub4 del_o_id_len[NDISTS];
    ub4 c_id_len[NDISTS];
    int oid_ctx;
    int cid_ctx;
    OCIBind *olamt_bp;
    ub2 w_id_len[NDISTS];
    ub2 d_id_len[NDISTS];
    ub2 del_date_len[NDISTS];
    ub2 carrier_id_len[NDISTS];
    ub2 amt_len[NDISTS];
    ub2 del_o_id_rcode[NDISTS];
    ub2 cons_rcode[NDISTS];
    ub2 w_id_rcode[NDISTS];
    ub2 d_id_rcode[NDISTS];
    ub2 del_date_rcode[NDISTS];
    ub2 carrier_id_rcode[NDISTS];
    ub2 amt_rcode[NDISTS];
    int del_o_id[NDISTS];
    int del_d_id[NDISTS];
    int cons[NDISTS];
    int w_id[NDISTS];
    int d_id[NDISTS];
    int c_id[NDISTS];
    int carrier_id[NDISTS];
    int amt[NDISTS];
    ub4 del_o_id_rcnt;
    int retry;

```

```

OCIRowid *no_rowid_ptr[NDISTS];
OCIRowid *o_rowid_ptr[NDISTS];
OCIDate del_date[NDISTS];
OCISmt *curd0;
OCISmt *curd1;
OCISmt *curd2;
OCISmt *curd3;
OCISmt *curd4;
OCISmt *curd5;
OCISmt *curd6;
OCISmt *curdtest;
OCIBind *w_id_bp;
OCIBind *w_id_bp3;
OCIBind *w_id_bp4;
OCIBind *w_id_bp5;
OCIBind *w_id_bp6;
OCIBind *d_id_bp;
OCIBind *d_id_bp3;
OCIBind *d_id_bp4;
OCIBind *d_id_bp6;
OCIBind *o_id_bp;
OCIBind *cr_date_bp;
OCIBind *c_id_bp;
OCIBind *c_id_bp3;
OCIBind *no_rowid_bp;
OCIBind *carrier_id_bp;
OCIBind *o_rowid_bp;
OCIBind *del_o_id_bp;
OCIBind *del_o_id_bp3;
OCIBind *amt_bp;
OCIBind *bstr1_bp[10];
OCIBind *bstr2_bp[10];
OCIBind *retry_bp;
OCIDefine *inum_dp;
OCIDefine *d_id_dp;
OCIDefine *del_o_id_dp;
OCIDefine *no_rowid_dp;
OCIDefine *c_id_dp;
OCIDefine *o_rowid_dp;
OCIDefine *cons_dp;
OCIDefine *amt_dp;
int norow;
};
typedef struct delctx delctx;
struct pldelctx {
    ub2 del_d_id_len[NDISTS];
    ub2 del_o_id_len[NDISTS];
    ub2 w_id_len;
    ub2 d_id_len[NDISTS];
    ub2 o_c_id_len[NDISTS];
    ub2 sums_len[NDISTS];
    ub2 carrier_id_len;
    ub2 ordcnt_len;
    ub2 del_date_len;
    int del_o_id[NDISTS];
    int del_d_id[NDISTS];
    int o_c_id[NDISTS];
    /* Replaced T.kato 03.09.09 Oracle10g tool kit */
    #if 0
    /* Replaced T.kato 03.07.18 Replaced New
    Oracle10i tool kit */
    /* int sums[NDISTS]; */
    #ifndef TSL
    ! int sums[NDISTS];
    #else
    ! float sums[NDISTS];
    #endif
    /* Replaced end */
    #endif
};
#ifdef USE_IEEE_NUMBER
    float sums[NDISTS];
#else
    int sums[NDISTS];
#endif
/* Replaced end */
OCIDate del_date;
int carrier_id;
int ordcnt;
ub4 del_o_id_rcnt;
ub4 del_d_id_rcnt;
ub4 o_c_id_rcnt;
ub4 sums_rcnt;
int retry;
OCISmt *curp1;
OCISmt *curp2;
OCIBind *w_id_bp;
OCIBind *d_id_bp;
OCIBind *o_id_bp;
OCIBind *o_c_id_bp;
OCIBind *ordcnt_bp;
OCIBind *sums_bp;
OCIBind *del_date_bp;
OCIBind *carrier_id_bp;
OCIBind *retry_bp;
int norow;
};
typedef struct pldelctx pldelctx;
#ifdef DMLRETDL
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
    #ifndef USE_IEEE_NUMBER
    float s_remote[NITEMS];
    #else
    int s_remote[NITEMS];
    #endif
    /* Replaced end */
    char s_dist_info[NITEMS][25];
};
OCISmt *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;
OCISmt *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;
OCISmt *curo0;
OCISmt *curo1;
OCISmt *curo2;
OCISmt *curo3;
OCISmt *curo4;
OCIBind *c_id_bp;
OCIBind *w_id_bp[4];
OCIBind *d_id_bp[4];
OCIBind *c_last_bp[2];
OCIBind *o_id_bp;
OCIBind *c_rowid_bp;
OCIBind *o_rowid_bp;
OCIDefine *c_rowid_dp;
OCIDefine *c_last_dp[2];
OCIDefine *c_id_dp;
OCIDefine *c_first_dp[2];
OCIDefine *c_middle_dp[2];
OCIDefine *c_balance_dp[2];
OCIDefine *o_rowid_dp[2];
OCIDefine *o_id_dp[2];
OCIDefine *o_entry_d_dp[2];
OCIDefine *o_cr_id_dp[2];
OCIDefine *o_ol_cnt_dp[2];
OCIDefine *ol_d_d_dp;
OCIDefine *ol_i_id_dp;
OCIDefine *ol_supply_w_id_dp;
OCIDefine *ol_quantity_dp;
OCIDefine *ol_amount_dp;
OCIDefine *ol_d_base_dp;
OCIDefine *c_count_dp;
OCIRowid *c_rowid_ptr[100];
OCIRowid *c_rowid_cust;
OCIRowid *o_rowid;
int cs;
int cust_idx;
int norow;
int rcount;
int somerows;
};

typedef struct ordctx ordctx;

struct defctx
{
boolean reexec;
ub4 count;
};
typedef struct defctx defctx;
/* ----- */
/* Payment Struct */
/* ----- */
struct payctx {
OCISmt *curpi;
OCISmt *curp0;
OCISmt *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 {

```

```

OCISmt *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;
.....
SvrDBDepend_Oracle/log.cpp
.....
*****
*
* 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 *
*****
**/
#include <windows.h>
#include <stdio.h>
#include <STDARG.h>
#include <time.h>
#include "log.h"
void userlog (char *file_name, int line_no, char*
fmtp, ...)
{
HANDLE hMutex;
FILE* fp;
int tld;
char* fname;
struct tm *nowtime;
time_t long_time;
va_list va;
hMutex = CreateMutex(NULL, FALSE,
MUTEX_NAME);
WaitForSingleObject(hMutex, INFINITE);
time(&long_time);
nowtime = localtime(&long_time);
tld = GetCurrentThreadId();
// Get file name from a path.
fname = 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 [%4d] %-
32s(%3d) : ",
nowtime->tm_hour,
nowtime->tm_min, nowtime->tm_sec, tld, fname,
line_no);
vfprintf(fp, fmtp, va);
if ((*fmtp + strlen(fmtp) - 1) != '\n')

```

```

        fprintf(fp, "\n");
        va_end(va);
        fclose(fp);
        ReleaseMutex(hMutex);
        CloseHandle(hMutex);
        return;
    }
    .....
    SvrDBDepend_Oracle/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 2002 *
    .....
    **/
    void userlog (char *file_name, int line_no, char*
    ftmp, ...);
    extern char GLB_LogFilePath[MAX_PATH];
    #define MUTEX_NAME "TPCC_SvrAPL"
    #define LOG_FILE_NAME
    "log\DBDepend_Userlog.log"
    #define LOG_FILE_LINE __FILE__,
    __LINE__
    .....
    SvrDBDepend_Oracle/pldel.cpp
    .....
    #ifndef 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 <windows.h>
    #include "log.h"
    #include "tpcc.h"
    #include "tpccsvr_Obj.h"
    #define SQLTXT "BEGIN inittpc.init_del ; END;"
    #define SQLTXT1 "DELETE FROM nord WHERE
    no_d_id = :d_id \

```

```

    AND no_w_id = :w_id and rownum <= 1
    \
    RETURNING no_o_id into :o_id "
    #define SQLTXT3 "UPDATE ordr SET
    o_carrier_id = :carrier_id \
    WHERE o_id = :o_id and o_d_id
    = :d_id and o_w_id = :w_id \
    returning o_c_id into :o_c_id"
    #define SQLTXT4 "UPDATE ordl \
    SET ol_delivery_d = :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 DMLRETDL
    sb4 no_data(dvoid *ctxp, OCIBind *bp, ub4 iter,
    ub4 index,
    dvoid **bufpp, ub4 *alenp, ub1 *piecep,
    dvoid **indpp)
    {
    ctxp;
    bp;
    iter;
    index;
    *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)
    {
    /* Added 03.05.14 Move from object(CDBAccess).
    */
    CDBAccess* obj_pointer;
    /* Added end */
    ctxp;
    bp;
    index;
    /* Replaced 03.05.14 Move from
    object(CDBAccess). */
    #if 0
    ! *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;
    #endif
    /* Get object control information pointer */
    obj_pointer =
    (CDBAccess*)TlsGetValue(GLB_TlsIndex);
    if(obj_pointer == 0) {
    userlog(LOG_FILE_LINE, "Failed TlsGetValue.
    Getting Value is 0. But Continue.");
    }
    *bufpp = &obj_pointer->dctx->del_o_id[iter];
    *indpp = &obj_pointer->dctx->del_o_id_ind[iter];
    obj_pointer->dctx-
    >c_id_len[iter]=sizeof(obj_pointer->dctx->c_id[0]);
    *alenp = &obj_pointer->dctx->c_id_len[iter];
    *rcodepp = &obj_pointer->dctx->c_id_rcode[iter];
    *piecep = OCI_ONE_PIECE;
    /* Replaced end */
    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];

```

```

    obj_pointer->dctx-
    >del_o_id_len[iter]=sizeof(obj_pointer->dctx-
    >del_o_id[0]);
    *alenp = &obj_pointer->dctx->del_o_id_len[iter];
    *rcodepp = &obj_pointer->dctx->del_o_id_rcode-
    >del_o_id_rcode[iter];
    *piecep =OCI_ONE_PIECE;
    /* Replaced end */
    return (OCI_CONTINUE);
    }
    sb4 cid_data(dvoid *ctxp, OCIBind *bp, ub4 iter,
    ub4 index,
    dvoid **bufpp, ub4 **alenp, ub1 *piecep,
    dvoid **indpp, ub2 **rcodepp)
    {
    /* Added 03.05.14 Move from object(CDBAccess).
    */
    CDBAccess* obj_pointer;
    /* Added end */
    ctxp;
    bp;
    index;
    /* Replaced 03.05.14 Move from
    object(CDBAccess). */
    #if 0
    ! *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;
    #endif
    /* Get object control information pointer */
    obj_pointer =
    (CDBAccess*)TlsGetValue(GLB_TlsIndex);
    if(obj_pointer == 0) {
    userlog(LOG_FILE_LINE, "Failed TlsGetValue.
    Getting Value is 0. But Continue.");
    }
    *bufpp = &obj_pointer->dctx->c_id[iter];
    *indpp = &obj_pointer->dctx->c_id_ind[iter];
    obj_pointer->dctx-
    >c_id_len[iter]=sizeof(obj_pointer->dctx->c_id[0]);
    *alenp = &obj_pointer->dctx->c_id_len[iter];
    *rcodepp = &obj_pointer->dctx->c_id_rcode[iter];
    *piecep = OCI_ONE_PIECE;
    /* Replaced end */
    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 *ctx, OCIBind *bp, ub4 iter,
ub4 index,
    dvoid **bufpp, ub4 **alenp, ub1 *piecep,
    dvoid **indpp, ub2 **rcodepp)
{
    bp;
    iter;
    amtctx *actx;
    actx =(amtctx *)ctx;
    *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
CDBAccess::tkvcddinit (int plsqliflag)
{
    text stmbuff[SQL_BUF_SIZE];
    if (plsqliflag)
    {
        pldctx = (pldelctx *) malloc (sizeof(pldelctx));
        DISCARD
        memset(pldctx,(char)0,(ub4)sizeof(pldelctx));
        /* Initialize */
        DISCARD OCISnapAlloc(tpcenv,
(dvoid**) &pldctx->curp1, OCI_HTYPE_STMT, 0,
(dvoid**)0);
        DISCARD sprintf ((char *) stmbuff, SQLTXT);
        DISCARD OCISmtPrepare(pldctx->curp1,
errhp, stmbuff,
            (ub4) strlen((char *)stmbuff),
            OCI_NTV_SYNTAX,
            OCI_DEFAULT);
        DISCARD OCIERROR(errhp,
OCISmtExecute(tpcenv,pldctx-
>curp1,errhp,1,0,NULLP(OCISnapshot),
NULLP(OCISnapshot),
OCI_DEFAULT));
        DISCARD OCISnapAlloc(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",stmbuff);
        #endif
        #if defined(ISO6)
            sqlfile("../blocks/tkvcpdel_iso6.sql",stmbuff);
        #endif
        #if defined(ISO8)
            sqlfile("../blocks/tkvcpdel_iso8.sql",stmbuff);
        #endif
        #else
        /* Replaced 03.05.16 COM+ Client */
        #if 0
        ! sqlfile("../blocks/tkvcpdel.sql",stmbuff);
        #endif
        sqlfile("../blocks/tkvcpdel.sql",stmbuff);
        /* Replaced end */
    }
}

```

```

# endif
DISCARD OCISmtPrepare(pldctx->curp2,
errhp, stmbuff,
    (ub4)strlen((char *)stmbuff),
    OCI_NTV_SYNTAX, OCI_DEFAULT);
OCIBNDPL(pldctx->curp2, pldctx->w_id_bp ,
errhp,":w_id",
    ADR(w_id), SIZ(int), 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));
    OCISnapAlloc(tpcenv, (dvoid**)(&dctx-
>curd1), OCI_HTYPE_STMT, 0,
(dvoid**)0);
    DISCARD sprintf ((char *) stmbuff, "%s",
SQLTXT1);
    DISCARD OCISmtPrepare(dctx->curd1, errhp,
stmbuff,
        strlen((char
*)stmbuff),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 OCISnapAlloc(tpcenv, (dvoid
**)(&dctx->curd3), OCI_HTYPE_STMT,
0, (dvoid**)0);
    DISCARD sprintf ((char *) stmbuff, SQLTXT3);
    DISCARD OCISmtPrepare(dctx->curd3, errhp,
stmbuff, strlen((char *)stmbuff),
        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),
    SOLT_INT, NULL, NULL, NULL);
    OCIBNDRA(dctx->curd3, dctx->d_id_bp3,
errhp, "d_id", dctx->d_id,SIZ(int),
    SOLT_INT,NULL, NULL, NULL);
    OCIBNDRA(dctx->curd3, dctx->del_o_id_bp3,
errhp, "o_id", dctx->del_o_id,
    SIZ(int), SOLT_INT,NULL,NULL,NULL);
    OCIBNDRAD(dctx->curd3, dctx->c_id_bp3,
errhp, "o_c_id", SIZ(int),
    SOLT_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, SOLTXT4);
    DISCARD OCISmtPrepare(dctx->curd4, errhp,
stmbuf, strlen((char *)stmbuf),
    OCI_NTV_SYNTAX,
OCI_DEFAULT);
    /* bind variables */
    OCIBND(dctx->curd4, dctx-
>w_id_bp4,errhp,"w_id",dctx->w_id,
    SIZ(int), SOLT_INT);
    OCIBND(dctx->curd4, dctx-
>d_id_bp4,errhp,"d_id",dctx->d_id,
    SIZ(int), SOLT_INT);
    OCIBND(dctx->curd4, dctx-
>o_id_bp,errhp,"o_id",dctx->del_o_id,
    SIZ(int),SOLT_INT);
    OCIBND(dctx->curd4, dctx-
>cr_date_bp,errhp,"cr_date", dctx->del_date,
    SIZ(OCIDate), SOLT_ODT);
    OCIBNDRAD(dctx->curd4, dctx->olamt_bp,
errhp, "ol_amount",
    SIZ(int), SOLT_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, SOLTXT6);
    DISCARD OCISmtPrepare(dctx->curd6, errhp,
stmbuf, strlen((char *)stmbuf),
    OCI_NTV_SYNTAX,
OCI_DEFAULT);
    /* bind variables */
    OCIBND(dctx->curd6,dctx-
>amt_bp,errhp,"amt",dctx->amt,SIZ(int),
    SOLT_INT);
    OCIBND(dctx->curd6,dctx-
>w_id_bp6,errhp,"w_id",dctx->w_id,SIZ(int),
    SOLT_INT);
    OCIBND(dctx->curd6,dctx-
>d_id_bp6,errhp,"d_id",dctx->d_id,SIZ(int),
    SOLT_INT);
    OCIBND(dctx->curd6,dctx-
>c_id_bp,errhp,"c_id",dctx->c_id,SIZ(int),
    SOLT_INT);
    }
    return (0);
}
void CDBAcess::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];
    }
}
CDBAcess::tkvcd (int plsqflg)
{
    /*int i, j;
    int i;
    /*int rpc,rcount,count;
    int rpc,rcount;
    int invalid;
    if (plsqflg)
    {
        pldctx->w_id_len = sizeof (int);
        pldctx->carrier_id_len = sizeof (int);
        for (i = 0; i < NDISTS; i++)
        {
            pldctx->del_o_id_len[i] = sizeof(int);
            del_o_id[i] = 0;
        }
        pldctx->del_date_len = DEL_DATE_LEN;
        DISCARD memcpy(&pldctx-
>del_date,&cr_date,sizeof(OCIDate));
        pldctx->retry=0;
        DISCARD OCIERROR(errhp,
    OCISmtExecute(tpcsvc,pldctx-
>curp2,errhp,1,0,NULLP(CONST OCISnapshot),
NULLP(OCISnapshot),OCI_DEFAULT));
        for (i = 0; i < NDISTS; i++)
        {
            del_o_id[i] = 0;
        }
        for (i = 0; (unsigned int)i < pldctx-
>del_o_id_rcnt; i++)
            del_o_id[pldctx->del_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=OCISmtExecute(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=OCISmtExecute(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)
{
  userlog (LOG_FILE_LINE, "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,rcp,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) {
  userlog
  (LOG_FILE_LINE, "Error in TPC-C
server %d: %d ordrs updated, %d ordl
updated\n",
  proc_no, rpc, rcount);
}
#endif
/* array update of customer table */
execstatus=OCIStmtExecute(tpcsvc,dctx-
>curd6,errhp,rcp,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) {
  userlog (LOG_FILE_LINE, "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 CDBAccess::tkvcddone (int plsqflag)
{
  if (plsqflag)
  {
    if (pldctx)
    {
      DISCARD OCIHandleFree((dvoid *)dctx-
>curd0,OCI_HTYPE_STMT);
    }
  }
}

```

```

DISCARD free(pldctx);
}
else
{
  if (dctx)
  {
    OCIHandleFree((dvoid *)dctx-
>curd1,OCI_HTYPE_STMT);
    OCIHandleFree((dvoid *)dctx-
>curd2,OCI_HTYPE_STMT);
    OCIHandleFree((dvoid *)dctx-
>curd3,OCI_HTYPE_STMT);
    OCIHandleFree((dvoid *)dctx-
>curd4,OCI_HTYPE_STMT);
    OCIHandleFree((dvoid *)dctx-
>curd5,OCI_HTYPE_STMT);
    OCIHandleFree((dvoid *)dctx-
>curd6,OCI_HTYPE_STMT);
    DISCARD free (dctx);
  }
}
.....
SvrDBDepend_Oracle/plnew.cpp
.....
#ifdef RCSID
static char *RCSid =
  "$Header: tkvnew.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 <windows.h>
#include "log.h"
#ifdef ORA_TPCC
# define ORA_TPCC
# include "tpcc.h"
#endif
#include "tpccsvr_Obj.h"
#define SQLTXT2 "BEGIN
inittpcc.init_no:(idx1arr); END;"
#define NITEMS 15
#define ROWIDLEN 20
#define OCIROWLEN 20
CDBAccess::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->curm1),
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 03.05.16 COM+ Client */
#if 0
! sqlfile("../blocks/tkvcpnew.sql",stmbuf);
#endif
sqlfile("../blocks/tkvcpnew.sql",stmbuf);
/* Replaced end */
#endif
#endif
/* Replaced end */
DISCARD
OCIERROR(errhp,OCIStmtPrepare(nctx->curm1,
errhp, stmbuf,
strlen((char *)stmbuf), OCI_NTV_SYNTAX,
OCI_DEFAULT));
/* bind variables */
OCIBNDPL(nctx->curm1, nctx->w_id_bp, errhp,
"w_id",ADR(w_id),SIZ(w_id),
SQLT_INT, &nctx->w_id_len);
OCIBNDPL(nctx->curm1, nctx->d_id_bp, errhp,
"d_id",ADR(d_id),SIZ(d_id),
SQLT_INT, &nctx->d_id_len);
OCIBNDPL(nctx->curm1, nctx->c_id_bp, errhp,
"c_id",ADR(c_id),SIZ(c_id),
SQLT_INT, &nctx->c_id_len);
OCIBNDPL(nctx->curm1, 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->curm1, 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->curm1, nctx->w_tax_bp, errhp,
"w_tax",ADR(w_tax),SIZ(w_tax),

```

```

SQLT_FLT, &nctx->w_tax_len);
OCIBNDPL(nctx->curm1, nctx->d_tax_bp, errhp,
"d_tax",ADR(d_tax),SIZ(d_tax),
SQLT_FLT, &nctx->d_tax_len);
OCIBNDPL(nctx->curm1, nctx->o_id_bp, errhp,
"o_id",ADR(o_id),SIZ(o_id),
SQLT_INT, &nctx->o_id_len);
OCIBNDPL(nctx->curm1, nctx->c_discount_bp,
errhp, "c_discount",
ADR(c_discount),
SIZ(c_discount),SQLT_FLT, &nctx-
>c_discount_len);
OCIBNDPL(nctx->curm1, nctx->c_credit_bp,
errhp, "c_credit",c_credit,
SIZ(c_credit),SQLT_CHR, &nctx-
>c_credit_len);
OCIBNDPL(nctx->curm1, nctx->c_last_bp, errhp,
"c_last",c_last,SIZ(c_last),
SQLT_STR, &nctx->c_last_len);
OCIBNDPL(nctx->curm1, nctx->retries_bp, errhp,
"retry",ADR(retries),
SIZ(retries),SQLT_INT, &nctx->retries_len);
OCIBNDPL(nctx->curm1, nctx->cr_date_bp,
errhp, "cr_date",&cr_date,
SIZ(OCIDate), SQLT_ODT, &nctx-
>cr_date_len);
OCIBNDPLA(nctx->curm1, 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->curm1, 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->curm1, nctx-
>ol_quantity_bp,errhp,"ol_quantity",
! nol_quantity, SIZ(int),SQLT_INT,nctx-
>nol_quantity_len,
! NITEMS,&nctx->nol_q_count);
! OCIBNDPLA(nctx->curm1, 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->curm1, nctx-
>ol_quantity_bp,errhp,"ol_quantity",
nol_quantity,
SIZ(float),SQLT_BFLOAT,nctx->nol_quantity_len,
NITEMS,&nctx->nol_q_count);
OCIBNDPLA(nctx->curm1, 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->curm1, nctx-
>ol_quantity_bp,errhp,"ol_quantity",
nol_quantity, SIZ(int),SQLT_INT,nctx-
>nol_quantity_len,
NITEMS,&nctx->nol_q_count);
OCIBNDPLA(nctx->curm1, 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->curm1, 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->curm1, nctx-
>s_quantity_bp,errhp,"s_quantity",s_quantity,
! SIZ(int), SQLT_INT,nctx-
>s_quantity_len,NITEMS,&nctx->nol_qty_count);
#endif

#ifdef USE_IEEE_NUMBER
OCIBNDPLA(nctx->curm1, nctx-
>s_quantity_bp,errhp,"s_quantity",s_quantity,
SIZ(float), SQLT_BFLOAT,nctx-
>s_quant_len,NITEMS,&nctx->nol_qty_count);
#else
OCIBNDPLA(nctx->curm1, nctx-
>s_quantity_bp,errhp,"s_quantity",s_quantity,
SIZ(int),SQLT_INT,nctx-
>s_quant_len,NITEMS,&nctx->nol_qty_count);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */
OCIBNDPLA(nctx->curm1, nctx-
>s_bg_bp,errhp,"brand_generic",brand_generic,
SIZ(char), SQLT_CHR,nctx-
>s_bg_len,NITEMS,&nctx->nol_bg_count);
/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
! OCIBNDPLA(nctx->curm1, nctx-
>ol_amount_bp,errhp,"ol_amount",nol_amount,
! SIZ(int),SQLT_INT, nctx-
>nol_amount_len,NITEMS,&nctx-
>nol_am_count);
! OCIBNDPLA(nctx->curm1, nctx-
>s_remote_bp,errhp,"s_remote",nctx->s_remote,
! SIZ(int),SQLT_INT, nctx-
>s_remote_len,NITEMS,&nctx->s_remote_count);
#endif
#ifdef USE_IEEE_NUMBER
OCIBNDPLA(nctx->curm1, nctx-
>ol_amount_bp,errhp,"ol_amount",nol_amount,
SIZ(float),SQLT_BFLOAT, nctx-
>nol_amount_len,NITEMS,&nctx-
>nol_am_count);
OCIBNDPLA(nctx->curm1, nctx-
>s_remote_bp,errhp,"s_remote",nctx->s_remote,
SIZ(float),SQLT_BFLOAT, nctx-
>s_remote_len,NITEMS,&nctx->s_remote_count);
#else
OCIBNDPLA(nctx->curm1, nctx-
>ol_amount_bp,errhp,"ol_amount",nol_amount,
SIZ(int),SQLT_INT, nctx-
>nol_amount_len,NITEMS,&nctx-
>nol_am_count);
OCIBNDPLA(nctx->curm1, nctx-
>s_remote_bp,errhp,"s_remote",nctx->s_remote,
SIZ(int),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->curm2),

```

```

OCI_HTYPE_STMT, 0, (dvoid**)0);
DISCARD sprintf ((char *) stmbuf, SQLTXT2);
DISCARD
OCIERROR(errhp,OCIStmtPrepare(nctx->cur2,
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->cur2,
idx1arr_bp,errhp,"idx1arr",idx1arr,
        SI2(int), SQLT_INT, idx1arr_len,
NITEMS,&idx1arr_count);
    DBGLOG("NEW:[1]Start",0);
    execstatus = OCIStmtExecute(tpcsvc,nctx-
>cur2,errhp,1,0,
        NULLP(CONST
OCI_Snapshot),NULLP(OCI_Snapshot),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);
}
CDBAcess::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 (noI_i_id[i] == 0) {
                o_ol_cnt = i;
                break;
            }
        }
        if (noI_supply_w_id[i] != w_id) {
/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#endif
! 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->noI_i_count = o_ol_cnt;
nctx->noI_q_count = o_ol_cnt;
nctx->noI_s_count = o_ol_cnt;
nctx->s_remote_count = o_ol_cnt;
nctx->noI_qty_count = 0;
nctx->noI_bg_count = 0;
nctx->noI_item_count = 0;
nctx->noI_name_count = 0;
nctx->noI_am_count = 0;
/* initialization for array operations */
for (i = 0; i < o_ol_cnt; i++) {
    nctx->oI_number[i] = i + 1;
    nctx->noI_i_id_len[i] = sizeof(int);
    nctx->noI_supply_w_id_len[i] = sizeof(int);
    nctx->noI_quantity_len[i] = sizeof(int);
    nctx->noI_amount_len[i] = sizeof(int);
    nctx->oI_o_id_len[i] = sizeof(int);
    nctx->oI_number_len[i] = sizeof(int);
    nctx->oI_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->noI_i_id_len[i] = 0;
    nctx->noI_supply_w_id_len[i] = 0;
    nctx->noI_quantity_len[i] = 0;
    nctx->noI_amount_len[i] = 0;
    nctx->oI_o_id_len[i] = 0;
    nctx->oI_number_len[i] = 0;
    nctx->oI_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-
>cur1,errhp,1,0,0,0,

```

```

OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
DBGLOG("NEW:[2]End >%d",execstatus);
if(execstatus != OCI_SUCCESS) {
OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
    errcode = OCIERROR(errhp,execstatus);
    if(errcode == NOT_SERIALIZABLE) {
        retries++;
        goto retry;
    } else if (errcode == RECOVER) {
        retries++;
        goto retry;
    }
/* Deleted T.Kato 02.10.25 */
}
if 0
! ) else if (errcode == SNAPSHOT_TOO_OLD)
{
! retries++;
! goto retry;
#endif
/* Deleted end */
} else {
    return -1;
}
}
/* did the txn succeed ? */
if (rcount != o_ol_cnt)
{
    status = rcount - o_ol_cnt;
    o_ol_cnt = rcount;
}
#ifdef DEBUG
    printf("w_id = %d, d_id = %d, c_id = %d\n",w_id,
d_id, c_id);
#endif
    return (0);
}
void CDBAcess::tkvcndone ()
{
/* for warning */
/* int i;*/
if (nctx)
{
    DISCARD OCIHandleFree((dvoid *)nctx-
>cur1,OCI_HTYPE_STMT);
    DISCARD OCIHandleFree((dvoid *)nctx-
>cur2,OCI_HTYPE_STMT);
    free (nctx);
}
}
.....
SvrDBDepend_Oracle/plord.cpp
.....
/* 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_tpccitr
lwang 07/23/01 - fix include
lwang 07/23/01 - Creation
*/
#include <Windows.h>
#include "log.h"
#include "tpcc.h"
#include "tpccsvr_Obj.h"
/*-----
PRIVATE TYPES AND
CONSTANTS
-----*/

/*-----
STATIC FUNCTION
DECLARATIONS
-----*/
#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(ordr) CLUSTER(ordr) */ \
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"
CDBAccess::tkvcoint ()
{
int i;
text stmbuff[SQL_BUF_SIZE];
octx = (ordctx *) malloc (sizeof(ordctx));
DISCARD memset(octx, (char)0, sizeof(ordctx));
octx->cs = 1;
octx->norow = 0;
octx->somerows = 10;
/* get the rowid handles */
OCIERROR(errhp, OCIDescriptorAlloc((dvoid *)tpcenv, (dvoid **) &octx->o_rowid, (ub4)OCI_DTYPE_ROWID, (size_t)0, (dvoid **)0);
for(i=0; i<100; i++) {
DISCARD OCIERROR(errhp, OCIDescriptorAlloc(tpcenv, (dvoid **) &octx->c_rowid_ptr[i], OCI_DTYPE_ROWID, 0, (dvoid **)0);
}
DISCARD OCIERROR(errhp, OCIHandleAlloc(tpcenv, (dvoid **) &octx->uro0, OCI_HTYPE_STMT, 0, (dvoid **)0);
DISCARD OCIERROR(errhp, OCIHandleAlloc(tpcenv, (dvoid **) &octx->uro0, OCI_HTYPE_STMT, 0, (dvoid **)0);
DISCARD OCIERROR(errhp, OCIHandleAlloc(tpcenv, (dvoid **) &octx->uro1, OCI_HTYPE_STMT, 0, (dvoid **)0);
DISCARD OCIERROR(errhp, OCIHandleAlloc(tpcenv, (dvoid **) &octx->uro2, OCI_HTYPE_STMT, 0, (dvoid **)0);
DISCARD OCIERROR(errhp, OCIHandleAlloc(tpcenv, (dvoid **) &octx->uro3, OCI_HTYPE_STMT, 0, (dvoid **)0);
DISCARD OCIERROR(errhp, OCIHandleAlloc(tpcenv, (dvoid **) &octx->uro4, OCI_HTYPE_STMT, 0, (dvoid **)0);

/* c_id = 0, use find customer by lastname. Get an array of rowid's back */
DISCARD sprintf((char *) stmbuff, SQLCUR0);
DISCARD OCIERROR(errhp, OCISmtPrepare(octx->uro0, errhp, stmbuff, (ub4)strlen((char *)stmbuff));

OCI_NTV_SYNTAX, OCI_DEFAULT));
DISCARD OCIERROR(errhp, OCIAttrSet(octx->uro0, OCI_HTYPE_STMT, &octx->norow, 0, OCI_ATTR_PREFETCH_ROWS, errhp));
/* get order/customer info back based on rowid */
DISCARD sprintf((char *) stmbuff, SQLCUR1);
DISCARD OCIERROR(errhp, OCISmtPrepare(octx->uro1, errhp, stmbuff, (ub4)strlen((char *)stmbuff));

OCI_NTV_SYNTAX, OCI_DEFAULT));
DISCARD OCIERROR(errhp, OCIAttrSet(octx->uro1, OCI_HTYPE_STMT, &octx->norow, 0, OCI_ATTR_PREFETCH_ROWS, errhp));
/* c_id == 0, use lastname to find customer */

```

```

DISCARD sprintf((char *) stmbuff, SQLCUR2);
DISCARD OCIERROR(errhp, OCISmtPrepare(octx->uro2, errhp, stmbuff, (ub4)strlen((char *)stmbuff));

OCI_NTV_SYNTAX, OCI_DEFAULT));
DISCARD OCIERROR(errhp, OCIAttrSet(octx->uro2, OCI_HTYPE_STMT, &octx->norow, 0, OCI_ATTR_PREFETCH_ROWS, errhp));
DISCARD sprintf((char *) stmbuff, SQLCUR3);
DISCARD OCIERROR(errhp, OCISmtPrepare(octx->uro3, errhp, stmbuff, (ub4)strlen((char *)stmbuff));

OCI_NTV_SYNTAX, OCI_DEFAULT));
DISCARD OCIERROR(errhp, OCIAttrSet(octx->uro3, OCI_HTYPE_STMT, &octx->norow, 0, OCI_ATTR_PREFETCH_ROWS, errhp));
DISCARD sprintf((char *) stmbuff, SQLCUR4);
DISCARD OCIERROR(errhp, OCISmtPrepare(octx->uro4, errhp, stmbuff, (ub4)strlen((char *)stmbuff));

OCI_NTV_SYNTAX, OCI_DEFAULT));
DISCARD OCIERROR(errhp, OCIAttrSet(octx->uro4, 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->uro0, octx->w_id_bp[0], errhp, ":w_id", ADR(w_id), SIZ(int), SOLT_INT);
OCIBND(octx->uro0, octx->d_id_bp[0], errhp, ":d_id", ADR(d_id), SIZ(int), SOLT_INT);
OCIBND(octx->uro0, octx->c_last_bp[0], errhp, ":c_last", c_last, SIZ(c_last), SOLT_STR);
OCIDFNRA(octx->uro0, octx->c_rowid_dp, errhp, 1, octx->c_rowid_ptr, SIZ(OCIRowid*), SOLT_RDD, NULL, octx->c_rowid_len, NULL);

```

```

OCIBND(octx->куро1,octx-
>c_rowid_bp,errhp,"cust_rowid", &octx-
>c_rowid_cust,
    sizeof(octx->c_rowid_ptr[0]),SQLT_RDD);
OCIDEF(octx->куро1,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->куро1,octx-
>c_balance_dp[0],errhp,2,ADR(c_balance),
! SIZ(double),SQLT_FLT);
#endif

#ifdef USE_IEEE_NUMBER
OCIDEF(octx->куро1,octx-
>c_balance_dp[0],errhp,2,ADR(c_balance),
    SIZ(double),SQLT_BDOUBLE);
#else
OCIDEF(octx->куро1,octx-
>c_balance_dp[0],errhp,2,ADR(c_balance),
    SIZ(double),SQLT_FLT);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

OCIDEF(octx->куро1,octx-
>c_first_dp[0],errhp,3,c_first,SIZ(c_first)-1,
    SOLT_CHR);
OCIDEF(octx->куро1,octx-
>c_middle_dp[0],errhp,4,c_middle,
    SIZ(c_middle)-1,SQLT_AFC);
OCIDEF(octx->куро1,octx-
>c_last_dp[0],errhp,5,c_last,SIZ(c_last)-1,
    SOLT_CHR);
OCIDEF(octx->куро1,octx-
>o_id_dp[0],errhp,6,ADR(o_id),SIZ(int),SQLT_INT
);
OCIDEF(octx->куро1,octx-
>o_entry_d_dp[0],errhp,7,
&o_entry_d_base,SIZ(OCIDate),SQLT_ODT);
OCIDEF(octx->куро1,octx-
>o_cr_id_dp[0],errhp,8,ADR(o_carrier_id),
    SIZ(int),SQLT_INT);
OCIDEF(octx->куро1,octx-
>o_ol_cnt_dp[0],errhp,9,ADR(o_ol_cnt),
    SIZ(int),SQLT_INT);
OCIDEF(octx->куро1,octx-
>o_rowid_dp[0],errhp,10,ADR(octx->o_rowid),
    SIZ(OCIRowid*),SQLT_RDD);

/* Bind for third cursor , no-zero customer id */
OCIBND(octx->куро2,octx-
>w_id_bp[1],errhp,"w_id",ADR(w_id),
    SIZ(int),SQLT_INT);
OCIBND(octx->куро2,octx-
>d_id_bp[1],errhp,"d_id",ADR(d_id),
    SIZ(int),SQLT_INT);
OCIBND(octx->куро2,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->куро2,octx-
>c_balance_dp[1],errhp,1,ADR(c_balance),
! SIZ(double),SQLT_FLT);

```

```

#endif

#ifdef USE_IEEE_NUMBER
OCIDEF(octx->куро2,octx-
>c_balance_dp[1],errhp,1,ADR(c_balance),
    SIZ(double),SQLT_BDOUBLE);
#else
OCIDEF(octx->куро2,octx-
>c_balance_dp[1],errhp,1,ADR(c_balance),
    SIZ(double),SQLT_FLT);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

OCIDEF(octx->куро2,octx-
>c_first_dp[1],errhp,2,c_first,SIZ(c_first)-1,
    SOLT_CHR);
OCIDEF(octx->куро2,octx-
>c_middle_dp[1],errhp,3,c_middle,
    SIZ(c_middle)-1,SQLT_AFC);
OCIDEF(octx->куро2,octx-
>c_last_dp[1],errhp,4,c_last,SIZ(c_last)-1,
    SOLT_CHR);
OCIDEF(octx->куро2,octx-
>o_id_dp[1],errhp,5,ADR(o_id),SIZ(int),SQLT_INT
);
OCIDEF(octx->куро2,octx-
>o_entry_d_dp[1],errhp,6, &o_entry_d_base,
    SIZ(OCIDate),SQLT_ODT);
OCIDEF(octx->куро2, octx-
>o_cr_id_dp[1],errhp,7,ADR(o_carrier_id),
    SIZ(int), SQLT_INT);
OCIDEF(octx->куро2,octx-
>o_ol_cnt_dp[1],errhp,8,ADR(o_ol_cnt),
    SIZ(int),SQLT_INT);
OCIDEF(octx->куро2,octx-
>o_rowid_dp[1],errhp,9,ADR(octx->o_rowid),
    SIZ(OCIRowid*),SQLT_RDD);

/* Bind for last cursor */
/*
OCIBND(octx->куро3,octx-
>w_id_bp[2],errhp,"w_id",ADR(w_id),
    SIZ(int),SQLT_INT);
OCIBND(octx->куро3,octx-
>d_id_bp[2],errhp,"d_id",ADR(d_id),
    SIZ(int),SQLT_INT);
OCIBND(octx->куро3,octx-
>o_id_bp,errhp,"o_id",ADR(o_id),
    SIZ(int),SQLT_INT);
OCIBND(octx->куро3,octx-
>c_id_bp,errhp,"c_id",ADR(c_id),
    SIZ(int),SQLT_INT);
*/
OCIBND(octx->куро3,octx-
>o_rowid_bp,errhp,"odr_rowid",
    &octx->o_rowid,
    SIZ(OCIRowid*),SQLT_RDD);
OCIDFNRA(octx->куро3, octx->ol_i_id_dp,
errhp, 1, ol_i_id,SIZ(int),SQLT_INT,
    NULL,octx->ol_i_id_len, NULL);
OCIDFNRA(octx->куро3,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->куро3, octx-
>ol_quantity_dp,errhp,3, ol_quantity,SIZ(int),

```

```

! SQLT_INT, NULL,octx->ol_quantity_len,
NULL);
! OCIDFNRA(octx->куро3,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->куро3, octx-
>ol_quantity_dp,errhp,3, ol_quantity,SIZ(float),
    SQLT_BFLOAT, NULL,octx-
>ol_quantity_len, NULL);
OCIDFNRA(octx->куро3,octx-
>ol_amount_dp,errhp,4,ol_amount, SIZ(float),
    SQLT_BFLOAT,NULL, octx-
>ol_amount_len, NULL);
#else
OCIDFNRA(octx->куро3, octx-
>ol_quantity_dp,errhp,3, ol_quantity,SIZ(int),
    SQLT_INT, NULL,octx->ol_quantity_len,
NULL);
OCIDFNRA(octx->куро3,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->куро3,octx-
>ol_d_base_dp,errhp,5,ol_d_base,SIZ(OCIDate),
    SQLT_ODT, NULL,octx-
>ol_delivery_d_len,NULL);
OCIBND(octx->куро4,octx-
>w_id_bp[3],errhp,"w_id",ADR(w_id),
    SIZ(int),SQLT_INT);
OCIBND(octx->куро4,octx-
>d_id_bp[3],errhp,"d_id",ADR(d_id),
    SIZ(int),SQLT_INT);
OCIBND(octx->куро4,octx-
>c_last_bp[1],errhp,"c_last",c_last,
    SIZ(c_last),SQLT_STR);
OCIDEF(octx->куро4,octx-
>c_count_dp,errhp,1,ADR(octx->rcount),SIZ(int),
    SQLT_INT);
return (0);
}
CDBAccess::tkvco ()
{
    int i;
    int rcount;
    #if defined(SO9)
    int secondread = 0;
    char sdate[30];
    ub4 datelen;
    sysdate(sdate);
    printf("Order Status started at: %s\n", sdate);
    #endif
    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));

```

```

for (i = 0; i < NITEMS; i++) {
    octx->ol_supply_w_id_len[i] = sizeof(int);
    octx->ol_i_id_len[i] = sizeof(int);
    octx->ol_quantity_len[i] = sizeof(int);
    octx->ol_amount_len[i] = sizeof(int);
    octx->ol_delivery_d_len[i] = sizeof(OCIDate);
}
octx->ol_supply_w_id_csize = NITEMS;
octx->ol_i_id_csize = NITEMS;
octx->ol_quantity_csize = NITEMS;
octx->ol_amount_csize = NITEMS;
octx->ol_delivery_d_csize = NITEMS;
retry:
if (bylastname)
{
    cbctx.reexec = FALSE;
    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 == RECOVER))
        {
            DISCARD
OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
            retries++;
            goto retry;
        } else {
            return -1;
        }
    }
    if (execstatus == OCI_NO_DATA) /* there are
no more rows */
    {
        /* get rowcount, find middle one */
        DISCARD OCIAttrGet(octx-
>curo0,OCI_HTYPE_STMT,&rcount,NULL,
OCI_ATTR_ROW_COUNT,errhp);
        if (rcount <1)
        {
            userlog(LOG_FILE_LINE,
"ORDERSTATUS rcount=%d\n",rcount);
            return (-1);
        }
        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 == RECOVER))
        {
            DISCARD
OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
            retries++;
            goto retry;
        } else {
            return -1;
        }
    }
}
if (octx->rcount+1 < 2*10)
    octx->cust_idx=(octx->rcount+1)/2 ;
else
    /* */
    {
        cbctx.reexec = TRUE;
        cbctx.count = (octx->rcount+1)/2 ;
        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)
        {
            userlog (LOG_FILE_LINE, "did not get all
rows ");
            return (-1);
        }
        if ((execstatus != OCI_NO_DATA) &&
(execstatus != OCI_SUCCESS))
        {
            errcode=OCIERROR(errhp, execstatus);
            if((errcode == NOT_SERIALIZABLE) ||
(errcode == RECOVER))
            {
                DISCARD
OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
                retries++;
                goto retry;
            } else {
                return -1;
            }
        }
        octx->cust_idx=0 ;
    }
    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 == RECOVER))
        {
            || (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 == RECOVER))
        {
            || (errcode == SNAPSHOT_TOO_OLD))
        {
            retries++;
            goto retry;
        }
        else
        {
            return -1;
        }
    }
}
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);
#ifdef ISO9
    sysdate (sdate);
if (lsecondread)
    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",
sdate);
if (lsecondread) {
    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 = OCISmtExecute(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);
OCIRowidToChar(octx->o_rowid, a_row_id,
&buf_len, errhp);
userlog(LOG_FILE_LINE, "DBG_LOG start :
w_id=%d d_id=%d c_id=%d\n", f_w_id, f_d_id,
f_c_id);
userlog(LOG_FILE_LINE, "DBG_LOG cur2 :
w_id=%d d_id=%d c_id=%d\n", c2_w_id, c2_d_id,
c2_c_id);
userlog(LOG_FILE_LINE, "DBG_LOG cur2 :
row_id=[%s]\n", b_row_id);
userlog(LOG_FILE_LINE, "DBG_LOG error :
row_id=[%s]\n", a_row_id);
DISCARD
OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
if((errcode == NOT_SERIALIZABLE) ||
(errcode == RECOVER))
|| (errcode == SNAPSHOT_TOO_OLD))
{
retries++;
goto retry;
}
else
{
return -1;
}
}
/* clean up and convert the delivery dates */
for (i = 0; i < o_ol_cnt; i++)
{
ol_de_len[i]=sizeof(ol_delivery_d[i]);
DISCARD
OCIERROR(errhp,OCIDateToText(errhp,&ol_d_b
ase[i],
(const
text*)SHORTDATE,(ub1)strlen(SHORTDATE),(te
xt*)0,0,
&ol_de_len[i], ol_delivery_d[i]));
/*
cvtdmy(ol_d_base[i],ol_delivery_d[i]);
*/
}
return (0);
}
void CDBAccess::tkvcodone ()
{
if (octx)
free (octx);
}
/* end of file tkvcord.c */

```

```

.....
SvrDBDepend_Oracle/plpay.cpp
.....
#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 <Windows.h>
#include "log.h"
#include "tpcc.h"
#include "tpccsvr_Obj.h"
#define SQLTXT_INIT "BEGIN inittpc.init_pay;
END;"
int CDBAccess::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,OCISmtPrepare(pctx->curpi,
errhp, stmbuf,
strlen((char *)stmbuf), OCI_NTV_SYNTAX,
OCI_DEFAULT));
DBGLOG("PAY:[1]Start",0);
DISCARD OCIERROR(errhp,
OCISmtExecute(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 03.05.16 COM+ Client */
#endif

```

```

! sqlfile("../blocks/paynz.sql",stmbuf);
#endif
sqlfile("../blocks/paynz.sql",stmbuf);
/* Replaced end */
DISCARD
OCIERROR(errhp,OCISmtPrepare(pctx->curp0,
errhp, stmbuf,
strlen((char *)stmbuf), OCI_NTV_SYNTAX,
OCI_DEFAULT));
/* customer id == 0, go by last name */
/* Replaced 03.05.16 COM+ Client */
#if 0
! sqlfile("../blocks/payz.sql",stmbuf); /* sqlfile
opens $O/bench/.../blocks/... */
#endif
sqlfile("../blocks/payz.sql",stmbuf); /* sqlfile opens
$O/bench/.../blocks/... */
/* Replaced end */
DISCARD
OCIERROR(errhp,OCISmtPrepare(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 */
#ifndef
! 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);
}
CDBAccess::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->h_amount_len = SIZ(h_amount);
    pctx->c_last_len = SIZ(c_last);
    pctx->w_street_1_len = 0;
    pctx->w_street_2_len = 0;
    pctx->w_city_len = 0;
    pctx->w_state_len = 0;
    pctx->w_zip_len = 0;
    pctx->d_street_1_len = 0;
    pctx->d_street_2_len = 0;
    pctx->d_city_len = 0;
    pctx->d_state_len = 0;
    pctx->d_zip_len = 0;
    pctx->c_first_len = 0;
    pctx->c_middle_len = 0;
    pctx->c_street_1_len = 0;
    pctx->c_street_2_len = 0;
    pctx->c_city_len = 0;
    pctx->c_state_len = 0;
    pctx->c_zip_len = 0;
    pctx->c_phone_len = 0;
    pctx->c_since_len = 0;

```

```

    pctx->c_credit_len = 0;
    pctx->c_credit_lim_len = 0;
    pctx->c_discount_len = 0;
    pctx->c_balance_len = sizeof(double);
    pctx->c_data_len = 0;
    pctx->h_date_len = 0;
    pctx->retries_len = SIZ(retries);
    pctx->cr_date_len = 7;
    if(bylastname) {
        DBGLOG("PAY:[2]Start",0);
        execstatus=OCIStmtExecute(tpcsvc,pctx-
>curp1,errhp,1,0,
        NULL(CONST
OCISnapshot),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,
        NULL(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 == RECOVER) {
            retries++;
            goto retry;
        } else if (errcode == SNAPSHOT_TOO_OLD) {
            retries++;
            goto retry;
        } else {
            return -1;
        }
    }
    return 0;
}
void CDBAccess::tkvcpdone ()
{
    if(pctx) {
        free(pctx);
    }
}
SvrDBDepend_Oracle/plsto.cpp
=====
#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 <Windows.h>
#include "log.h"
#include "tpcc.h"
#include "tpccsvr_Obj.h"
#ifdef PLSQLSTO
#define SQLTXT "BEGIN stocklevel.getstocklevel
(:w_id, :d_id, :threshold, \
 :low_stock); END;"
#else
/* 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) \
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
CDBAccess::tkvcsinit ()
{
    text stmbuff[SQL_BUF_SIZE];
    stcx = (stocx *)malloc(sizeof(stocx));
    memset(stcx, (char)0, sizeof(stocx));
    stcx->norow=0;
    OCIERROR(errhp,
        OCIHandleAlloc(tpcenv, (dvoid**)&stcx-
>curs, OCI_HTYPE_STMT, 0, (dvoid**)0));
    sprintf ((char *) stmbuff, SQLTXT);
    OCIERROR(errhp, OCIStmtPrepare(stcx-
>curs, errhp, stmbuff, strlen((char *)stmbuff),
OCI_NTV_SYNTAX, OCI_DEFAULT));
#ifdef PLSQLSTO
    OCIERROR(errhp,
        OCIAttrSet(stcx-
>curs, OCI_HTYPE_STMT, (dvoid*)&stcx->norow, 0,
OCI_ATTR_PREFETCH_ROWS, errhp));
#endif
    /* bind variables */
    OCIBND(stcx->curs, stcx->w_id_bp, errhp,
":w_id", ADR(w_id), sizeof(int),
    SOLT_INT);
    OCIBND(stcx->curs, stcx->d_id_bp, errhp, ":d_id",
ADR(d_id), sizeof(int),
    SOLT_INT);

/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#endif

```

```

! OCIBND(stcx->curs, stcx->threshold_bp, errhp,
":threshold", ADR(threshold),
!   sizeof(int), SOLT_INT);
#endif

#ifdef USE_IEEE_NUMBER
    OCIBND(stcx->curs, stcx->threshold_bp, errhp,
":threshold", ADR(threshold),
    sizeof(float), SOLT_BFLOAT);
#else
    OCIBND(stcx->curs, stcx->threshold_bp, errhp,
":threshold", ADR(threshold),
    sizeof(int), SOLT_INT);
#endif /* USE_IEEE_NUMBER */
/* Replaced end */

#ifdef PLSQLSTO
    OCIBND(stcx->curs, stcx-
>low_stock_bp, errhp, ":low_stock",
ADR(low_stock),
    sizeof(int), SOLT_INT);
#else
    OCIDEFINE(stcx->curs, stcx-
>low_stock_bp, errhp, 1, ADR(low_stock),
    sizeof(int), SOLT_INT);
#endif
return (0);
}
CDBAccess::tkvcs ()
{
    retry:
    DBGLOG("STO:[1]Start", 0);
    execstatus= OCISmtExecute(tpcsvc, stcx-
>curs, errhp, 1, 0, 0, 0,
OCI_COMMIT_ON_SUCCESS | OCI_DEFAULT);
    DBGLOG("STO:[1]End >%d", execstatus);
    if (execstatus != OCI_SUCCESS)
    {
        errcode=OCIERROR(errhp, execstatus);
    }
    OCITransCommit(tpcsvc, errhp, OCI_DEFAULT);
    if((errcode == NOT_SERIALIZABLE) ||
(errcode == RECOVER))
    {
        || (errcode == SNAPSHOT_TOO_OLD))
        {
            retries++;
            goto retry;
        } else {
            return -1;
        }
    }
    return (0);
}
void CDBAccess::tkvcsdone ()
{
    if(stcx) free(stcx);
}
+=====
+=====+
SvrDBDepend_Oracle/svrapl.def
+=====
+=====+
EXPORTS
ExSvrAplInitiate
ExSrvAplTerminate
ExTrxNewOrder
ExTrxPayment
ExTrxOrderStatus
ExTrxStockLevel
ExTrxDelivery

```

```

ExTrxStructSize
ExSvrAplCreate
ExSvrAplDelete
+=====
SvrDBDepend_Oracle/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.
+=====
+=====+
#ifdef TPCC_H
#define TPCC_H
#ifdef FALSE
#define FALSE 0
#endif
#ifdef TRUE
#define TRUE 1
#endif
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#ifdef boolean
#define boolean int
#endif
#include <oratypes.h>
#include <oci.h>
#include <ocidfn.h>
/*
#ifdef __STDC__
#include "ociapr.h"
#else
#include "ocikpr.h"
#endif
*/
/* 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 RECOVER -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
/* Deleted 03.05.19 No use. */
#if 0
!extern int tkvcss (); /* for alter session to get
memory size and trace */
!extern boolean multitrans;
#endif
/* Deleted end */
/* Deleted 03.05.16 For warning */
#if 0
!extern int ord_init;
#endif
/* Deleted end */
/* Deleted 03.05.19 No use. */
#if 0
!extern void errpt ();
#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)
    DbgLog(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 OCIEERROR(errp,function)\
    ocierror(LOG_FILE_LINE, (errp), (function));
#define OCIBND(stmp, bndp, errp, sqlvar, progvl,
ftype)\
    ocierror(LOG_FILE_LINE, (errp), \
        OCIBindByName((stmp), &(bndp), (errp), \
            (text *) (sqlvar), strlen((sqlvar)), \
                (progvl), (progvl), \
                    (ftype), 0, 0, 0, 0, OCI_DEFAULT));
/* bind arrays for sql */

```

```

#define
OCIBNDRA(stmp, bndp, errp, sqlvar, progvl, ftype,
indp, alen, arcode) \
    DISCARD ocierror(LOG_FILE_LINE, (errp), \
        OCIBindByName((stmp), &(bndp), (errp), (text
*) (sqlvar), strlen((sqlvar)), \
            (progvl), (progvl), (ftype), (indp), (alen), (arcod
e), 0, 0, 0, OCI_DEFAULT));
/* use with callback data */
#define
OCIBNDRAD(stmp, bndp, errp, sqlvar, progvl, ftype, i
ndp, ctxp, \
    cbf_nodata, cbf_data) \
    DISCARD ocierror(LOG_FILE_LINE, (errp), \
        OCIBindByName((stmp), &(bndp), (errp), (text
*) (sqlvar), \
            strlen((sqlvar)), 0, (progvl), (ftype), \
                indp, 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, progvl, ftype, alen) \
    DISCARD ocierror(LOG_FILE_LINE, (errp), \
        OCIBindByName((stmp), &(bndp), (errp), (CONST
text *) (sqlvar), \
            (sb4) strlen((CONST char *) (sqlvar)), \
                (dvoid *) (progvl), (progvl), (ftype), \
                    NULLP(dvoid), (alen), NULLP(ub2), \
                        0, NULLP(ub4), OCI_DEFAULT));
/* bind in values for plsql with indicator and rcode
*/
#define
OCIBNDR(stmp, bndp, errp, sqlvar, progvl, progvl, ftype,
indp, alen, arcode) \
    DISCARD ocierror(LOG_FILE_LINE, (errp), \
        OCIBindByName((stmp), &(bndp), (errp), (CONST
text *) (sqlvar), \
            (sb4) strlen((CONST char *) (sqlvar)), \
                (dvoid *) (progvl), (progvl), (ftype), \
                    NULLP(dvoid), (alen), NULLP(ub2), \
                        0, NULLP(ub4), OCI_DEFAULT));
/* bind in/out for plsql arrays without indicator and
rcode */

```

```

#define
OCIBNDPLA(stmp, bndp, errp, sqlvar, progvl, progvl, f
type, alen, ms, cu) \
    DISCARD ocierror(LOG_FILE_LINE, (errp), \
        OCIBindByName((stmp), &(bndp), (errp), (CONST
text *) (sqlvar), \
            (sb4) strlen((CONST char *)
(sqlvar)), (void *) (progvl), \
                (progvl), (ftype), NULL, (alen), NULL, (ms), (cu), OCI_
DEFAULT));
/* bind in/out values for plsql with indicator and
rcode */
#define
OCIBNDRAA(stmp, bndp, errp, sqlvar, progvl, progvl, f
type, indp, alen, arcode, \
    ms, cu) \
    ocierror(LOG_FILE_LINE, (errp), \
        OCIBindByName((stmp), &(bndp), (errp), (text
*) (sqlvar), strlen((sqlvar)), \
            (progvl), (progvl), (ftype), (indp), (alen), (arcod
e), (ms), \
                (cu), OCI_DEFAULT));
#define
OCIDEFINE(stmp, dfnp, errp, pos, progvl, progvl, ftype) \
    OCIDEFINE(stmp, &(dfnp), (errp), (pos), (prog
v), (progvl), (ftype), \
        0, 0, 0, OCI_DEFAULT);
#define
OCIDEF(stmp, dfnp, errp, pos, progvl, progvl, ftype) \
    OCIBindByName((stmp), &(dfnp), (errp), (pos), (prog
v), (progvl), \
        (ftype), NULL, NULL, NULL, OCI_DEFAULT); \
#define
OCIDFNRA(stmp, dfnp, errp, pos, progvl, progvl, ftype,
indp, alen, arcode) \
    OCIBindByName((stmp), &(dfnp), (errp), (pos), (prog
v), (progvl), \
        (ftype), NULL, NULL, NULL, OCI_DEFAULT); \
#define
OCIDFNDR(stmp, dfnp, errp, pos, progvl, progvl, ftype,
indp, alen, arcode) \
    OCIBindByName((stmp), &(dfnp), (errp), (pos), (prog
v), (progvl), \
        (ftype), NULL, NULL, NULL, OCI_DEFAULT); \
#define
OCIDFNDRD(stmp, dfnp, errp, pos, progvl, progvl, ftype,
indp, ctxp, cbf_data) \
    OCIBindByName((stmp), &(dfnp), (errp), (pos), (prog
v), (progvl), \
        (ftype), NULL, NULL, NULL, OCI_DEFAULT); \

```

```

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

OCIDefineByPos((stmp)&(dfnp),(errp),(pos),(prog
v),(progv),(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[20];
! 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 plsflag;
!};
!
!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 stoostruct {
! int terror;
! int low_stock;
! int retry;
!};
!
!struct stostruct {
! struct stoinstruct stoin;
! struct stoostruct stoo;
!};
#endif
#endif
.....
SvrDBDepend_Oracle/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.
+=====
=====*/
#endif 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 ALLTXNODEL 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
.....
SvrDBDepend_Oracle/tpccflags.h
.....
#define DMLRETDDEL
#ifndef TSL
#define USE_IEEE_NUMBER
#endif
.....
SvrDBDepend_Oracle/tpccpl.cpp
.....
#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 <WINDOWS.H>
#include <stdio.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 "tpccsvr_Obj.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 "dpbccore.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 CDBAccess::ocierror(char *fname, int lineno,
OCIError *errhp, sword status)
{
text errbuf[512];
sb4 errcode;
sb4 lstat;
ub4 recno=2;
switch (status) {
case OCI_SUCCESS:
break;
case OCI_SUCCESS_WITH_INFO:
userlog(LOG_FILE_LINE, "Module %s
Line %d\n", fname, lineno);
userlog(LOG_FILE_LINE, "Error -
OCI_SUCCESS_WITH_INFO\n");
lstat = OCIErrorGet (errhp, recno++, (text *)
NULL, &errcode, errbuf,
(ub4) sizeof(errbuf),
OCI_HTYPE_ERROR);
userlog(LOG_FILE_LINE, "Error - %s\n",
errbuf);
break;
case OCI_NEED_DATA:
userlog(LOG_FILE_LINE, "Module %s
Line %d\n", fname, lineno);
userlog(LOG_FILE_LINE, "Error -
OCI_NEED_DATA\n");
return (IRRECERR);
case OCI_NO_DATA:
userlog(LOG_FILE_LINE, "Module %s
Line %d\n", fname, lineno);
userlog(LOG_FILE_LINE, "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) {
userlog(LOG_FILE_LINE, "Module %s
Line %d\n", fname, lineno);
userlog(LOG_FILE_LINE, "Information -
NOT_SERIALIZABLE (OCI_ERROR)\n");
return (errcode);
}
if (errcode == SNAPSHOT_TOO_OLD) {
userlog(LOG_FILE_LINE, "Module %s
Line %d\n", fname, lineno);
userlog(LOG_FILE_LINE, "Information -
SNAPSHOT_TOO_OLD (OCI_ERROR)\n");
return (errcode);
}
/* Replaced end */
while (lstat != OCI_NO_DATA)
{
userlog(LOG_FILE_LINE, "Module %s
Line %d\n", fname, lineno);
userlog(LOG_FILE_LINE, "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:
userlog(LOG_FILE_LINE, "Module %s
Line %d\n", fname, lineno);
userlog(LOG_FILE_LINE, "Error -
OCI_INVALID_HANDLE\n");
/* Replaced 03.05.15 TPCexit no argument */
// TPCexit(1);
TPCexit();
/* Replaced end */
exit(-1);
case OCI_STILL_EXECUTING:
userlog(LOG_FILE_LINE, "Module %s
Line %d\n", fname, lineno);
userlog(LOG_FILE_LINE, "Error -
OCI_STILL_EXECUTING\n");
return (IRRECERR);
case OCI_CONTINUE:
userlog(LOG_FILE_LINE, "Module %s
Line %d\n", fname, lineno);
userlog(LOG_FILE_LINE, "Error -
OCI_CONTINUE\n");
return (IRRECERR);
default:

```

```

userlog(LOG_FILE_LINE, "Module %s
Line %d\n", fname, lineno);
userlog(LOG_FILE_LINE, "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) {
userlog(LOG_FILE_LINE, "fopen on %s
failed %d\n", fnam, fd);
exit(-1);
}
return (fd);
}
int CDBAccess::sqlfile(char *fnam, text "linebuf"
{
FILE *fd;
int nulpt = 0;
char realfile[512];
#ifdef DEBUG
fprintf(stderr, "sqlfile() fnam: %s, linebuf: %s\n",
fnam, linebuf);
#endif
/*
sprintf(realfile, "%s/bench/tpc/tpcc/blocks/%s", 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 *oradl)
{
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 CDBAccess::TPCexit (void)
{
    userlog(LOG_FILE_LINE, "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) {
    tkvcdone();
    del_init = 0;
}
}
#else
if (del_init_oci) {
    tkvcdone(0);
    del_init_oci = 0;
}
if (del_init_plsql) {
    tkvcdone(1);
    del_init_plsql = 0;
}
}
#endif
if (sto_init) {
    tkvcsdone();
    sto_init = 0;
}
/* Added T.Kato 030723 Can't disconnect server */
OCIERROR(errhp,OCISessionEnd(tpscvc,errhp,
tpcusr, OCI_DEFAULT));
OCIERROR(errhp,OCIServerDetach(tpcsrv,
errhp, OCI_DEFAULT));
/* Added end */
OCIHandleFree((dvoid *)tpcusr,
OCI_HTYPE_SESSION);
OCIHandleFree((dvoid *)tpscvc,
OCI_HTYPE_SVCCTX);
OCIHandleFree((dvoid *)errhp,
OCI_HTYPE_ERROR);
OCIHandleFree((dvoid *)tpcsrv,
OCI_HTYPE_SERVER);
OCIHandleFree((dvoid *)tpcenv,
OCI_HTYPE_ENV);
/* Close Delivery log */
if (lfp) {
    fclose (lfp);
    lfp = NULL;

```

```

    }
}
CDBAccess::TPCinit (int id, char *uid, char *pwd)
{
    /* Deleted T.Kato 02.10.24 Deleted delivery log
open
! char filename[40];
Deleted end */
text stmbuf[100];
/* Added T.Kato 02.10.24 */
sword rval;
/* Added End */
id;
/* 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) {
#ifndef TUX
! userlog ("Error in TPC-C server %d: Failed to
open %s\n",
! proc_no, filename);
#else
! fprintf (stderr, "Error in TPC-C server %d:
Failed to open %s\n",
! proc_no, filename);
#endif
! return (-1);
! }
#endif
// Init delivery flag
iflg = 0;
/* replaced end */
/* Replaced 03.05.19 For Thread */
#if 0
!
OCIInitialize(OCI_DEFAULT|OCI_OBJECT,(dvoid
*)0,0,0,0);
#endif

OCIInitialize(OCI_THREADED|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
**)&tpscvc, 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_SVCCCTX, (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_SVCCCTX,
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, SQLTX);
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, SQLTXTRC);
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_da
te));
if (tkvcninit ()) { /* new order */
TPCexit ();
return (-1);

```

```

}
else
new_init = 1;
if (tkvcpininit ()) { /* payment */
TPCexit ();
return (-1);
}
else
pay_init = 1;
if (tkvcoininit ()) { /* 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);
}
CDBAccess::TPCnew (struct newstruct *str)
{
/* Added T.Kato 02.11.25 */
#ifdef AVOID_DEADLOCK
static int
init_value_index[NITEMS]={0,1,2,3,4,5,6,7,8,9,10,
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;
#ifdef 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_da
te));
if ((str->newout.terror = tkvcn ()) != 0) {
if (str->newout.terror != RECOVERR)
str->newout.terror = IRECCERR;
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
/* 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 */
#ifdef AVOID_DEADLOCK
q_sort(indx, str, 0, ord_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);
}
CDBAccess::TPCpay (struct paystruct *str)
{
w_id = str->payin.w_id;
d_id = str->payin.d_id;
c_w_id = str->payin.c_w_id;
c_d_id = str->payin.c_d_id;

/* 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_da
te));
if (bylastname) {
c_id = 0;
strcpy (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);
}
/*
cvtmyhms(cr_date,h_date);
*/
hlen=SIZE(h_date);

OCIERROR(errhp,OCIDateToText(errhp,&cr_date,
(text*)FULLDATE,(ub1)strlen(FULLDATE),(text*)0
,0,&hlen,h_date));
/*
cvtmy(c_since,c_since_d);
*/
sincelen=SIZE(c_since_d);

OCIERROR(errhp,OCIDateToText(errhp,&c_sinc
e,
(text*)SHORTDATE,(ub1)strlen(SHORTDATE),(te
xt*)0,0,&sincelen,c_since_d);

str->payout.terror = NOERR;
strcpy (str->payout.w_street_1, w_street_1,
21);
strcpy (str->payout.w_street_2, w_street_2,
21);
strcpy (str->payout.w_city, w_city, 21);
strcpy (str->payout.w_state, w_state, 3);
strcpy (str->payout.w_zip, w_zip, 10);
strcpy (str->payout.d_street_1, d_street_1, 21);
strcpy (str->payout.d_street_2, d_street_2, 21);
strcpy (str->payout.d_city, d_city, 21);
strcpy (str->payout.d_state, d_state, 3);
strcpy (str->payout.d_zip, d_zip, 10);
str->payout.c_id = c_id;
strcpy (str->payout.c_first, c_first, 17);
strcpy (str->payout.c_middle, c_middle, 3);
strcpy (str->payout.c_last, c_last, 17);
strcpy (str->payout.c_street_1, c_street_1, 21);
strcpy (str->payout.c_street_2, c_street_2, 21);
strcpy (str->payout.c_city, c_city, 21);
strcpy (str->payout.c_state, c_state, 3);
strcpy (str->payout.c_zip, c_zip, 10);
strcpy (str->payout.c_phone, c_phone, 17);
strcpy (str->payout.c_since, (char*)c_since_d,
11);
strcpy (str->payout.c_credit, c_credit, 3);
/* Replaced T.Kato 03.08.15 */
/*str->payout.c_credit_lim =
(float)(c_credit_lim)/100;*/
long double long64bit;
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 */
strcpy (str->payout.c_data, c_data, 201);
strcpy (str->payout.h_date, (char*)h_date, 20);
str->payout.retry = retries;
return (0);
}
CDBAccess::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;
strcpy (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,SIZE(FULLDATE),(text*)0,0,
&datelen,o_entry_d));
str->ordout.terror = NOERR;
str->ordout.c_id = c_id;
strcpy (str->ordout.c_last, c_last, 17);
strcpy (str->ordout.c_first, c_first, 17);
strcpy (str->ordout.c_middle, c_middle, 3);
str->ordout.c_balance = c_balance/100;
str->ordout.o_id = o_id;
strcpy (str->ordout.o_entry_d, (char*)o_entry_d,
20);
if (o_carrier_id == 11)
str->ordout.o_carrier_id = 0;
else
str->ordout.o_carrier_id = o_carrier_id;
str->ordout.o_ol_cnt = o_ol_cnt;
for (i = 0; i < o_ol_cnt; i++) {
ol_delivery_d[i][10] = '\0';
if ( !strcmp((char*)ol_delivery_d[i],"15-09-
1911") )
strcpy((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);
}
CDBAccess::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;
SYSTEMTIME systemTime;
struct tm times;
char filename[40];
//int svrcnt;
/* Replaced end */
/* Added T.Kato 02.10.24 Open the delivery log
file */
if(!flg == 0)
{
// Execute first delivery transaction
/* Replaced T.Kato 03.04.14 Chaned file name of
delivery log */
#if 0
! srand ((unsigned)time( NULL ));
! svrcnt = rand();
! // Open delivery logging file
! sprintf (filename, "c:\tpcrt\tpcc\tpcc_%d.del",
svrcnt);
#endif
/* Replaced T.kato 03.05.28 Change file of
delivery log for object use */
/* sprintf (filename,
"c:\tpcrt\tpcc\tpcc_%05d.del",
GetCurrentThreadId());*/
sprintf (filename,
"c:\tpcrt\tpcc\tpcc_%08x.del", this);
/* Replaced end (for object use) */
/* Replaced end */
if ((flp = fopen (filename, "w")) == NULL) {
userlog (LOG_FILE_LINE, "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_da
te));
#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 = IRRRECERR;
return (-1);
}
/* Replaced T.Kato 02.10.24 Chened time stamp
method */
#if 0
! tr_end = gettime ();
! fprintf (flp, "%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
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(flp,"%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 */
for (i = 0; i < 10; i++) {
fprintf (flp, "%d %d", i + 1, del_o_id[i]);
if (del_o_id[i] <= 0) {
userlog (LOG_FILE_LINE, "DELIVERY: no
new order for w_id: %d, d_id %d\n",
w_id, i + 1);
}
}
fprintf (flp, "%d\n", retries);
str->delout.terror = NOERR;
str->delout.retry = retries;
return (0);
}
CDBAccess::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->stout.terror = tkvcs ()) != 0) {
if (str->stout.terror != RECOVERR)
str->stout.terror = IRRRECERR;
return (-1);
}
str->stout.terror = NOERR;
str->stout.low_stock = low_stock;
str->stout.retry = retries;
return (0);
}
#ifdef AVOID_DEADLOCK
/* Added T.Kato 02.11.22 */
void CDBAccess::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 CDBAccess::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 */
str;
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 CDBAccess::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 CDBAccess::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 Chnged Oracle 10i
tool kit */
    #if 0
    ! strcpy(tmpstr,str->newout.i_name[i]);
    ! strcpy(str->newout.l_name[i],str-
>newout.i_name[j]);
    ! strcpy(str->newout.l_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.l_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
.....
SvrDBDepend_Oracle/tpccsvr.cpp
.....
# 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.cpp
| DESCRIPTION
| Changed by T.Kato for New Client Application
(None TUXEDO)
| All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003
+=====
+=====*/
#include <windows.h>
#include <time.h> // for generation of random
seed for server id
#include "tpcc.h"
#include "tpcc_info.h"
#include "log.h"
#include "tpccsvr_Obj.h"
/* FUNCTION: void ExSvrAplInitiate ();
*
* PURPOSE: Initialize Server APL for Data Base
* ARGUMENTS: None
*
* RETURNS: None

```

```

*
* COMMENTS: None
*
*/
void ExSvrAplInitiate(void* prmPointer)
{
    int svrcnt;
    CDBAccess* pDbAccess;
    srand (( unsigned)time( NULL ) );
    svrcnt = rand();
    pDbAccess = (CDBAccess*)prmPointer;
    if (pDbAccess->TPCinit (svrcnt, "tpcc", "tpcc")) {
        userlog(LOG_FILE_LINE, "Failed in TPCinit
(probably connecting).");
        return;
    }
    return;
}
/* FUNCTION: void ExTrxStructSize ();
*
* PURPOSE: This function post the each struct
size of Transaction Data Area.
* ARGUMENTS: None
*
* RETURNS: None
*
* COMMENTS: None
*
*/
void ExTrxStructSize(int* size_table) {
    size_table[0] = sizeof(struct newstruct); //
size of NewOrder
    size_table[1] = sizeof(struct paystruct); //
size of Payment
    size_table[2] = sizeof(struct ordstruct); //
size of OrderStatus
    size_table[3] = sizeof(struct delstruct); //
size of Delivery
    size_table[4] = sizeof(struct stostruct); //
size of StockLevel
    return;
}
/* FUNCTION: void ExSrvAplTerminate ();
*
* PURPOSE: Terminate Server APL for Data
Base
* ARGUMENTS: None
*
* RETURNS: None
*
* COMMENTS: None
*
*/
void ExSrvAplTerminate(void* prmPointer)
{
    CDBAccess* pDbAccess;
    pDbAccess = (CDBAccess*)prmPointer;
    pDbAccess->TPCexit();
    return;
}
/* FUNCTION: int ExTrxNewOrder(void
*new_order_struct)
*
* PURPOSE: This function handles the new
order transaction.
*
* ARGUMENTS: new_order_struct : NewOrder
data struct

```

```

*
* RETURNS: int
*
* COMMENTS: None
*
*/
int ExTrxNewOrder(void *new_order_struct, void*
prmPointer)
{
    CDBAccess*      pDbAccess;
    pDbAccess = (CDBAccess*)prmPointer;
    pDbAccess->newinfo = (struct newstruct *)
new_order_struct;
    pDbAccess->newinfo->retval = pDbAccess-
>TPCnew (pDbAccess->newinfo); // set return
value to 0 or -1
    return(pDbAccess->newinfo->retval);
}
/* FUNCTION: int ExTrxPayment(void
*payment_struct)
*
* PURPOSE: This function handles the
payment transaction.
*
* ARGUMENTS: payment_struct : Payment
data struct
*
* RETURNS: int
*
* COMMENTS: None
*
*/
int ExTrxPayment(void *payment_struct, void*
prmPointer)
{
    CDBAccess*      pDbAccess;
    pDbAccess = (CDBAccess*)prmPointer;
    pDbAccess->payinfo = (struct paystruct *)
payment_struct;
    pDbAccess->payinfo->retval = pDbAccess-
>TPCpay (pDbAccess->payinfo); // set return
value to 1 or 0 or -1
    return(pDbAccess->payinfo->retval);
}
/* FUNCTION: int ExTrxOrderStatus(void
*order_status_struct)
*
* PURPOSE: This function handles the order
status transaction.
*
* ARGUMENTS: order_status_struct :
OrderStatus data struct
*
* RETURNS: int
*
* COMMENTS: None
*
*/
int ExTrxOrderStatus(void *order_status_struct,
void* prmPointer)
{
    CDBAccess*      pDbAccess;
    pDbAccess = (CDBAccess*)prmPointer;
    pDbAccess->ordinfo = (struct ordstruct *)
order_status_struct;
    pDbAccess->ordinfo->retval = pDbAccess-
>TPCord (pDbAccess->ordinfo); // set return
value to 0 or -1
    return(pDbAccess->ordinfo->retval);
}

```

```

}
/* FUNCTION: int ExTrxDelivery(void
*delivery_struct)
*
* PURPOSE: This function handles the delivery
transaction.
*
* ARGUMENTS: delivery_struct : Delivery data
struct
*
* RETURNS: int
*
* COMMENTS: None
*
*/
int ExTrxDelivery(void *delivery_struct, void*
prmPointer)
{
    CDBAccess*      pDbAccess;
    pDbAccess = (CDBAccess*)prmPointer;
    pDbAccess->delinfo = (struct delstruct *)
delivery_struct;
    pDbAccess->delinfo->retval = pDbAccess-
>TPCdel (pDbAccess->delinfo); // set return value
to 0 or -1
    return(pDbAccess->delinfo->retval);
}
/* FUNCTION: int ExTrxStockLevel(void
*stock_level_struct)
*
* PURPOSE: This function handles the stock
level transaction.
*
* ARGUMENTS: stock_level_struct :
StockLevel data struct
*
* RETURNS: int
*
* COMMENTS: None
*
*/
int ExTrxStockLevel(void *stock_level_struct, void*
prmPointer)
{
    CDBAccess*      pDbAccess;
    pDbAccess = (CDBAccess*)prmPointer;
    pDbAccess->stoinfo = (struct stostruct *)
stock_level_struct;
    pDbAccess->stoinfo->retval = pDbAccess-
>TPCsto (pDbAccess->stoinfo); // set return value
to 0 or -1
    return(pDbAccess->stoinfo->retval);
}
.....
SvrDBDepend_Oracle/tpccsvr_Obj.h
.....
*****
*
* TPC-C Client Application Program Source
*
*
* Entry Functions *
* SvrApl Object definition. *
*
* CREATE by TSL 2003.05.16
*
*

```

```

* All Right Reserved, Copyright Co. FUJITSU
LIMITED 2003 *
*****
**/
#include "tpccflags.h"
#include "TrmCntlInfo.h"
#include "GlobalArea.h"
class CDBAccess
{
public:
    CDBAccess();
    ~CDBAccess();
    /* ----- */
    /* Delivery (pldel.cpp) */
    /* ----- */
    pldelctx *pldctx;
    delctx *dctx;
#ifdef DMLRETDL
    amtctx *actx;
#endif
    /* ----- */
    /* NewOrder (plnew.cpp) */
    /* ----- */
    newctx *nctx;
    /* ----- */
    /* OrderStatus (plord.cpp) */
    /* ----- */
    ordctx *octx;
    defctx cbctx;
    /* ----- */
    /* 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_ORA81
    int del_init;
#else
    int del_init_oci;
    int del_init_plsql;
#endif
    int sto_init;
    int res_init;
    int execstatus;
    int errcode;
    OCIEncv *tpcenv;
    OCIServer *tpcsvr;
    OCIError *errhp;
    OCISvcCtx *tpcsvc;
    OCISession *tpcsur;
}

```

```

        OCISmt *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 sincelem;
    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 no_l_id[15];
    int no_supply_w_id[15];
/* Replaced T.kato 03.09.09 Oracle10g tool kit */
#if 0
!   int no_quantity[15];
!   int no_amount[15];
!   int s_quantity[15];
!   int i_price[15];
#endif
#ifdef USE_IEEE_NUMBER
float no_quantity[15];
float no_amount[15];
float s_quantity[15];
float i_price[15];
#else
int no_quantity[15];
int no_amount[15];
int s_quantity[15];
int i_price[15];
#endif /* USE_IEEE_NUMBER */
/* Replaced end */
    int no_quant10[15];
    int no_quant19[15];
    int no_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 stostruc *stoinfo;
    /* ----- */

```

```

        /* Prototype */
        /* ----- */
#ifdef DEL_ORA81
    tkvcddinit ();
    tkvcninit ();
    tkvcoint ();
    int tkvcpinit (void);
    tkvcsinit ();
    tkvcd ();
    tkvcn ();
    tkvcs ();
    tkvcp ();
    tkvco ();
    void tkvcddone ();
    void tkvcndone ();
    void tkvcsdone ();
    void tkvcpdone ();
    void tkvcodone ();
#endif
#else
    tkvcddinit (int plsqflflag);
    tkvcninit ();
    tkvcoint ();
    int tkvcpinit (void);
    tkvcsinit ();
    tkvcd (int plsqflflag);
    tkvcn ();
    tkvcs ();
    tkvcp ();
    tkvco ();
    void tkvcddone (int plsqflflag);
    void tkvcndone ();
    void tkvcsdone ();
    void tkvcpdone ();
    void tkvcodone ();
#endif
/* pldel */
void shiftdata(int from);
/* tpccpl Prototype */
TPCinit (int id, char* uid, char* pwd);
TPCnew (struct newstruct* str);
TPCdel (struct delstruct* str);
TPCpay (struct paystruct* str);
TPCord (struct ordstruct* str);
TPCsto (struct stostruc* str);
void TPCexit (void);
int ocierror(char* fname, int lineno,
OCIError* errhp, sword status);
int sqlfile(char* fname, text* linebuf);
#ifdef AVOID_DEADLOCK
    int indx[NITEMS], ordL_cnt;
    /* Added T.Kato 02.11.22 */
    void swap_item(struct newstruct *str, int i, int
));
    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
);
;
; SvrProxy/SvrProxy.def
LIBRARY SvrProxy.dll
DESCRIPTION 'SvrApi Proxy/Stub DLL'
EXPORTS

```

```

DllGetClassObject @1 PRIVATE
DllCanUnloadNow @2
PRIVATE
    DllRegisterServer @3 PRIVATE
    DllUnregisterServer @4 PRIVATE
.....
SvrProxy/SvrProxy.mak
.....
# Microsoft Developer Studio Generated NMAKE
File, Based on SvrProxy.dsp
!IF "$(CFG)" == ""
CFG=SvrProxy - Win32 Debug
!MESSAGE SvrProxy - Win32 Debug
!ENDIF
!IF "$(CFG)" != "SvrProxy - Win32 Release" &&
"$(CFG)" != "SvrProxy - Win32 Debug"
!MESSAGE "$(CFG)"
!MESSAGE NMAKE
!MESSAGE :
!MESSAGE
!MESSAGE NMAKE /f "SvrProxy.mak"
CFG="SvrProxy - Win32 Debug"
!MESSAGE
!MESSAGE :
!MESSAGE
!MESSAGE "SvrProxy - Win32 Release" ("Win32
(x86) Dynamic-Link Library")
!MESSAGE "SvrProxy - Win32 Debug" ("Win32
(x86) Dynamic-Link Library")
!MESSAGE
!ERROR
!ENDIF
!IF "$(OS)" == "Windows_NT"
NULL=
!ELSE
NULL=nul
!ENDIF
!IF "$(CFG)" == "SvrProxy - Win32 Release"
OUTDIR=.\\Release
INTDIR=.\\Release
# Begin Custom Macros
OutDir=.\\Release
# End Custom Macros
ALL : "$(OUTDIR)\SvrProxy.dll"
CLEAN :
    -@erase "$(INTDIR)\dlldata.obj"
    -@erase "$(INTDIR)\SvrApiBase_i.obj"
    -@erase "$(INTDIR)\SvrApiBase_p.obj"
    -@erase "$(INTDIR)\vc60.idb"
    -@erase "$(OUTDIR)\SvrProxy.dll"
    -@erase "$(OUTDIR)\SvrProxy.exp"
    -@erase "$(OUTDIR)\SvrProxy.lib"
    -@erase "$(OUTDIR)\SvrProxy.pdb"
"$(OUTDIR)" :
    if not exist "$(OUTDIR)\$(NULL)" mkdir
"$(OUTDIR)"
CPP=cl.exe
CPP_PROJ=/nologo /MD /W3 /GX /O2 /D
"NDEBUG" /D "WIN32" /D "_WINDOWS" /D
"_MBCS" /D "_USRDLL" /D
"SVRPROXY_EXPORTS" /D
"REGISTER_PROXY_DLL" /D "_WIN32_DCOM"
/D "_WIN32_WINNT=0x0400
/Fp "$(INTDIR)\SvrProxy.pch" /YX
/Fo "$(INTDIR)\\" /Fd "$(INTDIR)\\" /FD /c
.c $(INTDIR).obj:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cpp $(INTDIR).obj:

```

```

$(CPP) @<<
$(CPP_PROJ) $<
<<
.cxx $(INTDIR).obj:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.c $(INTDIR).sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cpp $(INTDIR).sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
.cxx $(INTDIR).sbr:
$(CPP) @<<
$(CPP_PROJ) $<
<<
MTL=midl.exe
MTL_PROJ=/nologo /D "NDEBUG" /mktyplib203
/win32
RSC=rc.exe
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/o "$(OUTDIR)\SvrProxy.bsc"
BSC32_SBRS= \

LINK32=link.exe
LINK32_FLAGS=rpcndr.lib rpcns4.lib rpctr4.lib
kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /dll /incremental:no
/pdb:"$(OUTDIR)\SvrProxy.pdb" /machine:i386
/def:".\SvrProxy.def"
/out:"$(OUTDIR)\SvrProxy.dll"
/implib:"$(OUTDIR)\SvrProxy.lib"
DEF_FILE= \
    ".\SvrProxy.def"
LINK32_OBJS= \
    "$(INTDIR)\dlldata.obj" \
    "$(INTDIR)\SvrApiBase_i.obj" \
    "$(INTDIR)\SvrApiBase_p.obj"
"$(OUTDIR)\SvrProxy.dll" : "$(OUTDIR)"
$(DEF_FILE) $(LINK32_OBJS)
    $(LINK32) @<<
    $(LINK32_FLAGS) $(LINK32_OBJS)
<<
SOURCE="$(InputPath)"
PostBuild_Desc=bin
DS_POSTBUILD_DEP=$(INTDIR)\postbld.dep
ALL : $(DS_POSTBUILD_DEP)
# Begin Custom Macros
OutDir=.\\Release
# End Custom Macros
$(DS_POSTBUILD_DEP) :
"$(OUTDIR)\SvrProxy.dll"

copy .\\Release\\SvrProxy.dll ..\\bin\\Release\\SvrPro
xy.dll

echo Helper for Post-build step >
"$(DS_POSTBUILD_DEP)"
!ELSEIF "$(CFG)" == "SvrProxy - Win32 Debug"
OUTDIR=.\\Debug
INTDIR=.\\Debug
# Begin Custom Macros
OutDir=.\\Debug
# End Custom Macros

```

```

ALL : "$(OUTDIR)\SvrProxy.dll"
CLEAN :
    -@erase "$(INTDIR)\dlldata.obj"
    -@erase "$(INTDIR)\SvrApiBase_i.obj"
    -@erase "$(INTDIR)\SvrApiBase_p.obj"
    -@erase "$(INTDIR)\vc60.idb"
    -@erase "$(INTDIR)\vc60.pdb"
    -@erase "$(OUTDIR)\SvrProxy.dll"
    -@erase "$(OUTDIR)\SvrProxy.exp"
    -@erase "$(OUTDIR)\SvrProxy.ilk"
    -@erase "$(OUTDIR)\SvrProxy.lib"
    -@erase "$(OUTDIR)\SvrProxy.pdb"
"$(OUTDIR)" :
    if not exist "$(OUTDIR)\$(NULL)" mkdir
"$(OUTDIR)"
CPP=cl.exe
CPP_PROJ=/nologo /MTd /W3 /Gm /GX /ZI /Od
/D "_DEBUG" /D "WIN32" /D "_WINDOWS" /D
"_MBCS" /D "_USRDLL" /D
"SVRPROXY_EXPORTS" /D
"REGISTER_PROXY_DLL" /D "_WIN32_DCOM"
/D "_WIN32_WINNT=0x0400
/Fp "$(INTDIR)\SvrProxy.pch" /YX
/Fo "$(INTDIR)\\" /Fd "$(INTDIR)\\" /FD /GZ /c
.c $(INTDIR).obj:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cpp $(INTDIR).obj:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cxx $(INTDIR).obj:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.c $(INTDIR).sbr:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cpp $(INTDIR).sbr:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
.cxx $(INTDIR).sbr:
    $(CPP) @<<
    $(CPP_PROJ) $<
<<
MTL=midl.exe
MTL_PROJ=/nologo /D "_DEBUG" /mktyplib203
/win32
RSC=rc.exe
BSC32=bscmake.exe
BSC32_FLAGS=/nologo
/o "$(OUTDIR)\SvrProxy.bsc"
BSC32_SBRS= \

LINK32=link.exe
LINK32_FLAGS=rpcndr.lib rpcns4.lib rpctr4.lib
kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /dll /incremental:yes
/pdb:"$(OUTDIR)\SvrProxy.pdb" /debug
/machine:i386 /def:".\SvrProxy.def"
/out:"$(OUTDIR)\SvrProxy.dll"
/implib:"$(OUTDIR)\SvrProxy.lib" /pdbtype:sept
DEF_FILE= \
    ".\SvrProxy.def"

```

```

LINK32_OBJS= \
    "$(INTDIR)\dlldata.obj" \
    "$(INTDIR)\SvrAplBase_i.obj" \
    "$(INTDIR)\SvrAplBase_p.obj"
"$(OUTDIR)\SvrProxy.dll" : "$(OUTDIR)"
$(DEF_FILE) $(LINK32_OBJS)
$(LINK32) @<<
$(LINK32_FLAGS) $(LINK32_OBJS)
<<
SOURCE="$(InputPath)"
PostBuild_Desc=bin
DS_POSTBUILD_DEP=$(INTDIR)\postbld.dep
ALL : $(DS_POSTBUILD_DEP)
# Begin Custom Macros
OutDir=.Debug
# End Custom Macros
$(DS_POSTBUILD_DEP) :
"$(OUTDIR)\SvrProxy.dll"

copy .Debug\SvrProxy.dll ..bin\Debug\SvrProxy.d
ll
    echo Helper for Post-build step >
"$(DS_POSTBUILD_DEP)"
!ENDIF
!IF "$(NO_EXTERNAL_DEPS)" != "1"
!IF EXISTS("SvrProxy.dep")
!INCLUDE "SvrProxy.dep"
!ELSE
!MESSAGE Warning: cannot find "SvrProxy.dep"
!ENDIF
!ENDIF
!IF "$(CFG)" == "SvrProxy - Win32 Release" ||
"$(CFG)" == "SvrProxy - Win32 Debug"
SOURCE=..\SvrAplBase\dlldata.c
"$(INTDIR)\dlldata.obj" : $(SOURCE) "$(INTDIR)"
$(CPP) $(CPP_PROJ) $(SOURCE)
SOURCE=..\SvrAplBase\SvrAplBase_i.c
"$(INTDIR)\SvrAplBase_i.obj" : $(SOURCE)
"$(INTDIR)"
$(CPP) $(CPP_PROJ) $(SOURCE)
SOURCE=..\SvrAplBase\SvrAplBase_p.c
"$(INTDIR)\SvrAplBase_p.obj" : $(SOURCE)
"$(INTDIR)"
$(CPP) $(CPP_PROJ) $(SOURCE)
!ENDIF
.....
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
inittpc.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
inittpc.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 = inittpc.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
inittpc.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, inittpc.ware_name || ' ' ||
inittpc.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 inittpc.ware_name,
:w_street_1, :w_street_2, :w_city, :w_s
tate, :w_zip;
SELECT rowid
BULK COLLECT INTO inittpc.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;
inittpc.c_num := sql%rowcount;
inittpc.cust_rowid :=
inittpc.row_id((inittpc.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 = inittpc.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 = inittpc.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/tkvcin.sql
.....
-- The initnew package for storing variables used
in the
-- New Order anonymous block
CREATE OR REPLACE PACKAGE inittpcc
AS
TYPE intarray IS TABLE OF INTEGER INDEX
BY BINARY_INTEGER;
TYPE distarray IS TABLE OF VARCHAR(24)
INDEX BY BINARY_INTEGER;
nulldate DATE;
TYPE rowidarray IS TABLE OF ROWID INDEX
BY PLS_INTEGER;
s_dist distarray;
idx1arr intarray;
s_remote intarray;
dist intarray;
row_id rowidarray;
cust_rowid rowid;
dist_name VARCHAR2(11);
ware_name VARCHAR2(11);
c_num PLS_INTEGER;
PROCEDURE init_no(idxarr intarray);
PROCEDURE init_del;
PROCEDURE init_pay;
END 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/tkvcpd.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 ord 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 ord SET o_delivery_d = :now
WHERE o_w_id = :w_id
AND o_d_id = :d_id(o)
AND o_o_id = :order_id(o)
RETURNING sum(o_l_amount) BULK
COLLECT INTO :sums;
FORALL c IN 1.. :ordcnt
UPDATE cust
SET c_balance = c_balance + :sums(c),
c_delivery_cnt = c_delivery_cnt +
1
WHERE c_w_id = :w_id
AND c_d_id = :d_id(c)
AND c_id = :o_c_id(c);
COMMIT;
EXIT;

```

```

EXCEPTION
WHEN not_serializable OR deadlock OR
snapshot_too_old
THEN
ROLLBACK;
:retry := :retry + 1;
END;
END LOOP; -- for retry
END;
.....
blocks/tkvcnew.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, inittpc.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, inittpc.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, inittpc.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, inittpc.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, inittpc.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, inittpc.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, inittpc.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
END
BULK COLLECT
INTO :i_price, :i_name, :s_quantity, inittpc.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
      END
    END
  BULK COLLECT
  INTO :i_price, :i_name, :s_quantity, inittpc.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);
  inittpc.s_dist(temp_index + 1) :=
inittpc.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;
  inittpc.s_dist(idx + rows_lost) := NULL;
  :brand_generic(idx + rows_lost) := '';
  :ol_amount(idx + rows_lost) := 0;
  rows_lost := rows_lost + 1;
  max_index := max_index + 1;
  END IF;
  END LOOP;
  END fix_items;
  BEGIN
  LOOP BEGIN
  cache_ol_cnt := :o_ol_cnt;

  UPDATE dist SET d_next_o_id =
d_next_o_id + 1
  WHERE d_id = :d_id AND d_w_id = :w_id
  RETURNING d_tax, d_next_o_id-1
  INTO :d_tax, :o_id;
  SELECT c_discount, c_last, c_credit, w_tax
  INTO :c_discount, :c_last, :c_credit , :w_tax
  FROM cust , ware
  WHERE c_id = :c_id AND c_d_id = :d_id
  AND c_w_id = :w_id
  AND w_id = :w_id;
  INSERT INTO nord (no_o_id, no_d_id,
no_w_id)
  VALUES (:o_id, :d_id, :w_id);
  INSERT INTO ord (o_id,o_d_id, o_w_id,
o_c_id, o_entry_d,
o_carrier_id, o_ol_cnt,
o_all_local)
  VALUES (:o_id, :d_id, :w_id, :c_id,
:cr_date, 11, :o_ol_cnt, :o_all_local);
  dummy_local := :d_id;
  IF (dummy_local < 6) THEN
  IF (dummy_local < 3) THEN
  IF (dummy_local = 1) THEN
  u1;
  ELSE
  u2;
  END IF;
  ELSE
  IF (dummy_local = 3) THEN

```

```

  u3;
  ELSIF (dummy_local = 4) then
  u4;
  ELSE
  u5;
  END IF;
  END IF;
  ELSE
  IF (dummy_local < 8) THEN
  IF (dummy_local = 6) THEN
  u6;
  ELSE
  u7;
  END IF;
  ELSE
  IF (dummy_local = 8) THEN
  u8;
  ELSIF (dummy_local = 9) then
  u9;
  ELSE
  u10;
  END IF;
  END IF;
  END IF;
  dummy_local := sql%rowcount;
  IF (dummy_local != cache_ol_cnt) THEN
  fix_items; END IF;
  FORALL idx IN 1..dummy_local
  INSERT INTO ordl
  (ol_o_id, ol_d_id, ol_w_id, ol_number,
ol_delivery_d, ol_i_id,
ol_supply_w_id,
ol_quantity,ol_amount,ol_dist_info)
  VALUES (:o_id, :d_id, :w_id,
inittpc.idx1arr(idx), inittpc.nulldate,
:ol_i_id(idx), :ol_supply_w_id(idx),
:ol_quantity(idx), :ol_amount(idx),
inittpc.s_dist(idx));
  IF (dummy_local != :o_ol_cnt) THEN
  :o_ol_cnt := dummy_local;
  ROLLBACK;
  END IF;
  EXIT;
  EXCEPTION
  WHEN not_serializable OR deadlock OR
snapshot_too_old THEN
  ROLLBACK;
  :retry := :retry + 1;
  END;
  END LOOP;
  END;

```

Appendix C: RTE Scripts

```

.....
RTE/rte01.conf
.....
#
#   rte01.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
  STARTRTE
    RTEHOST = rte01
    STARTSUT
      SUTHOST = cl001a
      SUTTERM = 1538
      SUTPASS = /w00
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl001a
      SUTTERM = 1538
      SUTPASS = /w01
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl001a
      SUTTERM = 1538
      SUTPASS = /w02
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl001a
      SUTTERM = 1538
      SUTPASS = /w03
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl001a
      SUTTERM = 1538
      SUTPASS = /w04
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl001a
      SUTTERM = 1538
      SUTPASS = /w05
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl001a
      SUTTERM = 2
      SUTPASS = /w06
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl002a
      SUTTERM = 1536
      SUTPASS = /w06
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl002a
      SUTTERM = 1538

```

```

      SUTPASS = /w07
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl002a
      SUTTERM = 1538
      SUTPASS = /w08
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl002a
      SUTTERM = 1538
      SUTPASS = /w09
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl002a
      SUTTERM = 1538
      SUTPASS = /w10
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl002a
      SUTTERM = 1542
      SUTPASS = /w11
      SUTLOG = 0
    ENDSUT
  ENDRTE
  PORT = 80
  SYNC = 0
  STARTVARIABLE
    VAL = U11 = 48000
    VAL = U21 = 0
    VAL = U31 = 14400
    VAL = U41 = 0
    VAL = U51 = 12010
    VAL = U61 = 12010
    VAL = U101 = 10020
    VAL = U111 = 5010
    VAL = U121 = 5010
    VAL = U171 = 18010
    VAL = U181 = 3020
    VAL = U191 = 2010
    VAL = U201 = 2010
    VAL = U211 = 2010
    VAL = U131 = 1023
    VAL = U141 = 255
    VAL = U151 = 8191
  ENDVARIABLE
ENDGROUP
.....
RTE/rte02.conf
.....
#
#   rte02.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
  STARTRTE
    RTEHOST = rte02
    STARTSUT
      SUTHOST = cl004a
      SUTTERM = 1538
      SUTPASS = /w00
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl004a

```

```

      SUTTERM = 1538
      SUTPASS = /w01
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl004a
      SUTTERM = 1538
      SUTPASS = /w02
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl004a
      SUTTERM = 1538
      SUTPASS = /w03
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl004a
      SUTTERM = 1538
      SUTPASS = /w04
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl004a
      SUTTERM = 2
      SUTPASS = /w06
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl005a
      SUTTERM = 1536
      SUTPASS = /w06
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl005a
      SUTTERM = 1538
      SUTPASS = /w07
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl005a
      SUTTERM = 1538
      SUTPASS = /w08
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl005a
      SUTTERM = 1538
      SUTPASS = /w09
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl005a
      SUTTERM = 1538
      SUTPASS = /w10
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl005a
      SUTTERM = 1542
      SUTPASS = /w11
      SUTLOG = 0

```

```

        ENDSUT
    ENDRTE
    PORT = 80
    SYNC = 0
    STARTVARIABLE
        VAL = U1I = 48000
        VAL = U2I = 0
        VAL = U3I = 14400
        VAL = U4I = 0
        VAL = U5I = 12010
        VAL = U6I = 12010
        VAL = U10I = 10020
        VAL = U11I = 5010
        VAL = U12I = 5010
        VAL = U17I = 18010
        VAL = U18I = 3020
        VAL = U19I = 2010
        VAL = U20I = 2010
        VAL = U21I = 2010
        VAL = U13I = 1023
        VAL = U14I = 255
        VAL = U15I = 8191
    ENDVARIABLE
ENDGROUP

.....
RTE/rte03.conf
.....
#
#   rte03.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
    STARTRTE
        RTEHOST = rte03
    STARTSUT
        SUTHOST = cl007a
        SUTTERM = 1538
        SUTPASS = /w00
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl007a
        SUTTERM = 1538
        SUTPASS = /w01
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl007a
        SUTTERM = 1538
        SUTPASS = /w02
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl007a
        SUTTERM = 1538
        SUTPASS = /w03
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl007a
        SUTTERM = 1538
        SUTPASS = /w04
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl007a
        SUTTERM = 1538

```

```

        SUTPASS = /w05
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl007a
        SUTTERM = 2
        SUTPASS = /w06
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl008a
        SUTTERM = 1536
        SUTPASS = /w06
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl008a
        SUTTERM = 1538
        SUTPASS = /w07
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl008a
        SUTTERM = 1538
        SUTPASS = /w08
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl008a
        SUTTERM = 1538
        SUTPASS = /w09
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl008a
        SUTTERM = 1538
        SUTPASS = /w10
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl008a
        SUTTERM = 1542
        SUTPASS = /w11
        SUTLOG = 0
    ENDSUT
    ENDRTE
    PORT = 80
    SYNC = 0
    STARTVARIABLE
        VAL = U1I = 48000
        VAL = U2I = 0
        VAL = U3I = 14400
        VAL = U4I = 0
        VAL = U5I = 12010
        VAL = U6I = 12010
        VAL = U10I = 10020
        VAL = U11I = 5010
        VAL = U12I = 5010
        VAL = U17I = 18010
        VAL = U18I = 3020
        VAL = U19I = 2010
        VAL = U20I = 2010
        VAL = U21I = 2010
        VAL = U13I = 1023
        VAL = U14I = 255
        VAL = U15I = 8191
    ENDVARIABLE
ENDGROUP

```

```

.....
RTE/rte04.conf
.....
#
#   rte04.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
    STARTRTE
        RTEHOST = rte04
    STARTSUT
        SUTHOST = cl010a
        SUTTERM = 1538
        SUTPASS = /w00
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl010a
        SUTTERM = 1538
        SUTPASS = /w01
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl010a
        SUTTERM = 1538
        SUTPASS = /w02
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl010a
        SUTTERM = 1538
        SUTPASS = /w03
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl010a
        SUTTERM = 1538
        SUTPASS = /w04
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl010a
        SUTTERM = 1538
        SUTPASS = /w05
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl010a
        SUTTERM = 2
        SUTPASS = /w06
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl011a
        SUTTERM = 1536
        SUTPASS = /w06
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl011a
        SUTTERM = 1538
        SUTPASS = /w07
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl011a
        SUTTERM = 1538

```

```

        SUTPASS = /w08
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl011a
        SUTTERM = 1538
        SUTPASS = /w09
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl011a
        SUTTERM = 1538
        SUTPASS = /w10
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl011a
        SUTTERM = 1542
        SUTPASS = /w11
        SUTLOG = 0
    ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
    VAL = U11 = 48000
    VAL = U21 = 0
    VAL = U31 = 14400
    VAL = U41 = 0
    VAL = U51 = 12010
    VAL = U61 = 12010
    VAL = U101 = 10020
    VAL = U111 = 5010
    VAL = U121 = 5010
    VAL = U171 = 18010
    VAL = U181 = 3020
    VAL = U191 = 2010
    VAL = U201 = 2010
    VAL = U211 = 2010
    VAL = U131 = 1023
    VAL = U141 = 255
    VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte05.conf
.....
#
#   rte05.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
    STARTRTE
        RTEHOST = rte05
        STARTSUT
            SUTHOST = cl013a
            SUTTERM = 1538
            SUTPASS = /w00
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl013a
            SUTTERM = 1538
            SUTPASS = /w01
            SUTLOG = 0
        ENDSUT
        STARTSUT
    ENDRTE
    PORT = 80
    SYNC = 0
    STARTVARIABLE

```

```

        SUTHOST = cl013a
        SUTTERM = 1538
        SUTPASS = /w02
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl013a
        SUTTERM = 1538
        SUTPASS = /w03
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl013a
        SUTTERM = 1538
        SUTPASS = /w04
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl013a
        SUTTERM = 1538
        SUTPASS = /w05
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl013a
        SUTTERM = 2
        SUTPASS = /w06
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl014a
        SUTTERM = 1536
        SUTPASS = /w06
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl014a
        SUTTERM = 1538
        SUTPASS = /w07
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl014a
        SUTTERM = 1538
        SUTPASS = /w08
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl014a
        SUTTERM = 1538
        SUTPASS = /w09
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl014a
        SUTTERM = 1538
        SUTPASS = /w10
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl014a
        SUTTERM = 1542
        SUTPASS = /w11
        SUTLOG = 0
    ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE

```

```

        VAL = U11 = 48000
        VAL = U21 = 0
        VAL = U31 = 14400
        VAL = U41 = 0
        VAL = U51 = 12010
        VAL = U61 = 12010
        VAL = U101 = 10020
        VAL = U111 = 5010
        VAL = U121 = 5010
        VAL = U171 = 18010
        VAL = U181 = 3020
        VAL = U191 = 2010
        VAL = U201 = 2010
        VAL = U211 = 2010
        VAL = U131 = 1023
        VAL = U141 = 255
        VAL = U151 = 8191
    ENDVARIABLE
ENDGROUP
.....
RTE/rte06.conf
.....
#
#   rte06.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
    STARTRTE
        RTEHOST = rte06
        STARTSUT
            SUTHOST = cl016a
            SUTTERM = 1538
            SUTPASS = /w00
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl016a
            SUTTERM = 1538
            SUTPASS = /w01
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl016a
            SUTTERM = 1538
            SUTPASS = /w02
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl016a
            SUTTERM = 1538
            SUTPASS = /w03
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl016a
            SUTTERM = 1538
            SUTPASS = /w04
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl016a
            SUTTERM = 1538
            SUTPASS = /w05
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl016a
            SUTTERM = 2
    ENDRTE
    PORT = 80
    SYNC = 0
    STARTVARIABLE

```

```

SUTPASS = /w06
SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl017a
  SUTTERM = 1536
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl017a
  SUTTERM = 1538
  SUTPASS = /w07
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl017a
  SUTTERM = 1538
  SUTPASS = /w08
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl017a
  SUTTERM = 1538
  SUTPASS = /w09
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl017a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl017a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
  VAL = U11 = 48000
  VAL = U21 = 0
  VAL = U31 = 14400
  VAL = U41 = 0
  VAL = U51 = 12010
  VAL = U61 = 12010
  VAL = U101 = 10020
  VAL = U111 = 5010
  VAL = U121 = 5010
  VAL = U171 = 18010
  VAL = U181 = 3020
  VAL = U191 = 2010
  VAL = U201 = 2010
  VAL = U211 = 2010
  VAL = U131 = 1023
  VAL = U141 = 255
  VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte07.conf
.....
#
#   rte07.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#

```

```

#
STARTGROUP
  STARTRTE
    RTEHOST = rte07
    STARTSUT
      SUTHOST = cl019a
      SUTTERM = 1538
      SUTPASS = /w00
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl019a
      SUTTERM = 1538
      SUTPASS = /w01
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl019a
      SUTTERM = 1538
      SUTPASS = /w02
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl019a
      SUTTERM = 1538
      SUTPASS = /w03
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl019a
      SUTTERM = 1538
      SUTPASS = /w04
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl019a
      SUTTERM = 1538
      SUTPASS = /w05
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl019a
      SUTTERM = 2
      SUTPASS = /w06
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl020a
      SUTTERM = 1536
      SUTPASS = /w06
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl020a
      SUTTERM = 1538
      SUTPASS = /w07
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl020a
      SUTTERM = 1538
      SUTPASS = /w08
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl020a
      SUTTERM = 1538
      SUTPASS = /w09
      SUTLOG = 0
    ENDSUT

```

```

ENDSUT
STARTSUT
  SUTHOST = cl020a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl020a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
  VAL = U11 = 48000
  VAL = U21 = 0
  VAL = U31 = 14400
  VAL = U41 = 0
  VAL = U51 = 12010
  VAL = U61 = 12010
  VAL = U101 = 10020
  VAL = U111 = 5010
  VAL = U121 = 5010
  VAL = U171 = 18010
  VAL = U181 = 3020
  VAL = U191 = 2010
  VAL = U201 = 2010
  VAL = U211 = 2010
  VAL = U131 = 1023
  VAL = U141 = 255
  VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte08.conf
.....
#
#   rte08.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
  STARTRTE
    RTEHOST = rte08
    STARTSUT
      SUTHOST = cl022a
      SUTTERM = 1538
      SUTPASS = /w00
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl022a
      SUTTERM = 1538
      SUTPASS = /w01
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl022a
      SUTTERM = 1538
      SUTPASS = /w02
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl022a
      SUTTERM = 1538
      SUTPASS = /w03
      SUTLOG = 0
    ENDSUT

```

```

SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl022a
  SUTTERM = 1538
  SUTPASS = /w04
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl022a
  SUTTERM = 1538
  SUTPASS = /w05
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl022a
  SUTTERM = 2
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl023a
  SUTTERM = 1536
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl023a
  SUTTERM = 1538
  SUTPASS = /w07
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl023a
  SUTTERM = 1538
  SUTPASS = /w08
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl023a
  SUTTERM = 1538
  SUTPASS = /w09
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl023a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl023a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
  VAL = U11 = 48000
  VAL = U21 = 0
  VAL = U31 = 14400
  VAL = U41 = 0
  VAL = U51 = 12010
  VAL = U61 = 12010
  VAL = U101 = 10020
  VAL = U111 = 5010
  VAL = U121 = 5010

```

```

VAL = U171 = 18010
VAL = U181 = 3020
VAL = U191 = 2010
VAL = U201 = 2010
VAL = U211 = 2010
VAL = U131 = 1023
VAL = U141 = 255
VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte09.conf
.....
#
#   rte09.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
STARTRTE
  RTEHOST = rte09
STARTSUT
  SUTHOST = cl025a
  SUTTERM = 1538
  SUTPASS = /w00
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl025a
  SUTTERM = 1538
  SUTPASS = /w01
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl025a
  SUTTERM = 1538
  SUTPASS = /w02
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl025a
  SUTTERM = 1538
  SUTPASS = /w03
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl025a
  SUTTERM = 1538
  SUTPASS = /w04
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl025a
  SUTTERM = 1538
  SUTPASS = /w05
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl025a
  SUTTERM = 2
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl026a
  SUTTERM = 1536
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT

```

```

STARTSUT
  SUTHOST = cl026a
  SUTTERM = 1538
  SUTPASS = /w07
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl026a
  SUTTERM = 1538
  SUTPASS = /w08
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl026a
  SUTTERM = 1538
  SUTPASS = /w09
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl026a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
  VAL = U11 = 48000
  VAL = U21 = 0
  VAL = U31 = 14400
  VAL = U41 = 0
  VAL = U51 = 12010
  VAL = U61 = 12010
  VAL = U101 = 10020
  VAL = U111 = 5010
  VAL = U121 = 5010
  VAL = U171 = 18010
  VAL = U181 = 3020
  VAL = U191 = 2010
  VAL = U201 = 2010
  VAL = U211 = 2010
  VAL = U131 = 1023
  VAL = U141 = 255
  VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte10.conf
.....
#
#   rte10.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
STARTRTE
  RTEHOST = rte10
STARTSUT
  SUTHOST = cl028a
  SUTTERM = 1538
  SUTPASS = /w00

```

```

SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl028a
  SUTTERM = 1538
  SUTPASS = /w01
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl028a
  SUTTERM = 1538
  SUTPASS = /w02
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl028a
  SUTTERM = 1538
  SUTPASS = /w03
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl028a
  SUTTERM = 1538
  SUTPASS = /w04
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl028a
  SUTTERM = 1538
  SUTPASS = /w05
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl028a
  SUTTERM = 2
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl029a
  SUTTERM = 1536
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl029a
  SUTTERM = 1538
  SUTPASS = /w07
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl029a
  SUTTERM = 1538
  SUTPASS = /w08
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl029a
  SUTTERM = 1538
  SUTPASS = /w09
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl029a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0
ENDSUT
STARTSUT

```

```

SUTHOST = cl029a
SUTTERM = 1542
SUTPASS = /w11
SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
  VAL = U11 = 48000
  VAL = U21 = 0
  VAL = U31 = 14400
  VAL = U41 = 0
  VAL = U51 = 12010
  VAL = U61 = 12010
  VAL = U101 = 10020
  VAL = U111 = 5010
  VAL = U121 = 5010
  VAL = U171 = 18010
  VAL = U181 = 3020
  VAL = U191 = 2010
  VAL = U201 = 2010
  VAL = U211 = 2010
  VAL = U131 = 1023
  VAL = U141 = 255
  VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte11.conf
.....
#
# rte11.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
STARTRTE
  RTEHOST = rte11
  STARTSUT
    SUTHOST = cl031a
    SUTTERM = 1538
    SUTPASS = /w00
    SUTLOG = 0
  ENSUT
  STARTSUT
    SUTHOST = cl031a
    SUTTERM = 1538
    SUTPASS = /w01
    SUTLOG = 0
  ENSUT
  STARTSUT
    SUTHOST = cl031a
    SUTTERM = 1538
    SUTPASS = /w02
    SUTLOG = 0
  ENSUT
  STARTSUT
    SUTHOST = cl031a
    SUTTERM = 1538
    SUTPASS = /w03
    SUTLOG = 0
  ENSUT
  STARTSUT
    SUTHOST = cl031a
    SUTTERM = 1538
    SUTPASS = /w04
    SUTLOG = 0
  ENSUT

```

```

STARTSUT
  SUTHOST = cl031a
  SUTTERM = 1538
  SUTPASS = /w05
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl031a
  SUTTERM = 2
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl032a
  SUTTERM = 1536
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl032a
  SUTTERM = 1538
  SUTPASS = /w07
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl032a
  SUTTERM = 1538
  SUTPASS = /w08
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl032a
  SUTTERM = 1538
  SUTPASS = /w09
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl032a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl032a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
  VAL = U11 = 48000
  VAL = U21 = 0
  VAL = U31 = 14400
  VAL = U41 = 0
  VAL = U51 = 12010
  VAL = U61 = 12010
  VAL = U101 = 10020
  VAL = U111 = 5010
  VAL = U121 = 5010
  VAL = U171 = 18010
  VAL = U181 = 3020
  VAL = U191 = 2010
  VAL = U201 = 2010
  VAL = U211 = 2010
  VAL = U131 = 1023
  VAL = U141 = 255
  VAL = U151 = 8191

```

```

ENDVARIABLE
ENDGROUP
.....
RTE/rte12.conf
.....
#
#   rte12.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
STARTRTE
  RTEHOST = rte12
STARTSUT
  SUTHOST = cl034a
  SUTTERM = 1538
  SUTPASS = /w00
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl034a
  SUTTERM = 1538
  SUTPASS = /w01
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl034a
  SUTTERM = 1538
  SUTPASS = /w02
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl034a
  SUTTERM = 1538
  SUTPASS = /w03
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl034a
  SUTTERM = 1538
  SUTPASS = /w04
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl034a
  SUTTERM = 1538
  SUTPASS = /w05
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl034a
  SUTTERM = 2
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl035a
  SUTTERM = 1536
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl035a
  SUTTERM = 1538
  SUTPASS = /w07
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl035a

```

```

SUTTERM = 1538
SUTPASS = /w08
SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl035a
  SUTTERM = 1538
  SUTPASS = /w09
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl035a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl035a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
  VAL = U11 = 48000
  VAL = U21 = 0
  VAL = U31 = 14400
  VAL = U41 = 0
  VAL = U51 = 12010
  VAL = U61 = 12010
  VAL = U101 = 10020
  VAL = U111 = 5010
  VAL = U121 = 5010
  VAL = U171 = 18010
  VAL = U181 = 3020
  VAL = U191 = 2010
  VAL = U201 = 2010
  VAL = U211 = 2010
  VAL = U131 = 1023
  VAL = U141 = 255
  VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte13.conf
.....
#
#   rte13.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
STARTRTE
  RTEHOST = rte13
STARTSUT
  SUTHOST = cl037a
  SUTTERM = 1538
  SUTPASS = /w00
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl037a
  SUTTERM = 1538
  SUTPASS = /w01
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl037a
  SUTTERM = 1538
  SUTPASS = /w02
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl037a
  SUTTERM = 1538
  SUTPASS = /w03
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl037a
  SUTTERM = 1538
  SUTPASS = /w04
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl037a
  SUTTERM = 1538
  SUTPASS = /w05
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl037a
  SUTTERM = 2
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTTERM = 1536
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTTERM = 1538
  SUTPASS = /w07
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTTERM = 1538
  SUTPASS = /w08
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTTERM = 1538
  SUTPASS = /w09
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE

```

```

SUTHOST = cl037a
SUTTERM = 1538
SUTPASS = /w02
SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl037a
  SUTTERM = 1538
  SUTPASS = /w03
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl037a
  SUTTERM = 1538
  SUTPASS = /w04
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl037a
  SUTTERM = 1538
  SUTPASS = /w05
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl037a
  SUTTERM = 2
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTTERM = 1536
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTTERM = 1538
  SUTPASS = /w07
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTTERM = 1538
  SUTPASS = /w08
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTTERM = 1538
  SUTPASS = /w09
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl038a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE

```

```

VAL = U11 = 48000
VAL = U21 = 0
VAL = U31 = 14400
VAL = U41 = 0
VAL = U51 = 12010
VAL = U61 = 12010
VAL = U101 = 10020
VAL = U111 = 5010
VAL = U121 = 5010
VAL = U171 = 18010
VAL = U181 = 3020
VAL = U191 = 2010
VAL = U201 = 2010
VAL = U211 = 2010
VAL = U131 = 1023
VAL = U141 = 255
VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte14.conf
.....
#
# rte14.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
RTEHOST = rte14
STARTSUT
SUTHOST = cl040a
SUTTERM = 1538
SUTPASS = /w00
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl040a
SUTTERM = 1538
SUTPASS = /w01
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl040a
SUTTERM = 1538
SUTPASS = /w02
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl040a
SUTTERM = 1538
SUTPASS = /w03
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl040a
SUTTERM = 1538
SUTPASS = /w04
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl040a
SUTTERM = 1538
SUTPASS = /w05
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl040a

```

```

SUTTERM = 2
SUTPASS = /w06
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl041a
SUTTERM = 1536
SUTPASS = /w06
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl041a
SUTTERM = 1538
SUTPASS = /w07
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl041a
SUTTERM = 1538
SUTPASS = /w08
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl041a
SUTTERM = 1538
SUTPASS = /w09
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl041a
SUTTERM = 1538
SUTPASS = /w10
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl041a
SUTTERM = 1542
SUTPASS = /w11
SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
VAL = U11 = 48000
VAL = U21 = 0
VAL = U31 = 14400
VAL = U41 = 0
VAL = U51 = 12010
VAL = U61 = 12010
VAL = U101 = 10020
VAL = U111 = 5010
VAL = U121 = 5010
VAL = U171 = 18010
VAL = U181 = 3020
VAL = U191 = 2010
VAL = U201 = 2010
VAL = U131 = 1023
VAL = U141 = 255
VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte15.conf
.....
#

```

```

# rte15.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
RTEHOST = rte15
STARTSUT
SUTHOST = cl043a
SUTTERM = 1538
SUTPASS = /w00
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl043a
SUTTERM = 1538
SUTPASS = /w01
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl043a
SUTTERM = 1538
SUTPASS = /w02
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl043a
SUTTERM = 1538
SUTPASS = /w03
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl043a
SUTTERM = 1538
SUTPASS = /w04
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl043a
SUTTERM = 1538
SUTPASS = /w05
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl043a
SUTTERM = 2
SUTPASS = /w06
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl044a
SUTTERM = 1536
SUTPASS = /w06
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl044a
SUTTERM = 1538
SUTPASS = /w07
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl044a
SUTTERM = 1538
SUTPASS = /w08
SUTLOG = 0
ENDSUT
STARTSUT

```

```

SUTHOST = cl044a
SUTTERM = 1538
SUTPASS = /w09
SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl044a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl044a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
  VAL = U11 = 48000
  VAL = U21 = 0
  VAL = U31 = 14400
  VAL = U41 = 0
  VAL = U51 = 12010
  VAL = U61 = 12010
  VAL = U101 = 10020
  VAL = U111 = 5010
  VAL = U121 = 5010
  VAL = U171 = 18010
  VAL = U181 = 3020
  VAL = U191 = 2010
  VAL = U201 = 2010
  VAL = U211 = 2010
  VAL = U131 = 1023
  VAL = U141 = 255
  VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte16.conf
.....
#
#  rte16.conf :configuration file for TPC-C
Rev3.0
#  Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
  STARTRTE
    RTEHOST = rte16
  STARTSUT
    SUTHOST = cl046a
    SUTTERM = 1538
    SUTPASS = /w00
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl046a
    SUTTERM = 1538
    SUTPASS = /w01
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl046a
    SUTTERM = 1538
    SUTPASS = /w02
    SUTLOG = 0
  ENDSUT

```

```

STARTSUT
  SUTHOST = cl046a
  SUTTERM = 1538
  SUTPASS = /w03
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl046a
  SUTTERM = 1538
  SUTPASS = /w04
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl046a
  SUTTERM = 1538
  SUTPASS = /w05
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl046a
  SUTTERM = 2
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl047a
  SUTTERM = 1536
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl047a
  SUTTERM = 1538
  SUTPASS = /w07
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl047a
  SUTTERM = 1538
  SUTPASS = /w08
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl047a
  SUTTERM = 1538
  SUTPASS = /w09
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl047a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl047a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
  VAL = U11 = 48000
  VAL = U21 = 0
  VAL = U31 = 14400
  VAL = U41 = 0
  VAL = U51 = 12010

```

```

VAL = U61 = 12010
VAL = U101 = 10020
VAL = U111 = 5010
VAL = U121 = 5010
VAL = U171 = 18010
VAL = U181 = 3020
VAL = U191 = 2010
VAL = U201 = 2010
VAL = U211 = 2010
VAL = U131 = 1023
VAL = U141 = 255
VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte17.conf
.....
#
#  rte17.conf :configuration file for TPC-C
Rev3.0
#  Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
  STARTRTE
    RTEHOST = rte17
  STARTSUT
    SUTHOST = cl049a
    SUTTERM = 1538
    SUTPASS = /w00
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl049a
    SUTTERM = 1538
    SUTPASS = /w01
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl049a
    SUTTERM = 1538
    SUTPASS = /w02
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl049a
    SUTTERM = 1538
    SUTPASS = /w03
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl049a
    SUTTERM = 1538
    SUTPASS = /w04
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl049a
    SUTTERM = 1538
    SUTPASS = /w05
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl049a
    SUTTERM = 2
    SUTPASS = /w06
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl050a

```

```

SUTTERM = 1536
SUTPASS = /w06
SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl050a
  SUTTERM = 1538
  SUTPASS = /w07
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl050a
  SUTTERM = 1538
  SUTPASS = /w08
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl050a
  SUTTERM = 1538
  SUTPASS = /w09
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl050a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl050a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
  VAL = U1I = 48000
  VAL = U2I = 0
  VAL = U3I = 14400
  VAL = U4I = 0
  VAL = U5I = 12010
  VAL = U6I = 12010
  VAL = U10I = 10020
  VAL = U11I = 5010
  VAL = U12I = 5010
  VAL = U17I = 18010
  VAL = U18I = 3020
  VAL = U19I = 2010
  VAL = U20I = 2010
  VAL = U21I = 2010
  VAL = U13I = 1023
  VAL = U14I = 255
  VAL = U15I = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte18.conf
.....
#
# rte18.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
STARTRTE
  RTEHOST = rte18
STARTSUT

```

```

SUTHOST = cl052a
SUTTERM = 1538
SUTPASS = /w00
SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl052a
  SUTTERM = 1538
  SUTPASS = /w01
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl052a
  SUTTERM = 1538
  SUTPASS = /w02
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl052a
  SUTTERM = 1538
  SUTPASS = /w03
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl052a
  SUTTERM = 1538
  SUTPASS = /w04
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl052a
  SUTTERM = 1538
  SUTPASS = /w05
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl052a
  SUTTERM = 2
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl053a
  SUTTERM = 1536
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl053a
  SUTTERM = 1538
  SUTPASS = /w07
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl053a
  SUTTERM = 1538
  SUTPASS = /w08
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl053a
  SUTTERM = 1538
  SUTPASS = /w09
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl053a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0

```

```

SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl053a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
  VAL = U1I = 48000
  VAL = U2I = 0
  VAL = U3I = 14400
  VAL = U4I = 0
  VAL = U5I = 12010
  VAL = U6I = 12010
  VAL = U10I = 10020
  VAL = U11I = 5010
  VAL = U12I = 5010
  VAL = U17I = 18010
  VAL = U18I = 3020
  VAL = U19I = 2010
  VAL = U20I = 2010
  VAL = U21I = 2010
  VAL = U13I = 1023
  VAL = U14I = 255
  VAL = U15I = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte19.conf
.....
#
# rte19.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
STARTRTE
  RTEHOST = rte19
STARTSUT
  SUTHOST = cl055a
  SUTTERM = 1538
  SUTPASS = /w00
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl055a
  SUTTERM = 1538
  SUTPASS = /w01
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl055a
  SUTTERM = 1538
  SUTPASS = /w02
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl055a
  SUTTERM = 1538
  SUTPASS = /w03
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl055a
  SUTTERM = 1538

```

```

SUTPASS = /w04
SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl055a
    SUTTERM = 1538
    SUTPASS = /w05
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl055a
    SUTTERM = 2
    SUTPASS = /w06
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl056a
    SUTTERM = 1536
    SUTPASS = /w06
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl056a
    SUTTERM = 1538
    SUTPASS = /w07
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl056a
    SUTTERM = 1538
    SUTPASS = /w08
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl056a
    SUTTERM = 1538
    SUTPASS = /w09
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl056a
    SUTTERM = 1538
    SUTPASS = /w10
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl056a
    SUTTERM = 1542
    SUTPASS = /w11
    SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
    VAL = U11 = 48000
    VAL = U21 = 0
    VAL = U31 = 14400
    VAL = U41 = 0
    VAL = U51 = 12010
    VAL = U61 = 12010
    VAL = U101 = 10020
    VAL = U111 = 5010
    VAL = U121 = 5010
    VAL = U171 = 18010
    VAL = U181 = 3020
    VAL = U191 = 2010
    VAL = U201 = 2010
    VAL = U211 = 2010

```

```

VAL = U131 = 1023
VAL = U141 = 255
VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte20.conf
.....
#
#   rte20.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
STARTRTE
    RTEHOST = rte20
    STARTSUT
        SUTHOST = cl058a
        SUTTERM = 1538
        SUTPASS = /w00
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl058a
        SUTTERM = 1538
        SUTPASS = /w01
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl058a
        SUTTERM = 1538
        SUTPASS = /w02
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl058a
        SUTTERM = 1538
        SUTPASS = /w03
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl058a
        SUTTERM = 1538
        SUTPASS = /w04
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl058a
        SUTTERM = 1538
        SUTPASS = /w05
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl058a
        SUTTERM = 2
        SUTPASS = /w06
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl059a
        SUTTERM = 1536
        SUTPASS = /w06
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl059a
        SUTTERM = 1538

```

```

SUTPASS = /w07
SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl059a
    SUTTERM = 1538
    SUTPASS = /w08
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl059a
    SUTTERM = 1538
    SUTPASS = /w09
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl059a
    SUTTERM = 1542
    SUTPASS = /w11
    SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
    VAL = U11 = 48000
    VAL = U21 = 0
    VAL = U31 = 14400
    VAL = U41 = 0
    VAL = U51 = 12010
    VAL = U61 = 12010
    VAL = U101 = 10020
    VAL = U111 = 5010
    VAL = U121 = 5010
    VAL = U171 = 18010
    VAL = U181 = 3020
    VAL = U191 = 2010
    VAL = U201 = 2010
    VAL = U211 = 2010
    VAL = U131 = 1023
    VAL = U141 = 255
    VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte21.conf
.....
#
#   rte21.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
STARTRTE
    RTEHOST = rte21
    STARTSUT
        SUTHOST = cl061a
        SUTTERM = 1538
        SUTPASS = /w00
        SUTLOG = 0
    ENDSUT
    STARTSUT
        SUTHOST = cl061a

```

```

SUTTERM = 1538
SUTPASS = /w01
SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl061a
  SUTTERM = 1538
  SUTPASS = /w02
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl061a
  SUTTERM = 1538
  SUTPASS = /w03
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl061a
  SUTTERM = 1538
  SUTPASS = /w04
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl061a
  SUTTERM = 1538
  SUTPASS = /w05
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl061a
  SUTTERM = 2
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl062a
  SUTTERM = 1536
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl062a
  SUTTERM = 1538
  SUTPASS = /w07
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl062a
  SUTTERM = 1538
  SUTPASS = /w08
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl062a
  SUTTERM = 1538
  SUTPASS = /w09
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl062a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl062a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0

```

```

ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
  VAL = U11 = 48000
  VAL = U21 = 0
  VAL = U31 = 14400
  VAL = U41 = 0
  VAL = U51 = 12010
  VAL = U61 = 12010
  VAL = U101 = 10020
  VAL = U111 = 5010
  VAL = U121 = 5010
  VAL = U171 = 18010
  VAL = U181 = 3020
  VAL = U191 = 2010
  VAL = U201 = 2010
  VAL = U211 = 2010
  VAL = U131 = 1023
  VAL = U141 = 255
  VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte22.conf
.....
#
# rte22.conf :configuration file for TPC-C
Rev3.0
# Author : mkdef -Auto Configurator for R3-
#

STARTGROUP
STARTRTE
  RTEHOST = rte22
  STARTSUT
    SUTHOST = cl064a
    SUTTERM = 1538
    SUTPASS = /w00
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl064a
    SUTTERM = 1538
    SUTPASS = /w01
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl064a
    SUTTERM = 1538
    SUTPASS = /w02
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl064a
    SUTTERM = 1538
    SUTPASS = /w03
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl064a
    SUTTERM = 1538
    SUTPASS = /w04
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl064a

```

```

SUTTERM = 1538
SUTPASS = /w05
SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl064a
  SUTTERM = 2
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl065a
  SUTTERM = 1536
  SUTPASS = /w06
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl065a
  SUTTERM = 1538
  SUTPASS = /w07
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl065a
  SUTTERM = 1538
  SUTPASS = /w08
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl065a
  SUTTERM = 1538
  SUTPASS = /w09
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl065a
  SUTTERM = 1538
  SUTPASS = /w10
  SUTLOG = 0
ENDSUT
STARTSUT
  SUTHOST = cl065a
  SUTTERM = 1542
  SUTPASS = /w11
  SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
  VAL = U11 = 48000
  VAL = U21 = 0
  VAL = U31 = 14400
  VAL = U41 = 0
  VAL = U51 = 12010
  VAL = U61 = 12010
  VAL = U101 = 10020
  VAL = U111 = 5010
  VAL = U121 = 5010
  VAL = U171 = 18010
  VAL = U181 = 3020
  VAL = U191 = 2010
  VAL = U201 = 2010
  VAL = U211 = 2010
  VAL = U131 = 1023
  VAL = U141 = 255
  VAL = U151 = 8191
ENDVARIABLE
ENDGROUP

```

```

.....
RTE/rte23.conf
.....
#
#   rte23.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
  STARTRTE
    RTEHOST = rte23
    STARTSUT
      SUTHOST = cl067a
      SUTTERM = 1538
      SUTPASS = /w00
      SUTLOG = 0
    ENDSUT
    STARTSUT
      SUTHOST = cl067a
      SUTTERM = 1538
      SUTPASS = /w01
      SUTLOG = 0
    ENDSUT
  STARTSUT
    SUTHOST = cl067a
    SUTTERM = 1538
    SUTPASS = /w02
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl067a
    SUTTERM = 1538
    SUTPASS = /w03
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl067a
    SUTTERM = 1538
    SUTPASS = /w04
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl067a
    SUTTERM = 1538
    SUTPASS = /w05
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl067a
    SUTTERM = 2
    SUTPASS = /w06
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl068a
    SUTTERM = 1536
    SUTPASS = /w06
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl068a
    SUTTERM = 1538
    SUTPASS = /w07
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl068a

```

```

      SUTTERM = 1538
      SUTPASS = /w08
      SUTLOG = 0
    ENDSUT
  STARTSUT
    SUTHOST = cl068a
    SUTTERM = 1538
    SUTPASS = /w09
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl068a
    SUTTERM = 1538
    SUTPASS = /w10
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl068a
    SUTTERM = 1542
    SUTPASS = /w11
    SUTLOG = 0
  ENDSUT
  ENDRTE
  PORT = 80
  SYNC = 0
  STARTVARIABLE
    VAL = U11 = 48000
    VAL = U21 = 0
    VAL = U31 = 14400
    VAL = U41 = 0
    VAL = U51 = 12010
    VAL = U61 = 12010
    VAL = U101 = 10020
    VAL = U111 = 5010
    VAL = U121 = 5010
    VAL = U171 = 18010
    VAL = U181 = 3020
    VAL = U191 = 2010
    VAL = U201 = 2010
    VAL = U211 = 2010
    VAL = U131 = 1023
    VAL = U141 = 255
    VAL = U151 = 8191
  ENDVARIABLE
ENDGROUP
.....
RTE/rte24.conf
.....
#
#   rte24.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
  STARTRTE
    RTEHOST = rte24
    STARTSUT
      SUTHOST = cl070a
      SUTTERM = 1538
      SUTPASS = /w00
      SUTLOG = 0
    ENDSUT
  STARTSUT
    SUTHOST = cl070a
    SUTTERM = 1538
    SUTPASS = /w01
    SUTLOG = 0
  ENDSUT
  ENDRTE
  PORT = 80
  SYNC = 0
  STARTVARIABLE

```

```

    SUTHOST = cl070a
    SUTTERM = 1538
    SUTPASS = /w02
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl070a
    SUTTERM = 1538
    SUTPASS = /w03
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl070a
    SUTTERM = 1538
    SUTPASS = /w04
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl070a
    SUTTERM = 2
    SUTPASS = /w06
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl071a
    SUTTERM = 1536
    SUTPASS = /w06
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl071a
    SUTTERM = 1538
    SUTPASS = /w07
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl071a
    SUTTERM = 1538
    SUTPASS = /w08
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl071a
    SUTTERM = 1538
    SUTPASS = /w09
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl071a
    SUTTERM = 1538
    SUTPASS = /w10
    SUTLOG = 0
  ENDSUT
  STARTSUT
    SUTHOST = cl071a
    SUTTERM = 1542
    SUTPASS = /w11
    SUTLOG = 0
  ENDSUT
  ENDRTE
  PORT = 80
  SYNC = 0
  STARTVARIABLE

```



```

        SUTPASS = /w11
        SUTLOG = 0
    ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
    VAL = U11 = 48000
    VAL = U21 = 0
    VAL = U31 = 14400
    VAL = U41 = 0
    VAL = U51 = 12010
    VAL = U61 = 12010
    VAL = U101 = 10020
    VAL = U111 = 5010
    VAL = U121 = 5010
    VAL = U171 = 18010
    VAL = U181 = 3020
    VAL = U191 = 2010
    VAL = U201 = 2010
    VAL = U211 = 2010
    VAL = U131 = 1023
    VAL = U141 = 255
    VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte27.conf
.....
#
#   rte27.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
    STARTRTE
        RTEHOST = rte27
        STARTSUT
            SUTHOST = cl077a
            SUTTERM = 769
            SUTPASS = /w00
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl077a
            SUTTERM = 769
            SUTPASS = /w01
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl077a
            SUTTERM = 769
            SUTPASS = /w02
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl077a
            SUTTERM = 769
            SUTPASS = /w03
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl077a
            SUTTERM = 769
            SUTPASS = /w04
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl077a

```

```

        SUTTERM = 769
        SUTPASS = /w05
        SUTLOG = 0
    ENDSUT
STARTSUT
    SUTHOST = cl077a
    SUTTERM = 769
    SUTPASS = /w06
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl077a
    SUTTERM = 769
    SUTPASS = /w07
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl077a
    SUTTERM = 769
    SUTPASS = /w08
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl077a
    SUTTERM = 769
    SUTPASS = /w09
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl077a
    SUTTERM = 769
    SUTPASS = /w10
    SUTLOG = 0
ENDSUT
STARTSUT
    SUTHOST = cl077a
    SUTTERM = 769
    SUTPASS = /w11
    SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
    VAL = U11 = 48000
    VAL = U21 = 0
    VAL = U31 = 14400
    VAL = U41 = 0
    VAL = U51 = 12010
    VAL = U61 = 12010
    VAL = U101 = 10020
    VAL = U111 = 5010
    VAL = U121 = 5010
    VAL = U171 = 18010
    VAL = U181 = 3020
    VAL = U191 = 2010
    VAL = U201 = 2010
    VAL = U211 = 2010
    VAL = U131 = 1023
    VAL = U141 = 255
    VAL = U151 = 8191
ENDVARIABLE
ENDGROUP
.....
RTE/rte28.conf
.....
#

```

```

#   rte28.conf :configuration file for TPC-C
Rev3.0
#   Author : mkdef -Auto Configurator for R3-
#
STARTGROUP
    STARTRTE
        RTEHOST = rte28
        STARTSUT
            SUTHOST = cl079a
            SUTTERM = 772
            SUTPASS = /w00
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl079a
            SUTTERM = 772
            SUTPASS = /w01
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl079a
            SUTTERM = 772
            SUTPASS = /w02
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl079a
            SUTTERM = 772
            SUTPASS = /w03
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl079a
            SUTTERM = 772
            SUTPASS = /w04
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl079a
            SUTTERM = 772
            SUTPASS = /w05
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl079a
            SUTTERM = 772
            SUTPASS = /w06
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl079a
            SUTTERM = 772
            SUTPASS = /w07
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl079a
            SUTTERM = 772
            SUTPASS = /w08
            SUTLOG = 0
        ENDSUT
        STARTSUT
            SUTHOST = cl079a
            SUTTERM = 772
            SUTPASS = /w09
            SUTLOG = 0
        ENDSUT
        STARTSUT

```

```

SUTHOST = cl079a
SUTTERM = 772
SUTPASS = /w10
SUTLOG = 0
ENDSUT
STARTSUT
SUTHOST = cl079a
SUTTERM = 778
SUTPASS = /w11
SUTLOG = 0
ENDSUT
ENDRTE
PORT = 80
SYNC = 0
STARTVARIABLE
VAL = U11 = 48000
VAL = U21 = 0
VAL = U31 = 14400
VAL = U41 = 0
VAL = U51 = 12010
VAL = U61 = 12010
VAL = U101 = 10020
VAL = U111 = 5010
VAL = U121 = 5010
VAL = U171 = 18010
VAL = U181 = 3020
VAL = U191 = 2010
VAL = U201 = 2010
VAL = U211 = 2010
VAL = U131 = 1023
VAL = U141 = 255
VAL = U151 = 8191
ENDVARIABLE
ENDGROUP

```

Appendix D: System Tunables

```

.....
Server_tunables/etc/system
.....
*ident "@(#)system 1.18 97/06/27 SMI" /*
SVR4 1.5 */
*
* SYSTEM SPECIFICATION FILE
*
* moddir:
*
* Set the search path for modules. This has a
format similar to the
* csh path variable. If the module isn't found in
the first directory
* it tries the second and so on. The default is
/kernel /usr/kernel
*
* Example:
* moddir: /kernel /usr/kernel
/other/modules
*
* root device and root filesystem configuration:
*
* The following may be used to override the
defaults provided by
* the boot program:
*
* rootfs: Set the filesystem type of the root.
*
* rootdev: Set the root device. This
should be a fully
* expanded physical pathname. The
default is the
* physical pathname of the device
where the boot
* program resides. The physical
pathname is
* highly platform and configuration
dependent.
*
* Example:
* rootfs:ufs
*
rootdev:/sbus@1,f800000/esp@0,800000/sd@3
,0:a
*
* (Swap device configuration should be
specified in /etc/vfstab.)
*
* exclude:
*
* Modules appearing in the moddir path which
are NOT to be loaded,
* even if referenced. Note that `exclude'
accepts either a module name,
* or a filename which includes the directory.
*
* Examples:
* exclude: win

```

```

* exclude: sys/shmsys
* forceload:
*
* Cause these modules to be loaded at boot
time, (just before mounting
* the root filesystem) rather than at first
reference. Note that
* forceload expects a filename which includes
the directory. Also
* note that loading a module does not
necessarily imply that it will
* be installed.
*
* Example:
* forceload: drv/foo
*
* set:
*
* Set an integer variable in the kernel or a
module to a new value.
* This facility should be used with caution.
See system(4).
*
* Examples:
*
* To set variables in 'unix':
*
* set nautopush=32
* set maxusers=40
*
* To set a variable named 'debug' in the
module named 'test_module'
*
* set test_module:debug = 0x13
* vxvm_START (do not remove)
forceload: drv/vxdmp
forceload: drv/vxio
forceload: drv/vxspec
set vxio_masio=8192
set vxio_maxkaicount=8192
set voliorem_chunk_size=1048576
set voliorem_kumap_size=15728649
* vxvm_END (do not remove)
forceload: drv/vxdmp
forceload: drv/vxdmp
set vxio:vol_maxio=32768
forceload: drv/vxdmp
set msgsys:msginfo_msgmap=800
set msgsys:msginfo_msgmni=100
set msgsys:msginfo_msgtql=160
set msgsys:msginfo_msgseg=4096
set shmsys:shminfo_shmmax=0x800000000000
set semsys:seminfo_semmmap=400
set semsys:seminfo_semmni=1000
set semsys:seminfo_semmns=6000
set semsys:seminfo_semmnu=1200
set semsys:seminfo_semmnl=1024
set semsys:seminfo_semume=200
set semsys:seminfo_semopm=1024
set sd:sd_max_throttle=32
set autoup = 86400
set tune_t_fsflshr = 86400
set minfree=512
set desfree=1024
set lotsfree=2048
set memscrub_period_sec=172800
set shmsys:shminfo_shmni=512
set shmsys:shminfo_shmseg=512
set seg_pwindow=0

```

```

.....
DB_tunables/init.ora
.....
query_rewrite_enabled = FALSE
_lightweight_hdrs = true
_kghdsidx_count = 1
_db_cache_pre_warm = FALSE
_column_compression_factor = 175
_cursor_cache_frame_bind_memory = TRUE
_db_block_cache_protect = FALSE
_db_writer_flush_imu = false
_imu_pools = 1500
_lgwr_async_io = FALSE
_smu_debug_mode = 1536
_two_pass = FALSE
aq_tm_processes = 0
compatible = 10.1.0.0
control_files = /ora_dev/control_001
cursor_space_for_time = TRUE
db_16k_cache_size = 85000M
db_2k_cache_size = 1500M
db_8k_cache_size = 400M
db_cache_size = 36000M
db_keep_cache_size = 31000M
db_recycle_cache_size = 53000M
shared_pool_size = 6000M
db_block_checking = FALSE
db_block_checksum = FALSE
db_block_size = 4096
db_files = 400
db_name = tpcc
db_writer_processes = 4
disk_asynch_io = TRUE
dml_locks = 1800
fast_start_mtr_target = 0
hash_join_enabled = FALSE
java_pool_size = 0
log_archive_start = FALSE
log_buffer = 33554432
log_checkpoint_interval = 0
log_checkpoint_timeout = 0
log_checkpoints_to_alert = TRUE
open_cursors = 4000
parallel_max_servers = 200
pga_aggregate_target = 0
plsql_optimize_level = 2
processes = 2000
recovery_parallelism = 128
replication_dependency_tracking = FALSE
sessions = 2000
statistics_level = basic
timed_statistics = FALSE
transactions = 2000
undo_management = auto
undo_retention = 1
undo_tablespace = undo_1
.....
Client_tunables/inetInfo_cl001.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\inetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\inetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00
,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00

```

```

"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:34,48,96,4f,53,35,c3,01,10,25,00,00,0
0,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0
1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl002.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00
,53,00,56,00,43,00,00,00,53,00,\

4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:34,48,96,4f,53,35,c3,01,10,25,00,00,0
0,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0
1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl004.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019

```

```

"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00
,53,00,56,00,43,00,00,00,53,00,\

4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000d04
"Last Help"=dword:00000d05
"First Counter"=dword:00000cc4
"First Help"=dword:00000cc5
"Library Validation
Code"=hex:7e,3a,5a,66,7e,34,c3,01,10,25,00,00,0
0,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0
1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl005.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00
,53,00,56,00,43,00,00,00,53,00,\

4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000d04
"Last Help"=dword:00000d05
"First Counter"=dword:00000cc4
"First Help"=dword:00000cc5
"Library Validation
Code"=hex:7e,3a,5a,66,7e,34,c3,01,10,25,00,00,0
0,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0
1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl007.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo]

```

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00
,53,00,56,00,43,00,00,00,53,00,\

4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000d04
"Last Help"=dword:00000d05
"First Counter"=dword:00000cc4
"First Help"=dword:00000cc5
"Library Validation
Code"=hex:7e,3a,5a,66,7e,34,c3,01,10,25,00,00,0
0,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0
1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl008.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00
,53,00,56,00,43,00,00,00,53,00,\

4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000d04
"Last Help"=dword:00000d05
"First Counter"=dword:00000cc4
"First Help"=dword:00000cc5
"Library Validation
Code"=hex:7e,3a,5a,66,7e,34,c3,01,10,25,00,00,0
0,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0
1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl010.reg
.....

```



```

"Library Validation
Code"=hex:7e,3a,5a,66,7e,34,c3,01,10,25,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl019.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:34,48,96,4f,53,35,c3,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl020.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42

```

```

"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:34,48,96,4f,53,35,c3,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl022.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:78,f7,15,2b,f8,5a,c3,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl023.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"

```

```

"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:84,b7,f5,38,f8,5a,c3,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl025.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:38,66,7a,be,f9,5a,c3,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl026.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001

```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:52,ed,94,c6,f9,5a,c3,01,10,25,00,00,0,0,0,0,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl028.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:ba,bb,ba,b5,f9,5a,c3,01,10,25,00,00,0,0,0,0,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl029.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
```

```
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:fc,60,da,a4,f9,5a,c3,01,10,25,00,00,0,0,0,0,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl031.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:4a,87,f3,0b,fb,5a,c3,01,10,25,00,00,0,0,0,0,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl032.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
```

```
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:f0,53,aa,19,fb,5a,c3,01,10,25,00,00,0,0,0,0,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl034.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:66,e4,6c,19,fb,5a,c3,01,10,25,00,00,0,0,0,0,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl035.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
```

```

"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:b0,7a,86,24,fb,5a,c3,01,10,25,00,00,0,0,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl037.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:b4,c7,13,ce,fa,5a,c3,01,10,25,00,00,0,0,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl038.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]

```

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:8e,c7,27,cc,fa,5a,c3,01,10,25,00,00,0,0,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl040.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:d0,22,1c,e0,fa,5a,c3,01,10,25,00,00,0,0,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl041.reg
.....

```

```

Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:58,76,b8,e0,fa,5a,c3,01,10,25,00,00,0,0,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl043.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:8c,70,b3,93,fb,5a,c3,01,10,25,00,00,0,0,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510

```

```

"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl044.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00
,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:84,00,64,a1,fb,5a,c3,01,10,25,00,00,0
0,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0
1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl046.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00
,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:bc,57,2f,a6,fb,5a,c3,01,10,25,00,00,0
0,00,00,00

```

```

"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0
1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl047.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00
,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:12,5a,a8,ac,fb,5a,c3,01,10,25,00,00,0
0,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0
1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl049.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00
,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03

```

```

"Library Validation
Code"=hex:d6,82,e4,6e,fc,5a,c3,01,10,25,00,00,0
0,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0
1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl050.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00
,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:3e,3e,7a,79,fc,5a,c3,01,10,25,00,00,0
0,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0
1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl052.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00
,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentCont
rolSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"

```

```

"Last Counter"=dword:0000b42
"Last Help"=dword:0000b43
"First Counter"=dword:0000b02
"First Help"=dword:0000b03
"Library Validation
Code"=hex:d2,4f,df,7d,fc,5a,c3,01,10,25,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl053.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:0000b42
"Last Help"=dword:0000b43
"First Counter"=dword:0000b02
"First Help"=dword:0000b03
"Library Validation
Code"=hex:be,de,57,8a,fc,5a,c3,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl055.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"

```

```

"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:0000b42
"Last Help"=dword:0000b43
"First Counter"=dword:0000b02
"First Help"=dword:0000b03
"Library Validation
Code"=hex:8e,6b,26,1d,fd,5a,c3,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl056.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:0000b42
"Last Help"=dword:0000b43
"First Counter"=dword:0000b02
"First Help"=dword:0000b03
"Library Validation
Code"=hex:8a,cc,d3,26,fd,5a,c3,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl058.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001

```

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:0000b42
"Last Help"=dword:0000b43
"First Counter"=dword:0000b02
"First Help"=dword:0000b03
"Library Validation
Code"=hex:f2,bc,75,31,fd,5a,c3,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl059.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:0000b42
"Last Help"=dword:0000b43
"First Counter"=dword:0000b02
"First Help"=dword:0000b03
"Library Validation
Code"=hex:5e,93,d2,37,fd,5a,c3,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl061.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff

```

```

"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:6e,3e,52,b1,fd,5a,c3,01,10,25,00,00,0,0,0,0,0,0
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0,1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl062.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:84,68,98,c1,fd,5a,c3,01,10,25,00,00,0,0,0,0,0,0
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0,1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl064.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00

```

```

"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:a0,c2,5c,cd,fd,5a,c3,01,10,25,00,00,0,0,0,0,0,0
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0,1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl065.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:2a,8f,93,d6,fd,5a,c3,01,10,25,00,00,0,0,0,0,0,0
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0,1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl067.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019

```

```

"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:3a,6f,97,3c,fe,5a,c3,01,10,25,00,00,0,0,0,0,0,0
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0,1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl068.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:f8,dd,62,49,fe,5a,c3,01,10,25,00,00,0,0,0,0,0,0
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,0,1
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
.....
Client_tunables\InetInfo_cl070.reg
.....
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]

```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINetInfoPerformanceData"
"Close"="CloseINetInfoPerformanceData"
"Collect"="CollectINetInfoPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:8c,0f,88,52,fe,5a,c3,01,10,25,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
Client_tunables\InetInfo_cl071.reg
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINetInfoPerformanceData"
"Close"="CloseINetInfoPerformanceData"
"Collect"="CollectINetInfoPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:02,18,89,57,fe,5a,c3,01,10,25,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
Client_tunables\InetInfo_cl073.reg
```

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINetInfoPerformanceData"
"Close"="CloseINetInfoPerformanceData"
"Collect"="CollectINetInfoPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:e0,8c,91,12,f2,5a,c3,01,10,25,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
Client_tunables\InetInfo_cl075.reg
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINetInfoPerformanceData"
"Close"="CloseINetInfoPerformanceData"
"Collect"="CollectINetInfoPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:1c,3b,bf,30,f3,5a,c3,01,10,25,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
```

```
Client_tunables\InetInfo_cl077.reg
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINetInfoPerformanceData"
"Close"="CloseINetInfoPerformanceData"
"Collect"="CollectINetInfoPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:18,fa,61,d2,f3,5a,c3,01,10,25,00,00,00,00,00
"WbemAdapFileTime"=hex:00,f0,7a,94,0f,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
Client_tunables\InetInfo_cl079.reg
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff
"PoolThreadLimit"=dword:00000384
"MaxPoolThreads"=dword:0000000f
"DisableMemoryCache"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINetInfoPerformanceData"
"Close"="CloseINetInfoPerformanceData"
"Collect"="CollectINetInfoPerformanceData"
"Last Counter"=dword:00000b42
"Last Help"=dword:00000b43
"First Counter"=dword:00000b02
"First Help"=dword:00000b03
"Library Validation
Code"=hex:bc,1f,b9,7d,f5,5a,c3,01,10,25,00,00,00,00,00
```



```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management]
"ClearPageFileAtShutdown"=dword:00000000
"DisablePagingExecutive"=dword:00000000
"IoPageLockLimit"=dword:00010000
"LargeSystemCache"=dword:00000001
"NonPagedPoolQuota"=dword:00000000
"NonPagedPoolSize"=dword:00000000
"PagedPoolQuota"=dword:00000000
"PagedPoolSize"=dword:00000000
"PagingFiles"=hex(7):43,00,3a,00,5c,00,70,00,61,00,67,00,65,00,66,00,69,00,6c,\
00,65,00,2e,00,73,00,79,00,73,00,20,00,31,00,35,00,33,00,36,00,20,00,33,00,\
30,00,37,00,32,00,00,00,00,00
"SecondLevelDataCache"=dword:00000200
"SystemPages"=dword:00000000
"PhysicalAddressExtension"=dword:00000000
.....
Client_tunables/TpApl001.ini
.....
[TPCC_INFO]
Term_Base="1"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl002.ini
.....
[TPCC_INFO]
Term_Base="9231"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl004.ini
.....
[TPCC_INFO]
Term_Base="18461"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl005.ini
.....
[TPCC_INFO]
Term_Base="27691"
NumWarehouses="48000"
```

```
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl007.ini
.....
[TPCC_INFO]
Term_Base="36921"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl008.ini
.....
[TPCC_INFO]
Term_Base="46151"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl010.ini
.....
[TPCC_INFO]
Term_Base="55381"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl011.ini
.....
[TPCC_INFO]
Term_Base="64611"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl013.ini
.....
[TPCC_INFO]
Term_Base="73841"
```

```
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl014.ini
.....
[TPCC_INFO]
Term_Base="83071"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl016.ini
.....
[TPCC_INFO]
Term_Base="92301"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl017.ini
.....
[TPCC_INFO]
Term_Base="101531"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl019.ini
.....
[TPCC_INFO]
Term_Base="110761"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl020.ini
.....
[TPCC_INFO]
```



```

Client_tunables/TpApl043.ini
.....
[TPCC_INFO]
Term_Base="258441"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl044.ini
.....
[TPCC_INFO]
Term_Base="267671"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl046.ini
.....
[TPCC_INFO]
Term_Base="276901"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl047.ini
.....
[TPCC_INFO]
Term_Base="286131"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl049.ini
.....
[TPCC_INFO]
Term_Base="295361"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"

```

```

.....
Client_tunables/TpApl050.ini
.....
[TPCC_INFO]
Term_Base="304591"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl052.ini
.....
[TPCC_INFO]
Term_Base="313821"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl053.ini
.....
[TPCC_INFO]
Term_Base="323051"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl055.ini
.....
[TPCC_INFO]
Term_Base="332281"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl056.ini
.....
[TPCC_INFO]
Term_Base="341511"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]

```

```

NumDBCntlProcesses="18"
.....
Client_tunables/TpApl058.ini
.....
[TPCC_INFO]
Term_Base="350741"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl059.ini
.....
[TPCC_INFO]
Term_Base="359971"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl061.ini
.....
[TPCC_INFO]
Term_Base="369201"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl062.ini
.....
[TPCC_INFO]
Term_Base="378431"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntlProcesses="18"
.....
Client_tunables/TpApl064.ini
.....
[TPCC_INFO]
Term_Base="387661"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueue="1000"
DeliveryPriority="1"

```

```

[TPCC_PROCESS]
NumDBCntIProcesses="18"
.....
Client_tunables/TpApl065.ini
.....
[TPCC_INFO]
Term_Base="396891"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueus="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntIProcesses="18"
.....
Client_tunables/TpApl067.ini
.....
[TPCC_INFO]
Term_Base="406121"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueus="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntIProcesses="18"
.....
Client_tunables/TpApl068.ini
.....
[TPCC_INFO]
Term_Base="415351"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueus="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntIProcesses="18"
.....
Client_tunables/TpApl070.ini
.....
[TPCC_INFO]
Term_Base="424581"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueus="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntIProcesses="18"
.....
Client_tunables/TpApl071.ini
.....
[TPCC_INFO]
Term_Base="433811"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueus="1000"

```

```

DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntIProcesses="18"
.....
Client_tunables/TpApl073.ini
.....
[TPCC_INFO]
Term_Base="443041"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueus="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntIProcesses="18"
.....
Client_tunables/TpApl075.ini
.....
[TPCC_INFO]
Term_Base="452271"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueus="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntIProcesses="18"
.....
Client_tunables/TpApl077.ini
.....
[TPCC_INFO]
Term_Base="461501"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9230"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueus="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntIProcesses="18"
.....
Client_tunables/TpApl079.ini
.....
[TPCC_INFO]
Term_Base="470731"
NumWarehouses="48000"
MaxUsers="480000"
MaxTerm of Client="9270"
CONTROL_Flag="0"
NumDeliveryThreads="36"
NumDeliveryQueus="1000"
DeliveryPriority="1"
[TPCC_PROCESS]
NumDBCntIProcesses="18"

```

Appendix E: Database Creation Code

```

.....
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 $tpcc_listfiles, does not execute
sql
if expr x$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
$tpcc_sqlplus $tpcc_user_pass <<!
  spool createts_$1.log
  set echo on
  drop tablespace $1 including contents;
  $createsql
  set echo off
  spool off
  exit ;
!
.....
analyze.sh
.....
#!/bin/sh
$tpcc_sqlplus $tpcc_user_pass
@${tpcc_sql_dir}/analyze > $tpcc_log_dir/junk
2>&1
# this one tends to fail if indices aren't made,
which is legal, so
# always exit without error.
exit 0
.....
analyze.sql
.....
spool analyze.log;
set echo on;
connect tpcc/tpcc
execute dbms_stats.GATHER_TABLE_STATS
(OWNNAME=>'TPCC', -
  TABNAME=>'STOK', -
  PARTNAME=>NULL, -
ESTIMATE_PERCENT=>1, -
BLOCK_SAMPLE=>TRUE, -
  METHOD_OPT=>'FOR
ALL COLUMNS SIZE 1', -
  DEGREE=>10, -
GRANULARITY=>'DEFAULT', -
  CASCADE=>TRUE);
execute dbms_stats.GATHER_TABLE_STATS
(OWNNAME=>'TPCC', -
  TABNAME=>'CUST', -
  PARTNAME=>NULL, -
ESTIMATE_PERCENT=>1, -
BLOCK_SAMPLE=>TRUE, -
  METHOD_OPT=>'FOR
ALL COLUMNS SIZE 1', -
  DEGREE=>10, -

```

```

  GRANULARITY=>'DEFAULT', -
    CASCADE=>TRUE);
execute dbms_stats.GATHER_TABLE_STATS
(OWNNAME=>'TPCC', -
  TABNAME=>'ORDR', -
  PARTNAME=>NULL, -
ESTIMATE_PERCENT=>1, -
BLOCK_SAMPLE=>TRUE, -
  METHOD_OPT=>'FOR
ALL COLUMNS SIZE 1', -
  DEGREE=>10, -
GRANULARITY=>'DEFAULT', -
  CASCADE=>TRUE);
execute dbms_stats.GATHER_TABLE_STATS
(OWNNAME=>'TPCC', -
  TABNAME=>'ORDL', -
  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;
:
:
assigntemp.sh
:
:
#!/bin/sh
echo Assigning temporary tablespace to user
tpcc...
${tpcc_sql}plus ${tpcc_dba_user_pass}
@${tpcc_sql_dir}/assigntemp > junk 2>&1
if test $? -ne 0
then
    exit 1;
else
    exit 0;
fi
:
:
assigntemp.sql
:
:
spool assigntemp.log;
set echo on;
alter user tpcc temporary tablespace temp_0;
set echo off;
spool off;
exit ;
:
:
bcexpr.sh
:
:
#!/bin/sh
# send command line to bc
echo "$*" | bc
:
:

```

```

buildcreatedb.sh
:
:
#!/bin/sh
# make sure this is not run accidentally
$tpcc_require tpcc_kit
# If we can do auto undo, add in the code here
undo_sql=
if expr $tpcc_auto_undo = t > /dev/null; then
    echo "Auto undo on."
    i=1
    while test $i -le $tpcc_np; do
        undo_sql=`cat <<!
        $undo_sql
        create undo tablespace undo_${i} datafile
        "${tpcc_disks_location}roll${i}" size
        $tpcc_undo_size reuse blocksize
        ${tpcc_undo_bs};
        !
        `
        if test $i -lt $tpcc_np;
        then
            undo_sql=`cat << !
            ${undo_sql}
            !
            `
        fi
        i=`expr $i + 1`;
        done
    fi
    # same with sysaux
    sysaux_sql=
    if test x$tpcc_use_sysaux = xt; then
        sysaux_sql=`cat <<!
        sysaux datafile "${tpcc_disks_location}aux.df"
        size $tpcc_sysaux_size reuse
        !
        `
    fi
    # system files
    datafile_sql=`
    datafile`
    i=1
    while test $i -le $tpcc_np; do
        datafile_sql=`cat <<!
        $datafile_sql
        "${tpcc_disks_location}system_${i}" size
        $tpcc_system_size reuse
        !
        `
        if test $i -lt $tpcc_np;
        then
            datafile_sql=`cat << !
            ${datafile_sql},
            !
            `
        fi
        i=`expr $i + 1`;
        done
    log_sql=
    i=2
    j=3
    while test $i -le $tpcc_np; do
        k=`expr $j + 1`;
        log_sql=`cat <<!
        $log_sql
        alter database add logfile thread $i group $j
        ("${tpcc_disks_location}log_${i}_1" size
        $tpcc_logfile_size reuse,

```

```

        group $k
        ("${tpcc_disks_location}log_${i}_2" size
        $tpcc_logfile_size reuse;
        alter database enable public thread $i;
        !
        `
        i=`expr $i + 1`;
        j=`expr $j + 2`;
        done
    # save to here
    createdbout=${tpcc_genscripts_dir}/createdb.sql
    rm -f $createdbout
    echo "/" created automatically by ' $0 `date` "' >
    $createdbout
    cat >> $createdbout <<!
    spool createdb.log
    set echo on
    shutdown abort
    startup pfile=p_create.ora nomount
    create database tpcc
    controlfile reuse
    maxinstances $tpcc_np $datafile_sql
    logfile "${tpcc_disks_location}log_1_1" size
    $tpcc_logfile_size reuse,
    "${tpcc_disks_location}log_1_2" size
    $tpcc_logfile_size reuse $sysaux_sql ;
    $log_sql
    $undo_sql
    set echo off
    exit sql.sqlcode
    !
    :
    :
    buildcreateindex.sh
    :
    :
    #!/bin/sh
    #make sure to spool to log directory so we don't
    leave a mess
    cd $tpcc_bench/log
    # hard code table variables for each table- not
    moving this
    # to stepenv because it's a bit verbose and
    doesn't need to
    # be configured.
    # maybe should move these to stepenv eventually.
    . $tpcc_tabledata
    # make sure the kit is running so we don't have
    any accidents
    if $tpcc_require tpcc_kit; then
        exit 1
    fi
    # save to here
    createindexout=${tpcc_genscripts_dir}/createindex_
    ${1}.sql
    rm -f $createindexout
    echo "/" created automatically by $0 `date` "' >
    $createindexout
    echo 'set timing on' >> $createindexout
    # for this version, just do one index at a time.
    #for index in $tpcc_index_list; do
    for index in $1; do
        # make sure we have all params
        if $tpcc_require tpcc_kit tpcc_${index}_imp
        tpcc_${index}_used tpcc_${index}_free
        tpcc_${index}_trans tpcc_${index}_flg
        tpcc_${index}_fl tpcc_${index}_warecol
        tpcc_${index}_cols tpcc_${index}_indices
        tpcc_${index}_indexon; then
            exit 0

```

```

fi
indeximp='tp $index imp`
# this is used in several places, so we create it
here.
indexfree='tp $index free`
indexfree=' $tpcc_notneg pctfree $indexfree`
indexused='tp $index used`
indexused=' $tpcc_notneg pctused $indexused`
indextrans='tp $index trans`
indextrans=' $tpcc_notneg initrans $indextrans`
indexflg='tp $index flg`
indexfl='tp $index fl`
if test x`tp $index autospace` = xt; then
    indexsto=
else
    if test -n "$tpcc_notneg flg
Sindexflg` ` $tpcc_notneg fl $indexfl "; then
    indexsto="freelists $indexfl freelist groups
Sindexflg "
fi
fi
indexindexon='tp $index indexon`
physicalattr='cat <<!
Sindexfree $indexused $indextrans
!
.
echo Creating index $index as $indeximp.
indexindices='tp $index indices`
indexindices=' indexnames $index $indexindices`
parallels=' $tpcc_bcexpr $tpcc_cpu ` 4`
case $indeximp in
    index)
        beginsql='cat <<!
create unique index $index on $indexindexon
($indexindices )
$physicalattr
storage (buffer_pool default $(indexsto))
parallel $parallels
tablespace ${index}_0 ;
!
.
;;
##### index end #####
partindex)
    indexpara=' $tpcc_bcexpr $tpcc_cpu ` 2`
    if test $tpcc_np -gt 1 -a "x$index" = "xiordr2" ;
then
        partsql='createindexpart_rac_iordr2`
        else
            partsql='createindexpart $index`
        fi
        if expr $? = 1 > /dev/null; then
            echo index $index may not be partitioned.
            exit 1
        fi
        beginsql='cat <<!
create unique index $index on $indexindexon
($indexindices )
$partsql
parallel $parallels
$physicalattr
storage (buffer_pool default $indexsto )
tablespace ${index}_0 ;
!
.
##### partindex end
#####
;;
none)

```

```

beginsql=
;;
*) echo Invalid index implementation for
$index : $indeximp
exit 1
;;
esac
if test $indeximp != none; then
cat >> $createindexout <<!
set sqlblanklines on
spool createindex_${index}.log ;
set echo on ;
drop index $index ;
    $beginsql
set echo off
spool off
exit sql.sqlcode;
!
else
#create empty shell
cat >> $createindexout <<!
exit 0;
!
fi
done
exit $?
:::
buildcreatetable.sh
:::
#!/bin/sh
# hard code table variables for each table- not
moving this
# to stepenv because it's a bit verbose and
doesn't need to
# be configured.
# maybe should move these to stepenv eventually.
. $tpcc_tabledata
initcluster(){
    #common cluster init for clusters and queue.
    if $tpcc_require tpcc_${table}_clustcols
tpcc_${table}_hkey tpcc_${table}_hash
tpcc_${table}_size; then
        exit 1
    fi
    clusterkeys='tp $table hkey`
    clusterhash='tp $table hash`
    clustersize='tp $table rsize`
    blocksizes='tp $table bs`
    if test x$blocksizes = xauto ; then
        blocksizes=$tpcc_defbs
    else
        blocksizes=' $tpcc_tokilobytes $blocksizes`
    fi
    # if auto, leave the cluster sql out.
    if test "x$clustersize" = "xauto"; then
        if test "x$table" = "xware" ; then
            clustersizesql="size ` $tpcc_bcexpr
\($blocksizes ` 1024 \) - 600"
        else
            if test "x$table" = "xdist" ; then
                clustersizesql="size ` $tpcc_bcexpr
\($blocksizes ` 1024 \) - 600"
            else
                clustersizesql=""
            fi
        fi
    else
        clustersizesql="size `tp $table rsize"

```

```

fi
}
# make sure the kit is running so we don't have
any accidents
if $tpcc_require tpcc_kit; then
    exit 1
fi
# save to here
createtableout=${tpcc_genscripts_dir}/createtable
_${1}.sql
rm -f $createtableout
echo `"/` created automatically by $0 `date` `"/>
$createtableout
echo `set timing on` >> $createtableout
# for this version, just do one index at a time.
#for index in $tpcc_table_list; do
for table in $1; do
# make sure we have all params
if $tpcc_require tpcc_kit tpcc_${table}_imp
tpcc_${table}_used tpcc_${table}_free
tpcc_${table}_trans tpcc_${table}_flg
tpcc_${table}_fl tpcc_${table}_warecol
tpcc_${table}_cols tpcc_${table}_indices; then
    exit 1
fi
tableimp='tp $table imp`
blocksize='tp $table bs`
if test x$blocksize = xauto ; then
    blocksize=$tpcc_defbs
else
    blocksize=' $tpcc_tokilobytes $blocksize`
fi
# this is used in several places, so we create it
here.
tablefree='tp $table free`
tablefree=' $tpcc_notneg pctfree $tablefree`
tableused='tp $table used`
tableused=' $tpcc_notneg pctused $tableused`
tabletrans='tp $table trans`
tabletrans=' $tpcc_notneg initrans $tabletrans`
tableflg='tp $table flg`
tablefl='tp $table fl`
if test x`tp $table autospace` = xt; then
    tablesto=
else
    if test -n "$tpcc_notneg flg
$tableflg` ` $tpcc_notneg fl $tablefl "; then
    tablesto="freelists $tablefl freelist groups
$tableflg "
fi
fi
physicalattr="$tablefree $tableused $tabletrans"
if test -n "`tp $table rac_storage" -a $tpcc_np -gt
1 > /dev/null ; then
    tpcc_buildtable_rac=$table
    export tpcc_buildtable_rac
    Nextent=' $tpcc_scripts/buildcreatets.sh`
    RowPerBlock='tp $table
$(blocksizes)_RowPerBlock`
    RowFactor='tp $table row_factor`
    Nrows=' $tpcc_bcexpr $tpcc_scale `
$RowFactor
    max_extsize=2096000
    nextsize=' $tpcc_bcexpr $Nrows ` $blocksizes /
$RowPerBlock / $Nextent`
    initsize=' $tpcc_bcexpr $nextsize + $blocksizes`
    nearboundary=$initsize
    while test `expr $nearboundary % 5` -ne 0 ; do

```

```

nearboundary=`expr $nearboundary +
$blocksize`
done
freelists=`$tpcc_bcexpr \($nearboundary -
$(nitsize) / $blocksize`
if test $freelists -ne 0 ; then
freelistssql="freelist groups $freelists"
else
freelistssql=""
fi
if test -z "`tp $table bpool`"; then
physicalsto="storage ( initial $(nitsize)k next
$(nextsize)k pctincrease 0 $freelistssql
buffer_pool default )"
else
physicalsto="storage ( initial $(nitsize)k next
$(nextsize)k pctincrease 0 $freelistssql
buffer_pool `tp $table bpool `)"
fi
else
if test -z "`tp $table bpool`"; then
physicalsto="storage ( buffer_pool default
$(tablesto) )"
else
physicalsto="storage ( buffer_pool `tp $table
bpool `$(tablesto) )"
fi
fi
echo Creating table $table as $tableimp.
tablecomp=`tp $table compress`
tablecols=`colize $table all f f $tablecomp`
case $tableimp in
table)
beginsql=`cat <<!
drop table $table ;
create table $table (
$tablecols
)
$physicalattr
$physicalsto
tablespace $(table)_0 ;
!
`
;;
##### table end #####
cluster)
initcluster
clusterusecols=`tp $table clustcols`
clustercols=`colize $table $clusterusecols f`
clusternames=`colize $table $clusterusecols t`
beginsql=`cat <<!
drop cluster $(table)cluster including tables ;
create cluster $(table)cluster (
$clustercols
)
single table
hashkeys $clusterkeys
hash is ( $clusterhash )
$clustersizesql
$physicalattr
$physicalsto
tablespace $(table)_0;
create table $table (
$tablecols
)
cluster $(table)cluster (
$clusternames
);
!
`
fi

```

```

##### cluster end
#####
;;
queue)
usecluster=`tp $table usecluster`
if test -n "$usecluster"; then
othercluster=`tp $table othercluster`
if expr x`tp $othercluster imp` != xqueue >
/dev/null; then
echo table $table shares queue with
$othercluster - $othercluster must be queue.
exit 1
fi
beginsql=`cat <<!
create table $table (
$tablecols
$usecluster ;
!
`
else #not using another cluster
#note- do we need physical attr in here?
queuesortcols=`tp $table queuesort`
if expr x$queuesortcols = x > /dev/null; then
echo table $table may not be a queue- missing
queuesortcols.
exit 1
fi
queuenames=`tp $table queuenames`
if expr x$queuenames = x > /dev/null; then
echo table $table may not be a queue- missing
queuenames.
exit 1
fi
initcluster
#NOTE THAT THIS CHANGES TABLE
COLUMNS FOR THE REST OF THE SCRIPT!
#Thankfully tablecols are already set in, so we can
do this.
if test -n "`tp $table qcols`"; then
eval "tpcc_$(table)_cols=\$tpcc_$(table)_qcols"
fi
queuecols=`indexnames $table $queuesortcols`
queuenamecol=`indexnames $table
$queuenames`
#may want to put single table in certain cases
here
beginsql=`cat <<!
drop cluster $(table)cluster_queue including
tables ;
create cluster $(table)cluster_queue (
$queuecols
)
hashkeys $clusterkeys
hash is ( $clusterhash )
$clustersizesql
tablespace $(table)_0;
create table $table (
$tablecols
, constraint $(table)_uk primary key
( $(queuenamecol) )
)
cluster $(table)cluster_queue (
$queuenamecol
);
!
`
fi

```

```

##### queue end
#####
;;
parttable)
partsql=`createpart $table`
if expr $? = 1 > /dev/null; then
echo table $table may not be partitioned.
exit 1
fi
beginsql=`cat <<!
drop table $table ;
create table $table (
$tablecols
)
$partsql
$physicalattr
$physicalsto
;
!
`
##### parttable end
#####
;;
iot)
iotindices=`tp $table indices`
tableind=`indexnames $table $iotindices`
if test -z "$tableind" > /dev/null; then
echo table $table does not have an index, so
it may not be an iot.
exit 1
fi
beginsql=`cat <<!
drop table $table ;
create table $table (
$tablecols
, constraint i$table primary key ( $tableind )
) organization index
$physicalattr
$physicalsto
tablespace $(table)_0 ;
!
`
##### iot end
#####
;;
partiot)
partsql=`createpart $table`
if expr $? = 1 > /dev/null; then
echo table $table may not be partitioned.
exit 1
fi
iotindices=`tp $table indices`
tableind=`indexnames $table $iotindices`
if expr x$tableind = x > /dev/null; then
echo table $table does not have an index, so
it may not be an partitioned iot.
exit 1
fi
beginsql=`cat <<!
drop table $table ;
create table $table (
$tablecols
, constraint i$table primary key ( $tableind )
) organization index
$partsql
$physicalattr
$physicalsto
storage ( buffer_pool default $tablesto )
!
`

```

```

##### partiot end
#####
;;
none)
;;
*) echo Invalid table implementation for
$stable : $stableimp
exit 1
;;
esac
#log to output file
if test $stableimp != none; then
cat >> $createtableout <<!
set sqlblanklines on
spool createtable_$stable.log
set echo on
$beginsql
set echo off
spool off
exit sql.sqlcode;
!
else
#create empty shell
cat >> $createtableout <<!
exit 0;
!
fi
if expr $? != 0 > /dev/null; then
echo table $stable failed creation.
exit 1
fi
done
:
:
buildcreatets.sh
:
:
#!/bin/sh
if test $tpcc_np -gt 1 -a ` $tpcc_bcexpr
$tpcc_scale % $tpcc_ldrive ` != 0 > /dev/null ; then
echo !!!Warning!!!! number of warehouses:
$tpcc_scale is not multiple of number of logical
drives: $tpcc_ldrive
echo !!!Warning!!!! This may result in not finding
good datafiles size for cust and stok
exit 0
fi
#make sure to spool to log directory so we don't
leave a mess
if expr x$tpcc_listfiles != xt > /dev/null; then
cd $tpcc_log_dir
fi
#save to here
if test -z "$tpcc_buildtable_rac" > /dev/null; then
createtsout=$(tpcc_genscripts_dir)/createts.sh
rm -f $createtsout
echo \#created automatically by $0 `date` >
$createtsout
if test $tpcc_np -gt 1 ; then
echo set -a >> $createtsout
fi
fi
#tp- get table param. (that is,
$tpcc_table_name_$tableparam)
tp(){
eval echo \${tpcc}_$1_$2
}
#numpart- given imp, parts, return 1 if not
partitioned, parts otherwise
numpart(){

```

```

case $1 in
parttable) echo $2 ;;
partindex) echo $2 ;;
partiot) echo $2 ;;
*) echo 1 ;;
esac
}
# we need the tabledata stuff to calculate
automatic base size
.$tpcc_tabledata
# for 'calc' sizes, use hash cluster values to
determine size
for table in $tpcc_table_list $tpcc_index_list temp;
do
if test -n "$tpcc_buildtable_rac" > /dev/null; then
if expr $tpcc_buildtable_rac != $table >
/dev/null; then
continue
fi
fi
tssize=`tp $stable size`
eval "tpcc_${table}_estsize="
if test x$tssize = xcalc ; then
tabhkey=`tp $stable hkey`
# 12/2/2002 - alex.ni, added crsize to
tabledata
tabsize=`tp $stable crsize`
if test x$tabsize = x; then
tabsize=`tp $stable rsize`
fi
#4/8/03 - alex.ni added def bs for auto
if test x$tabsize = xauto; then
tabsize=`$tpcc_bcexpr $tpcc_defbs ` 1024`
fi
if test x$tabhkey != xauto -a x$tabsize !=
xauto; then
etssize=`$tpcc_bcexpr \(` $tabhkey `
$tabsize `) / 1024`
eval "tpcc_${table}_estsize=$etssize"
# echo Table $stable est size ` $tpcc_estsize
$etssize `
fi
fi
if test -n "$tpcc_buildtable_rac" > /dev/null; then
if expr $tpcc_buildtable_rac = $table >
/dev/null; then
break
fi
fi
done
# combine ordl and ordr tablespaces if they
# are stored in a queue cluster
if expr $tpcc_ordr_imp = queue > /dev/null; then
tpcc_ordl_imp=none
tpcc_ordr_size=`$tpcc_bcexpr $tpcc_ordr_size +
$tpcc_ordl_size`
tpcc_ordr_growth=`$tpcc_bcexpr
$tpcc_ordr_growth + $tpcc_ordl_growth`
fi
#count total number of files, needed for createts
totalfiles=0
#figure out max for temp- if no indices (or really
small) use min index size
curmaxindexsize=`$tpcc_bcexpr
$tpcc_tempts_min / 2`
#note temp must come last since its size is based
on the indices
for table in $tpcc_table_list $tpcc_index_list temp;
do

```

```

blocksize=`tp $stable bs`
if test x$blocksize = xauto ; then
blocksize=`$tpcc_defbs
else
blocksize=`$tpcc_tokilobytes $blocksize`
fi
if test -n "$tpcc_buildtable_rac" > /dev/null; then
if expr $tpcc_buildtable_rac != $stable >
/dev/null; then
continue
fi
fi
# make sure we have all params
if $tpcc_require tpcc_kit tpcc_${table}_imp
tpcc_${table}_size tpcc_${table}_nf
tpcc_${table}_ext tpcc_${table}_growth
tpcc_${table}_bs; then
exit 0
fi
tableimp=`tp $stable imp`
if test $tableimp != none > /dev/null; then
# echo No tablespace created for $stable.
# else
# echo Creating tablespace for: $stable
# 1 - figure out the size of the tablespace, and the
extent growth type
growthtype=`tp $stable growth`
# first, get base size from params
# if we have an approximation already, use it
estsize=`tp $stable estsize`
if test x$estsize != x; then
tssize=$estsize
else
# scale up base size (for all sizes that grow)
# cases based on runlength, hist ordr, nordr, ordl
iordr1, iordr2, iordl, inordr, # and cases that do not
scale at all (item, item):
tssize=`tp $stable size`
if test "x$stable" = "xnord" ; then
tssize=`$tpcc_bcexpr 130 ` ` $blocksize `
fi
tssize=`$tpcc_tokilobytes $tssize`
if test $growthtype != 0; then
tssize=`$tpcc_bcexpr $tssize ` ` $tpcc_scale `
fi
fi
growthtype=`tp $stable growth`
if expr $growthtype != regular > /dev/null; then
# echo Growable table
#we want a growable extent size (that is, a
reasonable size) except for item/item/temp
if expr $growthtype != 0 > /dev/null; then
tssize=`$tpcc_bcexpr $tssize +
\(\(` $tpcc_scale ` ` 13 ` ` $tpcc_runlen `
$growthtype `) / 1000 `)
extentgrowth=t
else
extentgrowth=f
fi
else
# echo Fixed, scaled table
extentgrowth=f
fi
# 25% miscellany overhead
tssize=`$tpcc_bcexpr \(` $tssize ` ` 125 `) / 100`
# Another 10% miscellany overhead for fixed
tables
# actually, not sure if this is the problem, so just
use a large

```

```

# regular overhead.
# if test x$extentgrowth != xt > /dev/null; then
#   tssize=`$tpcc_bcepr \($tssize \* 110\) /
100`
# fi
if test -n "`tp $table rac_storage`" -a $tpcc_np -
gt 1 > /dev/null; then
  RowPerBlock=`tp $table
${blocksize}_RowPerBlock`
  RowFactor=`tp $table row_factor`
  Nrows=`$tpcc_bcepr $tpcc_scale \`
$RowFactor`
  tssize=`$tpcc_bcepr $Nrows \` $blocksize /
$RowPerBlock`
  fi
  # save maximum if type is index
  if test $tableimp = index -o $tableimp =
partindex > /dev/null; then
    if $tpcc_isneg `'$tpcc_bcepr
$curmaxindexsize - $tssize`; then
      curmaxindexsize=$tssize
    fi
  fi
  # set if temp
  # echo cur max is $curmaxindexsize
  if test $table = temp > /dev/null; then
    tssize=`$tpcc_bcepr $curmaxindexsize \` 2`
  fi
  # echo Base size after scale: $tssize k
# 2 - Calculate the number of files, and the size of
each file.
  numfiles=`tp $table nf`
  if test -z "`tp $table tsfileinc`"; then
    # default
    fileincrement=`'$tpcc_lcm $tpcc_np
$tpcc_ldrive`
  # fileincrement=$tpcc_ldrive
  else
    # regardless of number of drives and partitions,
just try to
    # keep the number of files down, or a fixed
amount.
    fileincrement=`tp $table tsfileinc`
  fi
  if expr x$numfiles = xcalc \ $x$numfiles = x >
/dev/null; then
    # start with the lowest possible number of files,
and go up
    numfiles=$fileincrement
  fi
  # calculate our max table filesize and max
extent size
  tablebs=`tp $table bs`
  if expr x$tablebs = xauto > /dev/null; then
    bskilo=`'$tpcc_defbs`
  else
    bskilo=`'$tpcc_tokilobytes $tablebs`
  fi
  # tpcc_fsze_limit_k is for 2k, so divide bskilo/2
and multiply to
  # find the limit for us
  max_fsze=`'$tpcc_bcepr \($bskilo \`
$tpcc_fsze_limit_k\) / 2`
  # do the same with the extent size
  max_extsize=`'$tpcc_bcepr \($bskilo \`
$tpcc_extent_limit_k\) / 2`
  # keep under maximum filesize
  if $tpcc_isneg $max_fsze - \($tssize /
$numfiles); then

```

```

# not enough files- make an estimate for how
many files we need
  numfiles=`'$tpcc_bcepr $numfiles +
\(\($tssize / $max_fsze\) / $fileincrement\) \`
$fileincrement`
  fi
  while $tpcc_isneg $max_fsze - \($tssize /
$numfiles); do
    numfiles=`'$tpcc_bcepr $numfiles +
$fileincrement`
  done
  if test -n "`tp $table rac_storage`" -a $tpcc_np -
gt 1 > /dev/null; then
    RowFactor=`tp $table row_factor`
    Nrows=`'$tpcc_bcepr $tpcc_scale \`
$RowFactor`
    RowPerBlock=`tp $table
${blocksize}_RowPerBlock`
    if test `'$tpcc_bcepr $Nrows %
$RowPerBlock` != 0; then
      echo Warning!!! In table $table, total
number of rows: $Nrows is not
      echo divisible by RowPerBlock:
$RowPerBlock
    fi
    while test `'$tpcc_bcepr \($Nrows \`
$blocksize\) % \($RowPerBlock \` $numfiles\)` !=
0; do
      numfiles=`'$tpcc_bcepr $numfiles +
$fileincrement`
    done
    max_extsize=2096000
    Nextent=`$numfiles
nextsize=`'$tpcc_bcepr $Nrows \` $blocksize /
$RowPerBlock / $Nextent`
    while expr `'$tpcc_bcepr \($Nrows \`
$blocksize\) % \($RowPerBlock \` $Nextent\)` !=
0 \ $nextsize > $max_extsize > /dev/null; do
      Nextent=`expr $Nextent + $numfiles`
      nextsize=`'$tpcc_bcepr $Nrows \` $blocksize
/ $RowPerBlock / $Nextent`
    done
    filesize=`'$tpcc_bcepr $tssize / $numfiles`
    filesize=`'$tpcc_bcepr $filesize + \($nextsize /
2)`
    if expr $filesize > $max_fsze > /dev/null; then
      filesize=$max_fsze
    fi
    numfiles=`expr $numfiles + 1`
    tssize=`expr $tssize + $filesize`
  else
    filesize=`'$tpcc_bcepr $tssize / $numfiles`
  fi
  if test -n "`$tpcc_buildtable_rac`" > /dev/null; then
    echo $Nextent
    if expr $tpcc_buildtable_rac = $table >
/dev/null; then
      break
    fi
  fi
  if $tpcc_isneg `'$tpcc_bcepr $filesize - 1024`;
then
    filesize=`'$tpcc_bcepr $filesize + 10`
  else
    filesize=`'$tpcc_bcepr $filesize + 5120`
  fi
  # we want a neat number, though, so round up
  cleanfsze=`'$tpcc_estsize $filesize`
  filesize=`'$tpcc_tokilobytes $cleanfsze`

```

```

# echo got filesize as $filesize

# make files at least 10m each, otherwise may
hit minimum limit.
  if $tpcc_isneg `'$tpcc_bcepr $filesize - 10240`;
then
    filesize=10240;
    cleanfsze=`10M`;
  fi
# 3 - Calculate the extent size of each file
# subtract 1/10 or 1 meg from filesize to give
some extent overhead
  extentsub=`'$tpcc_bcepr $filesize / 10`
  # if subtracting more than a meg, just subtract a
meg, since we don't need more than that
overhead.
  if $tpcc_isneg `'$tpcc_bcepr $extentsub -
1024`; then
    extfsze=`'$tpcc_bcepr $filesize - \($filesize /
10\)
  else
    extfsze=`'$tpcc_bcepr $filesize - 1024`
  fi
  # for indices, we need to make as many extents
as index creation threads (64 * cpu - used to be 4
times cpu, but usually we need more extents for a
higher degree of parallelism.)
  # 4/14/03 - alex.ni, check for size so we don't
make too small extents
  if test $tableimp = index -o $tableimp =
partindex > /dev/null; then
    if test $tpcc_scale -lt 50 -o $extfsze -lt 32000;
then
      numextents=`'$tpcc_bcepr $tpcc_cpu \` 4`
    else
      numextents=`'$tpcc_bcepr $tpcc_cpu \` 64`
    fi
  else
    numextents=1
  fi
  if expr $extentgrowth = t > /dev/null; then
    # for growable tables we want smaller extent
size, around 100M, so they're quicker to allocate.
    max_extsize=102400
  fi
  if test $table = temp > /dev/null; then
    max_extsize=204800
  fi
  # keep under maximum desired extent size
  while $tpcc_isneg $max_extsize - \($extfsze /
$numextents); do
    numextents=`'$tpcc_bcepr $numextents + 1`
  done
  if test $table = nord > /dev/null; then
    if test $numextents = 1 > /dev/null; then
      numextents=2
    fi
  fi
  if test "x$table" = "xcust" -o "x$table" = "xstok" >
/dev/null; then
    lastnumextents=$numextents
    while $tpcc_isneg 102400 - \($extfsze /
$numextents); do
      if $tpcc_isneg 5000 - \($numextents \`
$numfiles); then
        break
      fi
      lastnumextents=$numextents
      numextents=`'$tpcc_bcepr $numextents + 1`

```

```

done
numextents=$lastnumextents
fi
extentsize=`$tpcc_bcexpr $extfsize /
$numextents`;
# 4/14/03 - alex.ni, changed 1/10 to 1/20 since
1/10 wasted a lot of space
# subtract 1/20 or 1 meg extent size for more
overhead
extentsub=`$tpcc_bcexpr $extentsize / 20`
# if subtracting more than a meg, just subtract a
meg, since we don't need more than that
overhead.
if $tpcc_isneg ` $tpcc_bcexpr $extentsub -
1024 `; then
extentsize=`$tpcc_bcexpr $extentsize -
\($extentsub \)`
else
extentsize=`$tpcc_bcexpr $extentsize - 1024`
fi
# round down extent size so it looks neater
# this might not be that great an idea for extents
because they can be small - 10/8 alex.ni
#cleanextsize=`$tpcc_estsize $extentsize`
cleanextsize=$((extentsize)K)
# echo Table $table tssize $tssize k, files
$numfiles, file size $filesize k, ext size $extentsize
k
roundtssize=`$tpcc_fromkilobytes $tssize`
# 4- Add call createts to log file
tablebs=`tp $table bs`
# don't run tokilo on tablebs, since it can be
'auto'
autospace=`tp $table autospace`
# only give partitions if we are using a
partitioned table,
# otherwise tablespaces will be created that will
be unused.
tableparts=`numpart $tableimp $tpcc_np`
if test -n "`tp $table rac_storage" -a $tpcc_np -
gt 1 > /dev/null ; then
autospace=d
fi
if test $tpcc_np -gt 1 ; then
cat >> $createtsout <<!
# Tablespace $table, ts size $roundtssize
(($tssize)K)
# each file $cleanfsize (($filesize)K)
# extents $cleanextsize (($extentsize)K)
# $numfiles files
rac_count=$(( $tpcc_createts $table $numfiles
$tableparts \
$cleanfsize $cleanextsize $tpcc_os `expr
$table = temp` \
$totalfiles $tpcc_cpu $tablebs $autospace`
if expr ` $? != 0 > /dev/null; then
echo Creating tablespace for $table failed.
Exiting.
exit 0
fi
!
else
cat >> $createtsout <<!
# Tablespace $table, ts size $roundtssize
(($tssize)K)
# each file $cleanfsize (($filesize)K)
# extents $cleanextsize (($extentsize)K)
# $numfiles files
\ $tpcc_createts $table $numfiles $tableparts \

```

```

$cleanfsize $cleanextsize $tpcc_os `expr
$table = temp` \
$totalfiles $tpcc_cpu $tablebs $autospace
if expr ` $? != 0 > /dev/null; then
echo Creating tablespace for $table failed.
Exiting.
exit 0
fi
!
fi
totalfiles=`expr $totalfiles + $numfiles`
fi #imp != null
done
if test -z "$tpcc_buildtable_rac" > /dev/null -a
$tpcc_np -gt 1 ; then
cat >> $createtsout <<!
rac_count=$(( $tpcc_createts restbl 20 20 110M
10M $tpcc_os 0 $totalfiles $tpcc_cpu auto t`
if expr ` $? != 0 > /dev/null; then
echo Creating tablespace for restbl failed.
Exiting.
exit 0
fi
!
if test $tpcc_np -gt 1 ; then
/bin/sh $tpcc_scripts/buildcreatets_rac.sh
fi
fi
exit 0
:
buildcreatets_rac.sh
:
#!/bin/sh
#. stepenv.sh
set -a
tpcc_createts_print=t
tpcc_rac_createts_phase=1
i=1
while test $i -le $tpcc_np ; do
tpcc_rac_node=$i
rac_count=0
createtsout=${tpcc_genscripts_dir}/createts_node
${tpcc_rac_node}.sh
rm -f $createtsout
tpcc_rac_createts_phase=1
/bin/sh $tpcc_genscripts_dir/createts.sh
echo wait >> $createtsout
rac_count=0
tpcc_rac_createts_phase=2
/bin/sh $tpcc_genscripts_dir/createts.sh
echo wait >> $createtsout
i=`expr $i + 1`
done
tpcc_rac_create_phase=0
tpcc_rac_node=0
tpcc_createts_print=f
set +a
:
buildfixoo.sh
:
#!/bin/sh
if test $tpcc_ordl_imp = queue; then
${tpcc_scripts}/evenload.sh fixordrordl
$tpcc_scale b e o f '$tpcc_updateordrordl'
else
echo ' ' > $tpcc_fixordrordl
fi

```

```

:
buildloadcust.sh
:
#!/bin/sh
${tpcc_scripts}/evenload.sh cust $tpcc_scale b e c
:
buildloaddist.sh
:
#!/bin/sh
echo `cd $tpcc_bench` >
$tpcc_genscripts_dir/loaddist.sh
echo "`$tpcc_load -M $tpcc_scale -d >
loaddist.log 2>&1" >>
$tpcc_genscripts_dir/loaddist.sh
exit $?
:
buildloadhist.sh
:
#!/bin/sh
${tpcc_scripts}/evenload.sh hist $tpcc_scale b e h
exit $?
:
buildloaditem.sh
:
#!/bin/sh
echo `cd $tpcc_bench` >
$tpcc_genscripts_dir/loaditem.sh
echo "`$tpcc_load -M $tpcc_scale -i >
loaditem.log 2>&1" >>
$tpcc_genscripts_dir/loaditem.sh
exit $?
:
buildloadnord.sh
:
#!/bin/sh
if test $tpcc_np -gt 1 ; then
set -a
tpcc_rac_load=t
i=1
while test $i -le $tpcc_np ; do
tpcc_rac_node=$i
${tpcc_scripts}/evenload.sh nord $tpcc_scale b
e n
i=`expr $i + 1`
done
tpcc_rac_node=0
tpcc_rac_load=f
set +a
else
${tpcc_scripts}/evenload.sh nord $tpcc_scale b
e n
fi
:
buildloadordrordl.sh
:
#!/bin/sh
if test $tpcc_np -gt 1 ; then
set -a
tpcc_rac_load=t
i=1
while test $i -le $tpcc_np ; do
tpcc_rac_node=$i
${tpcc_scripts}/evenload.sh ordrordl
$tpcc_scale b e o t
i=`expr $i + 1`
done
tpcc_rac_node=0

```



```

rem spool cs_tpcc.log
connect tpcc/tpcc;
set echo on
DROP TABLE tpcc_run_desc;
DROP TABLE tpcc_run_int;
DROP TABLE bench_run_int;
DROP TABLE tpcc_back_res;
DROP TABLE tpcc_user_res;
DROP TABLE bench_user_res;
DROP TABLE tpcc_tpm;
DROP TABLE tpcc_new_res;
DROP TABLE bench_new_res;
DROP TABLE tpcc_pay_res;
DROP TABLE bench_pay_res;
DROP TABLE tpcc_ord_res;
DROP TABLE bench_ord_res;
DROP TABLE tpcc_del_res;
DROP TABLE bench_del_res;
DROP TABLE tpcc_sto_res;
DROP TABLE bench_sto_res;
rem
rem description of a run
rem
CREATE TABLE tpcc_run_desc
(
run_name VARCHAR2(20),
rundate DATE,
time NUMBER,
rampup NUMBER,
rampdown NUMBER,
warehouses NUMBER,
customers NUMBER,
users NUMBER,
driver VARCHAR2(40),
commnt VARCHAR2(80)
);
rem
rem throughput of new order transactions
rem
CREATE TABLE tpcc_run_int
(
run_name VARCHAR2(20),
interval NUMBER,
interval_count NUMBER,
response_time NUMBER,
think_time NUMBER
);
rem
rem throughput of new order transactions
rem
CREATE TABLE bench_run_int
(
run_name VARCHAR2(20),
proc_no NUMBER,
interval NUMBER,
interval_count NUMBER,
response_time NUMBER,
think_time NUMBER
);
rem
rem Results from delivery servers
rem
CREATE TABLE tpcc_back_res
(
run_name VARCHAR2(20),
in_timing_int NUMBER,
fast NUMBER,
resp_time NUMBER,

```

```

retries NUMBER
);
rem
rem Aggregate results for all generators.
rem These results are from the measurement
interval only.
rem These results are used to calculate the TPS
rate over
rem the measurement interval.
rem
CREATE TABLE tpcc_user_res
(
run_name VARCHAR2(20),
no_men NUMBER,
fast_men NUMBER,
in_flight_men NUMBER,
retry_men NUMBER,
min_time_men NUMBER,
max_time_men NUMBER,
sum_time_men NUMBER,
ninety_per_men NUMBER,
think_min_men NUMBER,
think_max_men NUMBER,
think_sum_men NUMBER,
key_min_men NUMBER,
key_max_men NUMBER,
key_sum_men NUMBER,
no_new NUMBER,
fast_new NUMBER,
in_flight_new NUMBER,
retry_new NUMBER,
min_time_new NUMBER,
max_time_new NUMBER,
sum_time_new NUMBER,
ninety_per_new NUMBER,
think_min_new NUMBER,
think_max_new NUMBER,
think_sum_new NUMBER,
key_min_new NUMBER,
key_max_new NUMBER,
key_sum_new NUMBER,
remote_new NUMBER,
rollback_new NUMBER,
sum_ol_new NUMBER,
remote_ol_new NUMBER,
allrollback_new NUMBER,
no_pay NUMBER,
fast_pay NUMBER,
in_flight_pay NUMBER,
retry_pay NUMBER,
min_time_pay NUMBER,
max_time_pay NUMBER,
sum_time_pay NUMBER,
ninety_per_pay NUMBER,
think_min_pay NUMBER,
think_max_pay NUMBER,
think_sum_pay NUMBER,
key_min_pay NUMBER,
key_max_pay NUMBER,
key_sum_pay NUMBER,
remote_pay NUMBER,
bylast_pay NUMBER,
no_ord NUMBER,
fast_ord NUMBER,
in_flight_ord NUMBER,
retry_ord NUMBER,
min_time_ord NUMBER,
max_time_ord NUMBER,
sum_time_ord NUMBER,

```

```

ninety_per_ord NUMBER,
think_min_ord NUMBER,
think_max_ord NUMBER,
think_sum_ord NUMBER,
key_min_ord NUMBER,
key_max_ord NUMBER,
key_sum_ord NUMBER,
bylast_ord NUMBER,
no_del NUMBER,
fast_del NUMBER,
in_flight_del NUMBER,
retry_del NUMBER,
min_time_del NUMBER,
max_time_del NUMBER,
sum_time_del NUMBER,
ninety_per_del NUMBER,
think_min_del NUMBER,
think_max_del NUMBER,
think_sum_del NUMBER,
key_min_del NUMBER,
key_max_del NUMBER,
key_sum_del NUMBER,
no_sto NUMBER,
fast_sto NUMBER,
in_flight_sto NUMBER,
retry_sto NUMBER,
min_time_sto NUMBER,
max_time_sto NUMBER,
sum_time_sto NUMBER,
ninety_per_sto NUMBER,
think_min_sto NUMBER,
think_max_sto NUMBER,
think_sum_sto NUMBER,
key_min_sto NUMBER,
key_max_sto NUMBER,
key_sum_sto NUMBER,
cpu_time NUMBER,
deadlocks NUMBER
);
rem
rem Results from individual generators.
rem These results are from the measurement
interval only.
rem These results are used to calculate the TPS
rate over
rem the measurement interval.
rem
CREATE TABLE bench_user_res
(
run_name VARCHAR2(20),
audit_str VARCHAR2(10),
proc_no NUMBER,
hid NUMBER,
no_men NUMBER,
fast_men NUMBER,
in_flight_men NUMBER,
retry_men NUMBER,
min_time_men NUMBER,
max_time_men NUMBER,
sum_time_men NUMBER,
ninety_per_men NUMBER,
think_min_men NUMBER,
think_max_men NUMBER,
think_sum_men NUMBER,
key_min_men NUMBER,
key_max_men NUMBER,
key_sum_men NUMBER,
no_new NUMBER,
fast_new NUMBER,

```

```

in_flight_new NUMBER,
retry_new NUMBER,
min_time_new NUMBER,
max_time_new NUMBER,
sum_time_new NUMBER,
ninety_per_new NUMBER,
think_min_new NUMBER,
think_max_new NUMBER,
think_sum_new NUMBER,
key_min_new NUMBER,
key_max_new NUMBER,
key_sum_new NUMBER,
remote_new NUMBER,
rollback_new NUMBER,
sum_of_new NUMBER,
remote_of_new NUMBER,
allrollback_new NUMBER,
no_pay NUMBER,
fast_pay NUMBER,
in_flight_pay NUMBER,
retry_pay NUMBER,
min_time_pay NUMBER,
max_time_pay NUMBER,
sum_time_pay NUMBER,
ninety_per_pay NUMBER,
think_min_pay NUMBER,
think_max_pay NUMBER,
think_sum_pay NUMBER,
key_min_pay NUMBER,
key_max_pay NUMBER,
key_sum_pay NUMBER,
remote_pay NUMBER,
bylast_pay NUMBER,
no_ord NUMBER,
fast_ord NUMBER,
in_flight_ord NUMBER,
retry_ord NUMBER,
min_time_ord NUMBER,
max_time_ord NUMBER,
sum_time_ord NUMBER,
ninety_per_ord NUMBER,
think_min_ord NUMBER,
think_max_ord NUMBER,
think_sum_ord NUMBER,
key_min_ord NUMBER,
key_max_ord NUMBER,
key_sum_ord NUMBER,
bylast_ord NUMBER,
no_del NUMBER,
fast_del NUMBER,
in_flight_del NUMBER,
retry_del NUMBER,
min_time_del NUMBER,
max_time_del NUMBER,
sum_time_del NUMBER,
ninety_per_del NUMBER,
think_min_del NUMBER,
think_max_del NUMBER,
think_sum_del NUMBER,
key_min_del NUMBER,
key_max_del NUMBER,
key_sum_del NUMBER,
no_sto NUMBER,
fast_sto NUMBER,
in_flight_sto NUMBER,
retry_sto NUMBER,
min_time_sto NUMBER,
max_time_sto NUMBER,
sum_time_sto NUMBER,

```

```

ninety_per_sto NUMBER,
think_min_sto NUMBER,
think_max_sto NUMBER,
think_sum_sto NUMBER,
key_min_sto NUMBER,
key_max_sto NUMBER,
key_sum_sto NUMBER,
cpu_time NUMBER,
deadlocks NUMBER
);
rem
rem Aggregate results for generators on each
host.
rem These results are from the measurement
interval only.
rem These results are used to calculate the TPM
rate over
rem the measurement interval.
rem
CREATE TABLE tpcc_tpm
(
run_name VARCHAR2(20),
hid NUMBER,
no_new NUMBER
);
rem
rem Aggregate results for new order transactions.
rem These results are from the measurement
interval only.
rem
CREATE TABLE tpcc_new_res
(
run_name VARCHAR2(20),
rep1 NUMBER,
rep2 NUMBER,
rep3 NUMBER,
rep4 NUMBER,
rep5 NUMBER,
rep6 NUMBER,
rep7 NUMBER,
rep8 NUMBER,
rep9 NUMBER,
rep10 NUMBER,
rep11 NUMBER,
rep12 NUMBER,
rep13 NUMBER,
rep14 NUMBER,
rep15 NUMBER,
rep16 NUMBER,
rep17 NUMBER,
rep18 NUMBER,
rep19 NUMBER,
rep20 NUMBER,
rep21 NUMBER,
rep22 NUMBER,
rep23 NUMBER,
rep24 NUMBER,
rep25 NUMBER,
rep26 NUMBER,
rep27 NUMBER,
rep28 NUMBER,
rep29 NUMBER,
rep30 NUMBER,
rep31 NUMBER,
rep32 NUMBER,
rep33 NUMBER,
rep34 NUMBER,
rep35 NUMBER,
rep36 NUMBER,

```

```

rep37 NUMBER,
rep38 NUMBER,
rep39 NUMBER,
rep40 NUMBER,
rep41 NUMBER,
rep42 NUMBER,
rep43 NUMBER,
rep44 NUMBER,
rep45 NUMBER,
rep46 NUMBER,
rep47 NUMBER,
rep48 NUMBER,
rep49 NUMBER,
rep50 NUMBER,
rep51 NUMBER,
rep52 NUMBER,
rep53 NUMBER,
rep54 NUMBER,
rep55 NUMBER,
rep56 NUMBER,
rep57 NUMBER,
rep58 NUMBER,
rep59 NUMBER,
rep60 NUMBER,
rep61 NUMBER,
rep62 NUMBER,
rep63 NUMBER,
rep64 NUMBER,
rep65 NUMBER,
rep66 NUMBER,
rep67 NUMBER,
rep68 NUMBER,
rep69 NUMBER,
rep70 NUMBER,
rep71 NUMBER,
rep72 NUMBER,
rep73 NUMBER,
rep74 NUMBER,
rep75 NUMBER,
rep76 NUMBER,
rep77 NUMBER,
rep78 NUMBER,
rep79 NUMBER,
rep80 NUMBER,
rep81 NUMBER,
rep82 NUMBER,
rep83 NUMBER,
rep84 NUMBER,
rep85 NUMBER,
rep86 NUMBER,
rep87 NUMBER,
rep88 NUMBER,
rep89 NUMBER,
rep90 NUMBER,
rep91 NUMBER,
rep92 NUMBER,
rep93 NUMBER,
rep94 NUMBER,
rep95 NUMBER,
rep96 NUMBER,
rep97 NUMBER,
rep98 NUMBER,
rep99 NUMBER,
rep100 NUMBER,
thk1 NUMBER,
thk2 NUMBER,
thk3 NUMBER,
thk4 NUMBER,
thk5 NUMBER,

```

```

thk6    NUMBER,
thk7    NUMBER,
thk8    NUMBER,
thk9    NUMBER,
thk10   NUMBER,
thk11   NUMBER,
thk12   NUMBER,
thk13   NUMBER,
thk14   NUMBER,
thk15   NUMBER,
thk16   NUMBER,
thk17   NUMBER,
thk18   NUMBER,
thk19   NUMBER,
thk20   NUMBER,
thk21   NUMBER,
thk22   NUMBER,
thk23   NUMBER,
thk24   NUMBER,
thk25   NUMBER,
key1    NUMBER,
key2    NUMBER,
key3    NUMBER,
key4    NUMBER,
key5    NUMBER,
key6    NUMBER,
key7    NUMBER,
key8    NUMBER,
key9    NUMBER,
key10   NUMBER
);

rem
rem Results for new order transactions.
rem These results are from the measurement
rem interval only.
rem
rem CREATE TABLE bench_new_res
rem (
rem   run_name  VARCHAR2(20),
rem   audit_str VARCHAR2(10),
rem   proc_no   NUMBER,
rem   rep1     NUMBER,
rem   rep2     NUMBER,
rem   rep3     NUMBER,
rem   rep4     NUMBER,
rem   rep5     NUMBER,
rem   rep6     NUMBER,
rem   rep7     NUMBER,
rem   rep8     NUMBER,
rem   rep9     NUMBER,
rem   rep10    NUMBER,
rem   rep11    NUMBER,
rem   rep12    NUMBER,
rem   rep13    NUMBER,
rem   rep14    NUMBER,
rem   rep15    NUMBER,
rem   rep16    NUMBER,
rem   rep17    NUMBER,
rem   rep18    NUMBER,
rem   rep19    NUMBER,
rem   rep20    NUMBER,
rem   rep21    NUMBER,
rem   rep22    NUMBER,
rem   rep23    NUMBER,
rem   rep24    NUMBER,
rem   rep25    NUMBER,
rem   rep26    NUMBER,
rem   rep27    NUMBER,

```

```

rep28   NUMBER,
rep29   NUMBER,
rep30   NUMBER,
rep31   NUMBER,
rep32   NUMBER,
rep33   NUMBER,
rep34   NUMBER,
rep35   NUMBER,
rep36   NUMBER,
rep37   NUMBER,
rep38   NUMBER,
rep39   NUMBER,
rep40   NUMBER,
rep41   NUMBER,
rep42   NUMBER,
rep43   NUMBER,
rep44   NUMBER,
rep45   NUMBER,
rep46   NUMBER,
rep47   NUMBER,
rep48   NUMBER,
rep49   NUMBER,
rep50   NUMBER,
rep51   NUMBER,
rep52   NUMBER,
rep53   NUMBER,
rep54   NUMBER,
rep55   NUMBER,
rep56   NUMBER,
rep57   NUMBER,
rep58   NUMBER,
rep59   NUMBER,
rep60   NUMBER,
rep61   NUMBER,
rep62   NUMBER,
rep63   NUMBER,
rep64   NUMBER,
rep65   NUMBER,
rep66   NUMBER,
rep67   NUMBER,
rep68   NUMBER,
rep69   NUMBER,
rep70   NUMBER,
rep71   NUMBER,
rep72   NUMBER,
rep73   NUMBER,
rep74   NUMBER,
rep75   NUMBER,
rep76   NUMBER,
rep77   NUMBER,
rep78   NUMBER,
rep79   NUMBER,
rep80   NUMBER,
rep81   NUMBER,
rep82   NUMBER,
rep83   NUMBER,
rep84   NUMBER,
rep85   NUMBER,
rep86   NUMBER,
rep87   NUMBER,
rep88   NUMBER,
rep89   NUMBER,
rep90   NUMBER,
rep91   NUMBER,
rep92   NUMBER,
rep93   NUMBER,
rep94   NUMBER,
rep95   NUMBER,
rep96   NUMBER,

```

```

rep97   NUMBER,
rep98   NUMBER,
rep99   NUMBER,
rep100  NUMBER,
thk1    NUMBER,
thk2    NUMBER,
thk3    NUMBER,
thk4    NUMBER,
thk5    NUMBER,
thk6    NUMBER,
thk7    NUMBER,
thk8    NUMBER,
thk9    NUMBER,
thk10   NUMBER,
thk11   NUMBER,
thk12   NUMBER,
thk13   NUMBER,
thk14   NUMBER,
thk15   NUMBER,
thk16   NUMBER,
thk17   NUMBER,
thk18   NUMBER,
thk19   NUMBER,
thk20   NUMBER,
thk21   NUMBER,
thk22   NUMBER,
thk23   NUMBER,
thk24   NUMBER,
thk25   NUMBER,
key1    NUMBER,
key2    NUMBER,
key3    NUMBER,
key4    NUMBER,
key5    NUMBER,
key6    NUMBER,
key7    NUMBER,
key8    NUMBER,
key9    NUMBER,
key10   NUMBER
);

rem
rem Aggregate results for payment transactions.
rem These results are from the measurement
rem interval only.
rem
rem CREATE TABLE tpcc_pay_res
rem (
rem   run_name  VARCHAR2(20),
rem   rep1     NUMBER,
rem   rep2     NUMBER,
rem   rep3     NUMBER,
rem   rep4     NUMBER,
rem   rep5     NUMBER,
rem   rep6     NUMBER,
rem   rep7     NUMBER,
rem   rep8     NUMBER,
rem   rep9     NUMBER,
rem   rep10    NUMBER,
rem   rep11    NUMBER,
rem   rep12    NUMBER,
rem   rep13    NUMBER,
rem   rep14    NUMBER,
rem   rep15    NUMBER,
rem   rep16    NUMBER,
rem   rep17    NUMBER,
rem   rep18    NUMBER,
rem   rep19    NUMBER,
rem   rep20    NUMBER,

```

```

rep21 NUMBER,
rep22 NUMBER,
rep23 NUMBER,
rep24 NUMBER,
rep25 NUMBER,
rep26 NUMBER,
rep27 NUMBER,
rep28 NUMBER,
rep29 NUMBER,
rep30 NUMBER,
rep31 NUMBER,
rep32 NUMBER,
rep33 NUMBER,
rep34 NUMBER,
rep35 NUMBER,
rep36 NUMBER,
rep37 NUMBER,
rep38 NUMBER,
rep39 NUMBER,
rep40 NUMBER,
rep41 NUMBER,
rep42 NUMBER,
rep43 NUMBER,
rep44 NUMBER,
rep45 NUMBER,
rep46 NUMBER,
rep47 NUMBER,
rep48 NUMBER,
rep49 NUMBER,
rep50 NUMBER,
rep51 NUMBER,
rep52 NUMBER,
rep53 NUMBER,
rep54 NUMBER,
rep55 NUMBER,
rep56 NUMBER,
rep57 NUMBER,
rep58 NUMBER,
rep59 NUMBER,
rep60 NUMBER,
rep61 NUMBER,
rep62 NUMBER,
rep63 NUMBER,
rep64 NUMBER,
rep65 NUMBER,
rep66 NUMBER,
rep67 NUMBER,
rep68 NUMBER,
rep69 NUMBER,
rep70 NUMBER,
rep71 NUMBER,
rep72 NUMBER,
rep73 NUMBER,
rep74 NUMBER,
rep75 NUMBER,
rep76 NUMBER,
rep77 NUMBER,
rep78 NUMBER,
rep79 NUMBER,
rep80 NUMBER,
rep81 NUMBER,
rep82 NUMBER,
rep83 NUMBER,
rep84 NUMBER,
rep85 NUMBER,
rep86 NUMBER,
rep87 NUMBER,
rep88 NUMBER,
rep89 NUMBER,

```

```

rep90 NUMBER,
rep91 NUMBER,
rep92 NUMBER,
rep93 NUMBER,
rep94 NUMBER,
rep95 NUMBER,
rep96 NUMBER,
rep97 NUMBER,
rep98 NUMBER,
rep99 NUMBER,
rep100 NUMBER,
thk1 NUMBER,
thk2 NUMBER,
thk3 NUMBER,
thk4 NUMBER,
thk5 NUMBER,
thk6 NUMBER,
thk7 NUMBER,
thk8 NUMBER,
thk9 NUMBER,
thk10 NUMBER,
thk11 NUMBER,
thk12 NUMBER,
thk13 NUMBER,
thk14 NUMBER,
thk15 NUMBER,
thk16 NUMBER,
thk17 NUMBER,
thk18 NUMBER,
thk19 NUMBER,
thk20 NUMBER,
thk21 NUMBER,
thk22 NUMBER,
thk23 NUMBER,
thk24 NUMBER,
thk25 NUMBER,
key1 NUMBER,
key2 NUMBER,
key3 NUMBER,
key4 NUMBER,
key5 NUMBER,
key6 NUMBER,
key7 NUMBER,
key8 NUMBER,
key9 NUMBER,
key10 NUMBER
);

```

```

rem
rem Results for payment transactions.
rem These results are from the measurement
interval only.
rem

```

```

CREATE TABLE bench_pay_res
(
  run_name VARCHAR2(20),
  audit_str VARCHAR2(10),
  proc_no NUMBER,
  rep1 NUMBER,
  rep2 NUMBER,
  rep3 NUMBER,
  rep4 NUMBER,
  rep5 NUMBER,
  rep6 NUMBER,
  rep7 NUMBER,
  rep8 NUMBER,
  rep9 NUMBER,
  rep10 NUMBER,
  rep11 NUMBER,

```

```

rep12 NUMBER,
rep13 NUMBER,
rep14 NUMBER,
rep15 NUMBER,
rep16 NUMBER,
rep17 NUMBER,
rep18 NUMBER,
rep19 NUMBER,
rep20 NUMBER,
rep21 NUMBER,
rep22 NUMBER,
rep23 NUMBER,
rep24 NUMBER,
rep25 NUMBER,
rep26 NUMBER,
rep27 NUMBER,
rep28 NUMBER,
rep29 NUMBER,
rep30 NUMBER,
rep31 NUMBER,
rep32 NUMBER,
rep33 NUMBER,
rep34 NUMBER,
rep35 NUMBER,
rep36 NUMBER,
rep37 NUMBER,
rep38 NUMBER,
rep39 NUMBER,
rep40 NUMBER,
rep41 NUMBER,
rep42 NUMBER,
rep43 NUMBER,
rep44 NUMBER,
rep45 NUMBER,
rep46 NUMBER,
rep47 NUMBER,
rep48 NUMBER,
rep49 NUMBER,
rep50 NUMBER,
rep51 NUMBER,
rep52 NUMBER,
rep53 NUMBER,
rep54 NUMBER,
rep55 NUMBER,
rep56 NUMBER,
rep57 NUMBER,
rep58 NUMBER,
rep59 NUMBER,
rep60 NUMBER,
rep61 NUMBER,
rep62 NUMBER,
rep63 NUMBER,
rep64 NUMBER,
rep65 NUMBER,
rep66 NUMBER,
rep67 NUMBER,
rep68 NUMBER,
rep69 NUMBER,
rep70 NUMBER,
rep71 NUMBER,
rep72 NUMBER,
rep73 NUMBER,
rep74 NUMBER,
rep75 NUMBER,
rep76 NUMBER,
rep77 NUMBER,
rep78 NUMBER,
rep79 NUMBER,
rep80 NUMBER,

```

```

rep81 NUMBER,
rep82 NUMBER,
rep83 NUMBER,
rep84 NUMBER,
rep85 NUMBER,
rep86 NUMBER,
rep87 NUMBER,
rep88 NUMBER,
rep89 NUMBER,
rep90 NUMBER,
rep91 NUMBER,
rep92 NUMBER,
rep93 NUMBER,
rep94 NUMBER,
rep95 NUMBER,
rep96 NUMBER,
rep97 NUMBER,
rep98 NUMBER,
rep99 NUMBER,
rep100 NUMBER,
thk1 NUMBER,
thk2 NUMBER,
thk3 NUMBER,
thk4 NUMBER,
thk5 NUMBER,
thk6 NUMBER,
thk7 NUMBER,
thk8 NUMBER,
thk9 NUMBER,
thk10 NUMBER,
thk11 NUMBER,
thk12 NUMBER,
thk13 NUMBER,
thk14 NUMBER,
thk15 NUMBER,
thk16 NUMBER,
thk17 NUMBER,
thk18 NUMBER,
thk19 NUMBER,
thk20 NUMBER,
thk21 NUMBER,
thk22 NUMBER,
thk23 NUMBER,
thk24 NUMBER,
thk25 NUMBER,
key1 NUMBER,
key2 NUMBER,
key3 NUMBER,
key4 NUMBER,
key5 NUMBER,
key6 NUMBER,
key7 NUMBER,
key8 NUMBER,
key9 NUMBER,
key10 NUMBER
);

rem
rem Aggregate results for order status
rem transactions.
rem These results are from the measurement
rem interval only.
rem
CREATE TABLE tpcc_ord_res
(
run_name VARCHAR2(20),
rep1 NUMBER,
rep2 NUMBER,
rep3 NUMBER,

```

```

rep4 NUMBER,
rep5 NUMBER,
rep6 NUMBER,
rep7 NUMBER,
rep8 NUMBER,
rep9 NUMBER,
rep10 NUMBER,
rep11 NUMBER,
rep12 NUMBER,
rep13 NUMBER,
rep14 NUMBER,
rep15 NUMBER,
rep16 NUMBER,
rep17 NUMBER,
rep18 NUMBER,
rep19 NUMBER,
rep20 NUMBER,
rep21 NUMBER,
rep22 NUMBER,
rep23 NUMBER,
rep24 NUMBER,
rep25 NUMBER,
rep26 NUMBER,
rep27 NUMBER,
rep28 NUMBER,
rep29 NUMBER,
rep30 NUMBER,
rep31 NUMBER,
rep32 NUMBER,
rep33 NUMBER,
rep34 NUMBER,
rep35 NUMBER,
rep36 NUMBER,
rep37 NUMBER,
rep38 NUMBER,
rep39 NUMBER,
rep40 NUMBER,
rep41 NUMBER,
rep42 NUMBER,
rep43 NUMBER,
rep44 NUMBER,
rep45 NUMBER,
rep46 NUMBER,
rep47 NUMBER,
rep48 NUMBER,
rep49 NUMBER,
rep50 NUMBER,
rep51 NUMBER,
rep52 NUMBER,
rep53 NUMBER,
rep54 NUMBER,
rep55 NUMBER,
rep56 NUMBER,
rep57 NUMBER,
rep58 NUMBER,
rep59 NUMBER,
rep60 NUMBER,
rep61 NUMBER,
rep62 NUMBER,
rep63 NUMBER,
rep64 NUMBER,
rep65 NUMBER,
rep66 NUMBER,
rep67 NUMBER,
rep68 NUMBER,
rep69 NUMBER,
rep70 NUMBER,
rep71 NUMBER,
rep72 NUMBER,

```

```

rep73 NUMBER,
rep74 NUMBER,
rep75 NUMBER,
rep76 NUMBER,
rep77 NUMBER,
rep78 NUMBER,
rep79 NUMBER,
rep80 NUMBER,
rep81 NUMBER,
rep82 NUMBER,
rep83 NUMBER,
rep84 NUMBER,
rep85 NUMBER,
rep86 NUMBER,
rep87 NUMBER,
rep88 NUMBER,
rep89 NUMBER,
rep90 NUMBER,
rep91 NUMBER,
rep92 NUMBER,
rep93 NUMBER,
rep94 NUMBER,
rep95 NUMBER,
rep96 NUMBER,
rep97 NUMBER,
rep98 NUMBER,
rep99 NUMBER,
rep100 NUMBER,
thk1 NUMBER,
thk2 NUMBER,
thk3 NUMBER,
thk4 NUMBER,
thk5 NUMBER,
thk6 NUMBER,
thk7 NUMBER,
thk8 NUMBER,
thk9 NUMBER,
thk10 NUMBER,
thk11 NUMBER,
thk12 NUMBER,
thk13 NUMBER,
thk14 NUMBER,
thk15 NUMBER,
thk16 NUMBER,
thk17 NUMBER,
thk18 NUMBER,
thk19 NUMBER,
thk20 NUMBER,
thk21 NUMBER,
thk22 NUMBER,
thk23 NUMBER,
thk24 NUMBER,
thk25 NUMBER,
key1 NUMBER,
key2 NUMBER,
key3 NUMBER,
key4 NUMBER,
key5 NUMBER,
key6 NUMBER,
key7 NUMBER,
key8 NUMBER,
key9 NUMBER,
key10 NUMBER
);

rem
rem Results for order status transactions.
rem These results are from the measurement
rem interval only.

```

```

rem
CREATE TABLE bench_ord_res
(
  run_name    VARCHAR2(20),
  audit_str   VARCHAR2(10),
  proc_no     NUMBER,
  rep1        NUMBER,
  rep2        NUMBER,
  rep3        NUMBER,
  rep4        NUMBER,
  rep5        NUMBER,
  rep6        NUMBER,
  rep7        NUMBER,
  rep8        NUMBER,
  rep9        NUMBER,
  rep10       NUMBER,
  rep11       NUMBER,
  rep12       NUMBER,
  rep13       NUMBER,
  rep14       NUMBER,
  rep15       NUMBER,
  rep16       NUMBER,
  rep17       NUMBER,
  rep18       NUMBER,
  rep19       NUMBER,
  rep20       NUMBER,
  rep21       NUMBER,
  rep22       NUMBER,
  rep23       NUMBER,
  rep24       NUMBER,
  rep25       NUMBER,
  rep26       NUMBER,
  rep27       NUMBER,
  rep28       NUMBER,
  rep29       NUMBER,
  rep30       NUMBER,
  rep31       NUMBER,
  rep32       NUMBER,
  rep33       NUMBER,
  rep34       NUMBER,
  rep35       NUMBER,
  rep36       NUMBER,
  rep37       NUMBER,
  rep38       NUMBER,
  rep39       NUMBER,
  rep40       NUMBER,
  rep41       NUMBER,
  rep42       NUMBER,
  rep43       NUMBER,
  rep44       NUMBER,
  rep45       NUMBER,
  rep46       NUMBER,
  rep47       NUMBER,
  rep48       NUMBER,
  rep49       NUMBER,
  rep50       NUMBER,
  rep51       NUMBER,
  rep52       NUMBER,
  rep53       NUMBER,
  rep54       NUMBER,
  rep55       NUMBER,
  rep56       NUMBER,
  rep57       NUMBER,
  rep58       NUMBER,
  rep59       NUMBER,
  rep60       NUMBER,
  rep61       NUMBER,
  rep62       NUMBER,
  rep63       NUMBER,

```

```

rep64        NUMBER,
rep65        NUMBER,
rep66        NUMBER,
rep67        NUMBER,
rep68        NUMBER,
rep69        NUMBER,
rep70        NUMBER,
rep71        NUMBER,
rep72        NUMBER,
rep73        NUMBER,
rep74        NUMBER,
rep75        NUMBER,
rep76        NUMBER,
rep77        NUMBER,
rep78        NUMBER,
rep79        NUMBER,
rep80        NUMBER,
rep81        NUMBER,
rep82        NUMBER,
rep83        NUMBER,
rep84        NUMBER,
rep85        NUMBER,
rep86        NUMBER,
rep87        NUMBER,
rep88        NUMBER,
rep89        NUMBER,
rep90        NUMBER,
rep91        NUMBER,
rep92        NUMBER,
rep93        NUMBER,
rep94        NUMBER,
rep95        NUMBER,
rep96        NUMBER,
rep97        NUMBER,
rep98        NUMBER,
rep99        NUMBER,
rep100       NUMBER,
thk1         NUMBER,
thk2         NUMBER,
thk3         NUMBER,
thk4         NUMBER,
thk5         NUMBER,
thk6         NUMBER,
thk7         NUMBER,
thk8         NUMBER,
thk9         NUMBER,
thk10        NUMBER,
thk11        NUMBER,
thk12        NUMBER,
thk13        NUMBER,
thk14        NUMBER,
thk15        NUMBER,
thk16        NUMBER,
thk17        NUMBER,
thk18        NUMBER,
thk19        NUMBER,
thk20        NUMBER,
thk21        NUMBER,
thk22        NUMBER,
thk23        NUMBER,
thk24        NUMBER,
thk25        NUMBER,
key1         NUMBER,
key2         NUMBER,
key3         NUMBER,
key4         NUMBER,
key5         NUMBER,
key6         NUMBER,
key7         NUMBER,

```

```

key8         NUMBER,
key9         NUMBER,
key10        NUMBER
);
rem
rem Aggregate results for delivery transactions.
rem These results are from the measurement
rem interval only.
rem
CREATE TABLE tpcc_del_res
(
  run_name    VARCHAR2(20),
  rep1        NUMBER,
  rep2        NUMBER,
  rep3        NUMBER,
  rep4        NUMBER,
  rep5        NUMBER,
  rep6        NUMBER,
  rep7        NUMBER,
  rep8        NUMBER,
  rep9        NUMBER,
  rep10       NUMBER,
  rep11       NUMBER,
  rep12       NUMBER,
  rep13       NUMBER,
  rep14       NUMBER,
  rep15       NUMBER,
  rep16       NUMBER,
  rep17       NUMBER,
  rep18       NUMBER,
  rep19       NUMBER,
  rep20       NUMBER,
  rep21       NUMBER,
  rep22       NUMBER,
  rep23       NUMBER,
  rep24       NUMBER,
  rep25       NUMBER,
  rep26       NUMBER,
  rep27       NUMBER,
  rep28       NUMBER,
  rep29       NUMBER,
  rep30       NUMBER,
  rep31       NUMBER,
  rep32       NUMBER,
  rep33       NUMBER,
  rep34       NUMBER,
  rep35       NUMBER,
  rep36       NUMBER,
  rep37       NUMBER,
  rep38       NUMBER,
  rep39       NUMBER,
  rep40       NUMBER,
  rep41       NUMBER,
  rep42       NUMBER,
  rep43       NUMBER,
  rep44       NUMBER,
  rep45       NUMBER,
  rep46       NUMBER,
  rep47       NUMBER,
  rep48       NUMBER,
  rep49       NUMBER,
  rep50       NUMBER,
  rep51       NUMBER,
  rep52       NUMBER,
  rep53       NUMBER,
  rep54       NUMBER,
  rep55       NUMBER,
  rep56       NUMBER,

```

```

rep57 NUMBER,
rep58 NUMBER,
rep59 NUMBER,
rep60 NUMBER,
rep61 NUMBER,
rep62 NUMBER,
rep63 NUMBER,
rep64 NUMBER,
rep65 NUMBER,
rep66 NUMBER,
rep67 NUMBER,
rep68 NUMBER,
rep69 NUMBER,
rep70 NUMBER,
rep71 NUMBER,
rep72 NUMBER,
rep73 NUMBER,
rep74 NUMBER,
rep75 NUMBER,
rep76 NUMBER,
rep77 NUMBER,
rep78 NUMBER,
rep79 NUMBER,
rep80 NUMBER,
rep81 NUMBER,
rep82 NUMBER,
rep83 NUMBER,
rep84 NUMBER,
rep85 NUMBER,
rep86 NUMBER,
rep87 NUMBER,
rep88 NUMBER,
rep89 NUMBER,
rep90 NUMBER,
rep91 NUMBER,
rep92 NUMBER,
rep93 NUMBER,
rep94 NUMBER,
rep95 NUMBER,
rep96 NUMBER,
rep97 NUMBER,
rep98 NUMBER,
rep99 NUMBER,
rep100 NUMBER,
thk1 NUMBER,
thk2 NUMBER,
thk3 NUMBER,
thk4 NUMBER,
thk5 NUMBER,
thk6 NUMBER,
thk7 NUMBER,
thk8 NUMBER,
thk9 NUMBER,
thk10 NUMBER,
thk11 NUMBER,
thk12 NUMBER,
thk13 NUMBER,
thk14 NUMBER,
thk15 NUMBER,
thk16 NUMBER,
thk17 NUMBER,
thk18 NUMBER,
thk19 NUMBER,
thk20 NUMBER,
thk21 NUMBER,
thk22 NUMBER,
thk23 NUMBER,
thk24 NUMBER,
thk25 NUMBER,

```

```

key1 NUMBER,
key2 NUMBER,
key3 NUMBER,
key4 NUMBER,
key5 NUMBER,
key6 NUMBER,
key7 NUMBER,
key8 NUMBER,
key9 NUMBER,
key10 NUMBER
);
rem
rem Results for delivery transactions.
rem These results are from the measurement
rem interval only.
rem
CREATE TABLE bench_del_res
(
run_name VARCHAR2(20),
audit_str VARCHAR2(10),
proc_no NUMBER,
rep1 NUMBER,
rep2 NUMBER,
rep3 NUMBER,
rep4 NUMBER,
rep5 NUMBER,
rep6 NUMBER,
rep7 NUMBER,
rep8 NUMBER,
rep9 NUMBER,
rep10 NUMBER,
rep11 NUMBER,
rep12 NUMBER,
rep13 NUMBER,
rep14 NUMBER,
rep15 NUMBER,
rep16 NUMBER,
rep17 NUMBER,
rep18 NUMBER,
rep19 NUMBER,
rep20 NUMBER,
rep21 NUMBER,
rep22 NUMBER,
rep23 NUMBER,
rep24 NUMBER,
rep25 NUMBER,
rep26 NUMBER,
rep27 NUMBER,
rep28 NUMBER,
rep29 NUMBER,
rep30 NUMBER,
rep31 NUMBER,
rep32 NUMBER,
rep33 NUMBER,
rep34 NUMBER,
rep35 NUMBER,
rep36 NUMBER,
rep37 NUMBER,
rep38 NUMBER,
rep39 NUMBER,
rep40 NUMBER,
rep41 NUMBER,
rep42 NUMBER,
rep43 NUMBER,
rep44 NUMBER,
rep45 NUMBER,
rep46 NUMBER,
rep47 NUMBER,

```

```

rep48 NUMBER,
rep49 NUMBER,
rep50 NUMBER,
rep51 NUMBER,
rep52 NUMBER,
rep53 NUMBER,
rep54 NUMBER,
rep55 NUMBER,
rep56 NUMBER,
rep57 NUMBER,
rep58 NUMBER,
rep59 NUMBER,
rep60 NUMBER,
rep61 NUMBER,
rep62 NUMBER,
rep63 NUMBER,
rep64 NUMBER,
rep65 NUMBER,
rep66 NUMBER,
rep67 NUMBER,
rep68 NUMBER,
rep69 NUMBER,
rep70 NUMBER,
rep71 NUMBER,
rep72 NUMBER,
rep73 NUMBER,
rep74 NUMBER,
rep75 NUMBER,
rep76 NUMBER,
rep77 NUMBER,
rep78 NUMBER,
rep79 NUMBER,
rep80 NUMBER,
rep81 NUMBER,
rep82 NUMBER,
rep83 NUMBER,
rep84 NUMBER,
rep85 NUMBER,
rep86 NUMBER,
rep87 NUMBER,
rep88 NUMBER,
rep89 NUMBER,
rep90 NUMBER,
rep91 NUMBER,
rep92 NUMBER,
rep93 NUMBER,
rep94 NUMBER,
rep95 NUMBER,
rep96 NUMBER,
rep97 NUMBER,
rep98 NUMBER,
rep99 NUMBER,
rep100 NUMBER,
thk1 NUMBER,
thk2 NUMBER,
thk3 NUMBER,
thk4 NUMBER,
thk5 NUMBER,
thk6 NUMBER,
thk7 NUMBER,
thk8 NUMBER,
thk9 NUMBER,
thk10 NUMBER,
thk11 NUMBER,
thk12 NUMBER,
thk13 NUMBER,
thk14 NUMBER,
thk15 NUMBER,
thk16 NUMBER,

```

```

thk17    NUMBER,
thk18    NUMBER,
thk19    NUMBER,
thk20    NUMBER,
thk21    NUMBER,
thk22    NUMBER,
thk23    NUMBER,
thk24    NUMBER,
thk25    NUMBER,
key1     NUMBER,
key2     NUMBER,
key3     NUMBER,
key4     NUMBER,
key5     NUMBER,
key6     NUMBER,
key7     NUMBER,
key8     NUMBER,
key9     NUMBER,
key10    NUMBER
);
rem
rem Aggregate results for stock level transactions.
rem These results are from the measurement
rem interval only.
rem
CREATE TABLE tpcc_sto_res
(
  run_name  VARCHAR2(20),
  rep1      NUMBER,
  rep2      NUMBER,
  rep3      NUMBER,
  rep4      NUMBER,
  rep5      NUMBER,
  rep6      NUMBER,
  rep7      NUMBER,
  rep8      NUMBER,
  rep9      NUMBER,
  rep10     NUMBER,
  rep11     NUMBER,
  rep12     NUMBER,
  rep13     NUMBER,
  rep14     NUMBER,
  rep15     NUMBER,
  rep16     NUMBER,
  rep17     NUMBER,
  rep18     NUMBER,
  rep19     NUMBER,
  rep20     NUMBER,
  rep21     NUMBER,
  rep22     NUMBER,
  rep23     NUMBER,
  rep24     NUMBER,
  rep25     NUMBER,
  rep26     NUMBER,
  rep27     NUMBER,
  rep28     NUMBER,
  rep29     NUMBER,
  rep30     NUMBER,
  rep31     NUMBER,
  rep32     NUMBER,
  rep33     NUMBER,
  rep34     NUMBER,
  rep35     NUMBER,
  rep36     NUMBER,
  rep37     NUMBER,
  rep38     NUMBER,
  rep39     NUMBER,
  rep40     NUMBER,
  rep41     NUMBER,

```

```

rep42    NUMBER,
rep43    NUMBER,
rep44    NUMBER,
rep45    NUMBER,
rep46    NUMBER,
rep47    NUMBER,
rep48    NUMBER,
rep49    NUMBER,
rep50    NUMBER,
rep51    NUMBER,
rep52    NUMBER,
rep53    NUMBER,
rep54    NUMBER,
rep55    NUMBER,
rep56    NUMBER,
rep57    NUMBER,
rep58    NUMBER,
rep59    NUMBER,
rep60    NUMBER,
rep61    NUMBER,
rep62    NUMBER,
rep63    NUMBER,
rep64    NUMBER,
rep65    NUMBER,
rep66    NUMBER,
rep67    NUMBER,
rep68    NUMBER,
rep69    NUMBER,
rep70    NUMBER,
rep71    NUMBER,
rep72    NUMBER,
rep73    NUMBER,
rep74    NUMBER,
rep75    NUMBER,
rep76    NUMBER,
rep77    NUMBER,
rep78    NUMBER,
rep79    NUMBER,
rep80    NUMBER,
rep81    NUMBER,
rep82    NUMBER,
rep83    NUMBER,
rep84    NUMBER,
rep85    NUMBER,
rep86    NUMBER,
rep87    NUMBER,
rep88    NUMBER,
rep89    NUMBER,
rep90    NUMBER,
rep91    NUMBER,
rep92    NUMBER,
rep93    NUMBER,
rep94    NUMBER,
rep95    NUMBER,
rep96    NUMBER,
rep97    NUMBER,
rep98    NUMBER,
rep99    NUMBER,
rep100   NUMBER,
thk1     NUMBER,
thk2     NUMBER,
thk3     NUMBER,
thk4     NUMBER,
thk5     NUMBER,
thk6     NUMBER,
thk7     NUMBER,
thk8     NUMBER,
thk9     NUMBER,
thk10    NUMBER,

```

```

thk11    NUMBER,
thk12    NUMBER,
thk13    NUMBER,
thk14    NUMBER,
thk15    NUMBER,
thk16    NUMBER,
thk17    NUMBER,
thk18    NUMBER,
thk19    NUMBER,
thk20    NUMBER,
thk21    NUMBER,
thk22    NUMBER,
thk23    NUMBER,
thk24    NUMBER,
thk25    NUMBER,
key1     NUMBER,
key2     NUMBER,
key3     NUMBER,
key4     NUMBER,
key5     NUMBER,
key6     NUMBER,
key7     NUMBER,
key8     NUMBER,
key9     NUMBER,
key10    NUMBER
);
rem
rem Results for stock level transactions.
rem These results are from the measurement
rem interval only.
rem
CREATE TABLE bench_sto_res
(
  run_name  VARCHAR2(20),
  audit_str VARCHAR2(10),
  proc_no   NUMBER,
  rep1      NUMBER,
  rep2      NUMBER,
  rep3      NUMBER,
  rep4      NUMBER,
  rep5      NUMBER,
  rep6      NUMBER,
  rep7      NUMBER,
  rep8      NUMBER,
  rep9      NUMBER,
  rep10     NUMBER,
  rep11     NUMBER,
  rep12     NUMBER,
  rep13     NUMBER,
  rep14     NUMBER,
  rep15     NUMBER,
  rep16     NUMBER,
  rep17     NUMBER,
  rep18     NUMBER,
  rep19     NUMBER,
  rep20     NUMBER,
  rep21     NUMBER,
  rep22     NUMBER,
  rep23     NUMBER,
  rep24     NUMBER,
  rep25     NUMBER,
  rep26     NUMBER,
  rep27     NUMBER,
  rep28     NUMBER,
  rep29     NUMBER,
  rep30     NUMBER,
  rep31     NUMBER,
  rep32     NUMBER,
  rep33     NUMBER,

```

```

rep34 NUMBER,
rep35 NUMBER,
rep36 NUMBER,
rep37 NUMBER,
rep38 NUMBER,
rep39 NUMBER,
rep40 NUMBER,
rep41 NUMBER,
rep42 NUMBER,
rep43 NUMBER,
rep44 NUMBER,
rep45 NUMBER,
rep46 NUMBER,
rep47 NUMBER,
rep48 NUMBER,
rep49 NUMBER,
rep50 NUMBER,
rep51 NUMBER,
rep52 NUMBER,
rep53 NUMBER,
rep54 NUMBER,
rep55 NUMBER,
rep56 NUMBER,
rep57 NUMBER,
rep58 NUMBER,
rep59 NUMBER,
rep60 NUMBER,
rep61 NUMBER,
rep62 NUMBER,
rep63 NUMBER,
rep64 NUMBER,
rep65 NUMBER,
rep66 NUMBER,
rep67 NUMBER,
rep68 NUMBER,
rep69 NUMBER,
rep70 NUMBER,
rep71 NUMBER,
rep72 NUMBER,
rep73 NUMBER,
rep74 NUMBER,
rep75 NUMBER,
rep76 NUMBER,
rep77 NUMBER,
rep78 NUMBER,
rep79 NUMBER,
rep80 NUMBER,
rep81 NUMBER,
rep82 NUMBER,
rep83 NUMBER,
rep84 NUMBER,
rep85 NUMBER,
rep86 NUMBER,
rep87 NUMBER,
rep88 NUMBER,
rep89 NUMBER,
rep90 NUMBER,
rep91 NUMBER,
rep92 NUMBER,
rep93 NUMBER,
rep94 NUMBER,
rep95 NUMBER,
rep96 NUMBER,
rep97 NUMBER,
rep98 NUMBER,
rep99 NUMBER,
rep100 NUMBER,
thk1 NUMBER,
thk2 NUMBER,

```

```

thk3 NUMBER,
thk4 NUMBER,
thk5 NUMBER,
thk6 NUMBER,
thk7 NUMBER,
thk8 NUMBER,
thk9 NUMBER,
thk10 NUMBER,
thk11 NUMBER,
thk12 NUMBER,
thk13 NUMBER,
thk14 NUMBER,
thk15 NUMBER,
thk16 NUMBER,
thk17 NUMBER,
thk18 NUMBER,
thk19 NUMBER,
thk20 NUMBER,
thk21 NUMBER,
thk22 NUMBER,
thk23 NUMBER,
thk24 NUMBER,
thk25 NUMBER,
key1 NUMBER,
key2 NUMBER,
key3 NUMBER,
key4 NUMBER,
key5 NUMBER,
key6 NUMBER,
key7 NUMBER,
key8 NUMBER,
key9 NUMBER,
key10 NUMBER
);
commit;
set echo off;
rem spool off;
rem exit;
:-----:
c_stat_rac.sql
:-----:
rem
rem
rem=====
+
rem Copyright (c) 1997 Oracle Corp,
Redwood Shores, CA |
rem All Rights Reserved
|
rem=====
+
rem FILENAME
rem cs_tpcc.sql
rem DESCRIPTION
rem Create tables for saving TPC-C results.
rem=====
+
rem Usage: sqlplus user/password @cs_tpcc.sql
rem spool cs_tpcc.log

connect tpcc/tpcc;
set echo on

DROP TABLE tpcc_run_desc;
DROP TABLE tpcc_run_int;
DROP TABLE bench_run_int;
DROP TABLE tpcc_back_res;
DROP TABLE tpcc_user_res;
DROP TABLE bench_user_res;

```

```

DROP TABLE tpcc_tpm;
DROP TABLE tpcc_new_res;
DROP TABLE bench_new_res;
DROP TABLE tpcc_pay_res;
DROP TABLE bench_pay_res;
DROP TABLE tpcc_ord_res;
DROP TABLE bench_ord_res;
DROP TABLE tpcc_del_res;
DROP TABLE bench_del_res;
DROP TABLE tpcc_sto_res;
DROP TABLE bench_sto_res;
rem
rem description of a run
rem
CREATE TABLE tpcc_run_desc
(
run_name VARCHAR2(20),
rundate DATE,
time NUMBER,
rampup NUMBER,
rampdown NUMBER,
warehouses NUMBER,
customers NUMBER,
users NUMBER,
driver VARCHAR2(40),
commnt VARCHAR2(80)
);
rem
rem throughput of new order transactions
rem
CREATE TABLE tpcc_run_int
(
run_name VARCHAR2(20),
interval NUMBER,
interval_count NUMBER,
response_time NUMBER,
think_time NUMBER
);
rem
rem throughput of new order transactions
rem
CREATE TABLE bench_run_int
(
run_name VARCHAR2(20),
proc_no NUMBER,
interval NUMBER,
interval_count NUMBER,
response_time NUMBER,
think_time NUMBER
)
partition by range (proc_no) (
tablespace restbl_0 values less than ( 120 )
tablespace restbl_0,
partition restbl_1 values less than ( 240 )
tablespace restbl_1,
partition restbl_2 values less than ( 360 )
tablespace restbl_2,
partition restbl_3 values less than ( 480 )
tablespace restbl_3,
partition restbl_4 values less than ( 600 )
tablespace restbl_4,
partition restbl_5 values less than ( 720 )
tablespace restbl_5,
partition restbl_6 values less than ( 840 )
tablespace restbl_6,
partition restbl_7 values less than ( 960 )
tablespace restbl_7,
partition restbl_8 values less than ( 1080 )
tablespace restbl_8,

```

```

partition restbl0_9 values less than ( 1200 )
tablespace restbl_9,
partition restbl0_10 values less than
( 1320 ) tablespace restbl_10,
partition restbl0_11 values less than
( 1440 ) tablespace restbl_11,
partition restbl0_12 values less than
( 1560 ) tablespace restbl_12,
partition restbl0_13 values less than
( 1680 ) tablespace restbl_13,
partition restbl0_14 values less than
( 1800 ) tablespace restbl_14,
partition restbl0_15 values less than
( 1920 ) tablespace restbl_15,
partition restbl0_16 values less than
( 2040 ) tablespace restbl_16,
partition restbl0_17 values less than
( 2160 ) tablespace restbl_17,
partition restbl0_18 values less than
( 2280 ) tablespace restbl_18,
partition restbl0_19 values less than
( maxvalue ) tablespace restbl_19
);

```

rem
rem Results from delivery servers
rem

```

CREATE TABLE tpcc_back_res
(
run_name VARCHAR2(20),
in_timing_int NUMBER,
fast NUMBER,
resp_time NUMBER,
retries NUMBER
);

```

rem
rem Aggregate results for all generators.
rem These results are from the measurement interval only.
rem These results are used to calculate the TPS rate over
rem the measurement interval.
rem

```

CREATE TABLE tpcc_user_res
(
run_name VARCHAR2(20),
no_men NUMBER,
fast_men NUMBER,
in_flight_men NUMBER,
retry_men NUMBER,
min_time_men NUMBER,
max_time_men NUMBER,
sum_time_men NUMBER,
ninety_per_men NUMBER,
think_min_men NUMBER,
think_max_men NUMBER,
think_sum_men NUMBER,
key_min_men NUMBER,
key_max_men NUMBER,
key_sum_men NUMBER,
no_new NUMBER,
fast_new NUMBER,
in_flight_new NUMBER,
retry_new NUMBER,
min_time_new NUMBER,
max_time_new NUMBER,
sum_time_new NUMBER,
ninety_per_new NUMBER,
think_min_new NUMBER,
think_max_new NUMBER,

```

```

think_sum_new NUMBER,
key_min_new NUMBER,
key_max_new NUMBER,
key_sum_new NUMBER,
remote_new NUMBER,
rollback_new NUMBER,
sum_of_new NUMBER,
remote_of_new NUMBER,
allrollback_new NUMBER,
no_pay NUMBER,
fast_pay NUMBER,
in_flight_pay NUMBER,
retry_pay NUMBER,
min_time_pay NUMBER,
max_time_pay NUMBER,
sum_time_pay NUMBER,
ninety_per_pay NUMBER,
think_min_pay NUMBER,
think_max_pay NUMBER,
think_sum_pay NUMBER,
key_min_pay NUMBER,
key_max_pay NUMBER,
key_sum_pay NUMBER,
remote_pay NUMBER,
bylast_pay NUMBER,
no_ord NUMBER,
fast_ord NUMBER,
in_flight_ord NUMBER,
retry_ord NUMBER,
min_time_ord NUMBER,
max_time_ord NUMBER,
sum_time_ord NUMBER,
ninety_per_ord NUMBER,
think_min_ord NUMBER,
think_max_ord NUMBER,
think_sum_ord NUMBER,
key_min_ord NUMBER,
key_max_ord NUMBER,
key_sum_ord NUMBER,
bylast_ord NUMBER,
no_del NUMBER,
fast_del NUMBER,
in_flight_del NUMBER,
retry_del NUMBER,
min_time_del NUMBER,
max_time_del NUMBER,
sum_time_del NUMBER,
ninety_per_del NUMBER,
think_min_del NUMBER,
think_max_del NUMBER,
think_sum_del NUMBER,
key_min_del NUMBER,
key_max_del NUMBER,
key_sum_del NUMBER,
no_sto NUMBER,
fast_sto NUMBER,
in_flight_sto NUMBER,
retry_sto NUMBER,
min_time_sto NUMBER,
max_time_sto NUMBER,
sum_time_sto NUMBER,
ninety_per_sto NUMBER,
think_min_sto NUMBER,
think_max_sto NUMBER,
think_sum_sto NUMBER,
key_min_sto NUMBER,
key_max_sto NUMBER,
key_sum_sto NUMBER,
cpu_time NUMBER,

```

```

deadlocks NUMBER
);

```

rem
rem Results from individual generators.
rem These results are from the measurement interval only.
rem These results are used to calculate the TPS rate over
rem the measurement interval.
rem

```

CREATE TABLE bench_user_res
(
run_name VARCHAR2(20),
audit_str VARCHAR2(10),
proc_no NUMBER,
hid NUMBER,
no_men NUMBER,
fast_men NUMBER,
in_flight_men NUMBER,
retry_men NUMBER,
min_time_men NUMBER,
max_time_men NUMBER,
sum_time_men NUMBER,
ninety_per_men NUMBER,
think_min_men NUMBER,
think_max_men NUMBER,
think_sum_men NUMBER,
key_min_men NUMBER,
key_max_men NUMBER,
key_sum_men NUMBER,
no_new NUMBER,
fast_new NUMBER,
in_flight_new NUMBER,
retry_new NUMBER,
min_time_new NUMBER,
max_time_new NUMBER,
sum_time_new NUMBER,
ninety_per_new NUMBER,
think_min_new NUMBER,
think_max_new NUMBER,
think_sum_new NUMBER,
key_min_new NUMBER,
key_max_new NUMBER,
key_sum_new NUMBER,
remote_new NUMBER,
rollback_new NUMBER,
sum_of_new NUMBER,
remote_of_new NUMBER,
allrollback_new NUMBER,
no_pay NUMBER,
fast_pay NUMBER,
in_flight_pay NUMBER,
retry_pay NUMBER,
min_time_pay NUMBER,
max_time_pay NUMBER,
sum_time_pay NUMBER,
ninety_per_pay NUMBER,
think_min_pay NUMBER,
think_max_pay NUMBER,
think_sum_pay NUMBER,
key_min_pay NUMBER,
key_max_pay NUMBER,
key_sum_pay NUMBER,
remote_pay NUMBER,
bylast_pay NUMBER,
no_ord NUMBER,
fast_ord NUMBER,
in_flight_ord NUMBER,

```

```

retry_ord NUMBER,
min_time_ord NUMBER,
max_time_ord NUMBER,
sum_time_ord NUMBER,
ninety_per_ord NUMBER,
think_min_ord NUMBER,
think_max_ord NUMBER,
think_sum_ord NUMBER,
key_min_ord NUMBER,
key_max_ord NUMBER,
key_sum_ord NUMBER,
bylast_ord NUMBER,
no_del NUMBER,
fast_del NUMBER,
in_flight_del NUMBER,
retry_del NUMBER,
min_time_del NUMBER,
max_time_del NUMBER,
sum_time_del NUMBER,
ninety_per_del NUMBER,
think_min_del NUMBER,
think_max_del NUMBER,
think_sum_del NUMBER,
key_min_del NUMBER,
key_max_del NUMBER,
key_sum_del NUMBER,
no_sto NUMBER,
fast_sto NUMBER,
in_flight_sto NUMBER,
retry_sto NUMBER,
min_time_sto NUMBER,
max_time_sto NUMBER,
sum_time_sto NUMBER,
ninety_per_sto NUMBER,
think_min_sto NUMBER,
think_max_sto NUMBER,
think_sum_sto NUMBER,
key_min_sto NUMBER,
key_max_sto NUMBER,
key_sum_sto NUMBER,
cpu_time NUMBER,
deadlocks NUMBER
)
partition by range (proc_no) (
partition restbl1_0 values less than ( 120 )
tablespace restbl_0,
partition restbl1_1 values less than ( 240 )
tablespace restbl_1,
partition restbl1_2 values less than ( 360 )
tablespace restbl_2,
partition restbl1_3 values less than ( 480 )
tablespace restbl_3,
partition restbl1_4 values less than ( 600 )
tablespace restbl_4,
partition restbl1_5 values less than ( 720 )
tablespace restbl_5,
partition restbl1_6 values less than ( 840 )
tablespace restbl_6,
partition restbl1_7 values less than ( 960 )
tablespace restbl_7,
partition restbl1_8 values less than ( 1080 )
tablespace restbl_8,
partition restbl1_9 values less than ( 1200 )
tablespace restbl_9,
partition restbl1_10 values less than
( 1320 ) tablespace restbl_10,
partition restbl1_11 values less than
( 1440 ) tablespace restbl_11,

```

```

partition restbl1_12 values less than
( 1560 ) tablespace restbl_12,
partition restbl1_13 values less than
( 1680 ) tablespace restbl_13,
partition restbl1_14 values less than
( 1800 ) tablespace restbl_14,
partition restbl1_15 values less than
( 1920 ) tablespace restbl_15,
partition restbl1_16 values less than
( 2040 ) tablespace restbl_16,
partition restbl1_17 values less than
( 2160 ) tablespace restbl_17,
partition restbl1_18 values less than
( 2280 ) tablespace restbl_18,
partition restbl1_19 values less than
( maxvalue ) tablespace restbl_19
);

rem
rem Aggregate results for generators on each
rem host.
rem These results are from the measurement
rem interval only.
rem These results are used to calculate the TPM
rem rate over
rem the measurement interval.
rem
CREATE TABLE tpcc_tpm
(
run_name VARCHAR2(20),
hid NUMBER,
no_new NUMBER
);

rem
rem Aggregate results for new order transactions.
rem These results are from the measurement
rem interval only.
rem
CREATE TABLE tpcc_new_res
(
run_name VARCHAR2(20),
rep1 NUMBER,
rep2 NUMBER,
rep3 NUMBER,
rep4 NUMBER,
rep5 NUMBER,
rep6 NUMBER,
rep7 NUMBER,
rep8 NUMBER,
rep9 NUMBER,
rep10 NUMBER,
rep11 NUMBER,
rep12 NUMBER,
rep13 NUMBER,
rep14 NUMBER,
rep15 NUMBER,
rep16 NUMBER,
rep17 NUMBER,
rep18 NUMBER,
rep19 NUMBER,
rep20 NUMBER,
rep21 NUMBER,
rep22 NUMBER,
rep23 NUMBER,
rep24 NUMBER,
rep25 NUMBER,
rep26 NUMBER,
rep27 NUMBER,

```

```

rep28 NUMBER,
rep29 NUMBER,
rep30 NUMBER,
rep31 NUMBER,
rep32 NUMBER,
rep33 NUMBER,
rep34 NUMBER,
rep35 NUMBER,
rep36 NUMBER,
rep37 NUMBER,
rep38 NUMBER,
rep39 NUMBER,
rep40 NUMBER,
rep41 NUMBER,
rep42 NUMBER,
rep43 NUMBER,
rep44 NUMBER,
rep45 NUMBER,
rep46 NUMBER,
rep47 NUMBER,
rep48 NUMBER,
rep49 NUMBER,
rep50 NUMBER,
rep51 NUMBER,
rep52 NUMBER,
rep53 NUMBER,
rep54 NUMBER,
rep55 NUMBER,
rep56 NUMBER,
rep57 NUMBER,
rep58 NUMBER,
rep59 NUMBER,
rep60 NUMBER,
rep61 NUMBER,
rep62 NUMBER,
rep63 NUMBER,
rep64 NUMBER,
rep65 NUMBER,
rep66 NUMBER,
rep67 NUMBER,
rep68 NUMBER,
rep69 NUMBER,
rep70 NUMBER,
rep71 NUMBER,
rep72 NUMBER,
rep73 NUMBER,
rep74 NUMBER,
rep75 NUMBER,
rep76 NUMBER,
rep77 NUMBER,
rep78 NUMBER,
rep79 NUMBER,
rep80 NUMBER,
rep81 NUMBER,
rep82 NUMBER,
rep83 NUMBER,
rep84 NUMBER,
rep85 NUMBER,
rep86 NUMBER,
rep87 NUMBER,
rep88 NUMBER,
rep89 NUMBER,
rep90 NUMBER,
rep91 NUMBER,
rep92 NUMBER,
rep93 NUMBER,
rep94 NUMBER,
rep95 NUMBER,
rep96 NUMBER,

```

```

rep97    NUMBER,
rep98    NUMBER,
rep99    NUMBER,
rep100   NUMBER,
thk1     NUMBER,
thk2     NUMBER,
thk3     NUMBER,
thk4     NUMBER,
thk5     NUMBER,
thk6     NUMBER,
thk7     NUMBER,
thk8     NUMBER,
thk9     NUMBER,
thk10    NUMBER,
thk11    NUMBER,
thk12    NUMBER,
thk13    NUMBER,
thk14    NUMBER,
thk15    NUMBER,
thk16    NUMBER,
thk17    NUMBER,
thk18    NUMBER,
thk19    NUMBER,
thk20    NUMBER,
thk21    NUMBER,
thk22    NUMBER,
thk23    NUMBER,
thk24    NUMBER,
thk25    NUMBER,
key1     NUMBER,
key2     NUMBER,
key3     NUMBER,
key4     NUMBER,
key5     NUMBER,
key6     NUMBER,
key7     NUMBER,
key8     NUMBER,
key9     NUMBER,
key10    NUMBER
);
rem
rem Results for new order transactions.
rem These results are from the measurement
rem interval only.
rem
CREATE TABLE bench_new_res
(
  run_name    VARCHAR2(20),
  audit_str   VARCHAR2(10),
  proc_no     NUMBER,
  rep1        NUMBER,
  rep2        NUMBER,
  rep3        NUMBER,
  rep4        NUMBER,
  rep5        NUMBER,
  rep6        NUMBER,
  rep7        NUMBER,
  rep8        NUMBER,
  rep9        NUMBER,
  rep10       NUMBER,
  rep11       NUMBER,
  rep12       NUMBER,
  rep13       NUMBER,
  rep14       NUMBER,
  rep15       NUMBER,
  rep16       NUMBER,
  rep17       NUMBER,
  rep18       NUMBER,
  rep19       NUMBER,

```

```

rep20    NUMBER,
rep21    NUMBER,
rep22    NUMBER,
rep23    NUMBER,
rep24    NUMBER,
rep25    NUMBER,
rep26    NUMBER,
rep27    NUMBER,
rep28    NUMBER,
rep29    NUMBER,
rep30    NUMBER,
rep31    NUMBER,
rep32    NUMBER,
rep33    NUMBER,
rep34    NUMBER,
rep35    NUMBER,
rep36    NUMBER,
rep37    NUMBER,
rep38    NUMBER,
rep39    NUMBER,
rep40    NUMBER,
rep41    NUMBER,
rep42    NUMBER,
rep43    NUMBER,
rep44    NUMBER,
rep45    NUMBER,
rep46    NUMBER,
rep47    NUMBER,
rep48    NUMBER,
rep49    NUMBER,
rep50    NUMBER,
rep51    NUMBER,
rep52    NUMBER,
rep53    NUMBER,
rep54    NUMBER,
rep55    NUMBER,
rep56    NUMBER,
rep57    NUMBER,
rep58    NUMBER,
rep59    NUMBER,
rep60    NUMBER,
rep61    NUMBER,
rep62    NUMBER,
rep63    NUMBER,
rep64    NUMBER,
rep65    NUMBER,
rep66    NUMBER,
rep67    NUMBER,
rep68    NUMBER,
rep69    NUMBER,
rep70    NUMBER,
rep71    NUMBER,
rep72    NUMBER,
rep73    NUMBER,
rep74    NUMBER,
rep75    NUMBER,
rep76    NUMBER,
rep77    NUMBER,
rep78    NUMBER,
rep79    NUMBER,
rep80    NUMBER,
rep81    NUMBER,
rep82    NUMBER,
rep83    NUMBER,
rep84    NUMBER,
rep85    NUMBER,
rep86    NUMBER,
rep87    NUMBER,
rep88    NUMBER,

```

```

rep89    NUMBER,
rep90    NUMBER,
rep91    NUMBER,
rep92    NUMBER,
rep93    NUMBER,
rep94    NUMBER,
rep95    NUMBER,
rep96    NUMBER,
rep97    NUMBER,
rep98    NUMBER,
rep99    NUMBER,
rep100   NUMBER,
thk1     NUMBER,
thk2     NUMBER,
thk3     NUMBER,
thk4     NUMBER,
thk5     NUMBER,
thk6     NUMBER,
thk7     NUMBER,
thk8     NUMBER,
thk9     NUMBER,
thk10    NUMBER,
thk11    NUMBER,
thk12    NUMBER,
thk13    NUMBER,
thk14    NUMBER,
thk15    NUMBER,
thk16    NUMBER,
thk17    NUMBER,
thk18    NUMBER,
thk19    NUMBER,
thk20    NUMBER,
thk21    NUMBER,
thk22    NUMBER,
thk23    NUMBER,
thk24    NUMBER,
thk25    NUMBER,
key1     NUMBER,
key2     NUMBER,
key3     NUMBER,
key4     NUMBER,
key5     NUMBER,
key6     NUMBER,
key7     NUMBER,
key8     NUMBER,
key9     NUMBER,
key10    NUMBER
)
partition by range (proc_no) (
  partition restbl2_0 values less than ( 120)
  tablespace restbl_0,
  partition restbl2_1 values less than ( 240)
  tablespace restbl_1,
  partition restbl2_2 values less than ( 360)
  tablespace restbl_2,
  partition restbl2_3 values less than ( 480)
  tablespace restbl_3,
  partition restbl2_4 values less than ( 600)
  tablespace restbl_4,
  partition restbl2_5 values less than ( 720)
  tablespace restbl_5,
  partition restbl2_6 values less than ( 840)
  tablespace restbl_6,
  partition restbl2_7 values less than ( 960)
  tablespace restbl_7,
  partition restbl2_8 values less than (1080)
  tablespace restbl_8,
  partition restbl2_9 values less than (1200)
  tablespace restbl_9,

```

```

    partition restbl2_10 values less than
( 1320 ) tablespace restbl_10,
    partition restbl2_11 values less than
( 1440 ) tablespace restbl_11,
    partition restbl2_12 values less than
( 1560 ) tablespace restbl_12,
    partition restbl2_13 values less than
( 1680 ) tablespace restbl_13,
    partition restbl2_14 values less than
( 1800 ) tablespace restbl_14,
    partition restbl2_15 values less than
( 1920 ) tablespace restbl_15,
    partition restbl2_16 values less than
( 2040 ) tablespace restbl_16,
    partition restbl2_17 values less than
( 2160 ) tablespace restbl_17,
    partition restbl2_18 values less than
( 2280 ) tablespace restbl_18,
    partition restbl2_19 values less than
( maxvalue ) tablespace restbl_19
);

rem
rem Aggregate results for payment transactions.
rem These results are from the measurement
rem interval only.
rem
CREATE TABLE tpcc_pay_res
(
    run_name    VARCHAR2(20),
    rep1        NUMBER,
    rep2        NUMBER,
    rep3        NUMBER,
    rep4        NUMBER,
    rep5        NUMBER,
    rep6        NUMBER,
    rep7        NUMBER,
    rep8        NUMBER,
    rep9        NUMBER,
    rep10       NUMBER,
    rep11       NUMBER,
    rep12       NUMBER,
    rep13       NUMBER,
    rep14       NUMBER,
    rep15       NUMBER,
    rep16       NUMBER,
    rep17       NUMBER,
    rep18       NUMBER,
    rep19       NUMBER,
    rep20       NUMBER,
    rep21       NUMBER,
    rep22       NUMBER,
    rep23       NUMBER,
    rep24       NUMBER,
    rep25       NUMBER,
    rep26       NUMBER,
    rep27       NUMBER,
    rep28       NUMBER,
    rep29       NUMBER,
    rep30       NUMBER,
    rep31       NUMBER,
    rep32       NUMBER,
    rep33       NUMBER,
    rep34       NUMBER,
    rep35       NUMBER,
    rep36       NUMBER,
    rep37       NUMBER,
    rep38       NUMBER,
    rep39       NUMBER,

```

```

rep40        NUMBER,
rep41        NUMBER,
rep42        NUMBER,
rep43        NUMBER,
rep44        NUMBER,
rep45        NUMBER,
rep46        NUMBER,
rep47        NUMBER,
rep48        NUMBER,
rep49        NUMBER,
rep50        NUMBER,
rep51        NUMBER,
rep52        NUMBER,
rep53        NUMBER,
rep54        NUMBER,
rep55        NUMBER,
rep56        NUMBER,
rep57        NUMBER,
rep58        NUMBER,
rep59        NUMBER,
rep60        NUMBER,
rep61        NUMBER,
rep62        NUMBER,
rep63        NUMBER,
rep64        NUMBER,
rep65        NUMBER,
rep66        NUMBER,
rep67        NUMBER,
rep68        NUMBER,
rep69        NUMBER,
rep70        NUMBER,
rep71        NUMBER,
rep72        NUMBER,
rep73        NUMBER,
rep74        NUMBER,
rep75        NUMBER,
rep76        NUMBER,
rep77        NUMBER,
rep78        NUMBER,
rep79        NUMBER,
rep80        NUMBER,
rep81        NUMBER,
rep82        NUMBER,
rep83        NUMBER,
rep84        NUMBER,
rep85        NUMBER,
rep86        NUMBER,
rep87        NUMBER,
rep88        NUMBER,
rep89        NUMBER,
rep90        NUMBER,
rep91        NUMBER,
rep92        NUMBER,
rep93        NUMBER,
rep94        NUMBER,
rep95        NUMBER,
rep96        NUMBER,
rep97        NUMBER,
rep98        NUMBER,
rep99        NUMBER,
rep100       NUMBER,
thk1         NUMBER,
thk2         NUMBER,
thk3         NUMBER,
thk4         NUMBER,
thk5         NUMBER,
thk6         NUMBER,
thk7         NUMBER,
thk8         NUMBER,

```

```

thk9         NUMBER,
thk10        NUMBER,
thk11        NUMBER,
thk12        NUMBER,
thk13        NUMBER,
thk14        NUMBER,
thk15        NUMBER,
thk16        NUMBER,
thk17        NUMBER,
thk18        NUMBER,
thk19        NUMBER,
thk20        NUMBER,
thk21        NUMBER,
thk22        NUMBER,
thk23        NUMBER,
thk24        NUMBER,
thk25        NUMBER,
key1         NUMBER,
key2         NUMBER,
key3         NUMBER,
key4         NUMBER,
key5         NUMBER,
key6         NUMBER,
key7         NUMBER,
key8         NUMBER,
key9         NUMBER,
key10        NUMBER
);

rem
rem Results for payment transactions.
rem These results are from the measurement
rem interval only.
rem
CREATE TABLE bench_pay_res
(
    run_name    VARCHAR2(20),
    audit_str   VARCHAR2(10),
    proc_no     NUMBER,
    rep1        NUMBER,
    rep2        NUMBER,
    rep3        NUMBER,
    rep4        NUMBER,
    rep5        NUMBER,
    rep6        NUMBER,
    rep7        NUMBER,
    rep8        NUMBER,
    rep9        NUMBER,
    rep10       NUMBER,
    rep11       NUMBER,
    rep12       NUMBER,
    rep13       NUMBER,
    rep14       NUMBER,
    rep15       NUMBER,
    rep16       NUMBER,
    rep17       NUMBER,
    rep18       NUMBER,
    rep19       NUMBER,
    rep20       NUMBER,
    rep21       NUMBER,
    rep22       NUMBER,
    rep23       NUMBER,
    rep24       NUMBER,
    rep25       NUMBER,
    rep26       NUMBER,
    rep27       NUMBER,
    rep28       NUMBER,
    rep29       NUMBER,
    rep30       NUMBER,
    rep31       NUMBER,

```

```

rep32 NUMBER,
rep33 NUMBER,
rep34 NUMBER,
rep35 NUMBER,
rep36 NUMBER,
rep37 NUMBER,
rep38 NUMBER,
rep39 NUMBER,
rep40 NUMBER,
rep41 NUMBER,
rep42 NUMBER,
rep43 NUMBER,
rep44 NUMBER,
rep45 NUMBER,
rep46 NUMBER,
rep47 NUMBER,
rep48 NUMBER,
rep49 NUMBER,
rep50 NUMBER,
rep51 NUMBER,
rep52 NUMBER,
rep53 NUMBER,
rep54 NUMBER,
rep55 NUMBER,
rep56 NUMBER,
rep57 NUMBER,
rep58 NUMBER,
rep59 NUMBER,
rep60 NUMBER,
rep61 NUMBER,
rep62 NUMBER,
rep63 NUMBER,
rep64 NUMBER,
rep65 NUMBER,
rep66 NUMBER,
rep67 NUMBER,
rep68 NUMBER,
rep69 NUMBER,
rep70 NUMBER,
rep71 NUMBER,
rep72 NUMBER,
rep73 NUMBER,
rep74 NUMBER,
rep75 NUMBER,
rep76 NUMBER,
rep77 NUMBER,
rep78 NUMBER,
rep79 NUMBER,
rep80 NUMBER,
rep81 NUMBER,
rep82 NUMBER,
rep83 NUMBER,
rep84 NUMBER,
rep85 NUMBER,
rep86 NUMBER,
rep87 NUMBER,
rep88 NUMBER,
rep89 NUMBER,
rep90 NUMBER,
rep91 NUMBER,
rep92 NUMBER,
rep93 NUMBER,
rep94 NUMBER,
rep95 NUMBER,
rep96 NUMBER,
rep97 NUMBER,
rep98 NUMBER,
rep99 NUMBER,
rep100 NUMBER,

```

```

thk1 NUMBER,
thk2 NUMBER,
thk3 NUMBER,
thk4 NUMBER,
thk5 NUMBER,
thk6 NUMBER,
thk7 NUMBER,
thk8 NUMBER,
thk9 NUMBER,
thk10 NUMBER,
thk11 NUMBER,
thk12 NUMBER,
thk13 NUMBER,
thk14 NUMBER,
thk15 NUMBER,
thk16 NUMBER,
thk17 NUMBER,
thk18 NUMBER,
thk19 NUMBER,
thk20 NUMBER,
thk21 NUMBER,
thk22 NUMBER,
thk23 NUMBER,
thk24 NUMBER,
thk25 NUMBER,
key1 NUMBER,
key2 NUMBER,
key3 NUMBER,
key4 NUMBER,
key5 NUMBER,
key6 NUMBER,
key7 NUMBER,
key8 NUMBER,
key9 NUMBER,
key10 NUMBER
)
partition by range (proc_no) (
partition restbl3_0 values less than ( 120 )
tablespace restbl_0,
partition restbl3_1 values less than ( 240 )
tablespace restbl_1,
partition restbl3_2 values less than ( 360 )
tablespace restbl_2,
partition restbl3_3 values less than ( 480 )
tablespace restbl_3,
partition restbl3_4 values less than ( 600 )
tablespace restbl_4,
partition restbl3_5 values less than ( 720 )
tablespace restbl_5,
partition restbl3_6 values less than ( 840 )
tablespace restbl_6,
partition restbl3_7 values less than ( 960 )
tablespace restbl_7,
partition restbl3_8 values less than ( 1080 )
tablespace restbl_8,
partition restbl3_9 values less than ( 1200 )
tablespace restbl_9,
partition restbl3_10 values less than
( 1320 ) tablespace restbl_10,
partition restbl3_11 values less than
( 1440 ) tablespace restbl_11,
partition restbl3_12 values less than
( 1560 ) tablespace restbl_12,
partition restbl3_13 values less than
( 1680 ) tablespace restbl_13,
partition restbl3_14 values less than
( 1800 ) tablespace restbl_14,
partition restbl3_15 values less than
( 1920 ) tablespace restbl_15,

```

```

partition restbl3_16 values less than
( 2040 ) tablespace restbl_16,
partition restbl3_17 values less than
( 2160 ) tablespace restbl_17,
partition restbl3_18 values less than
( 2280 ) tablespace restbl_18,
partition restbl3_19 values less than
( maxvalue ) tablespace restbl_19
);
rem
rem Aggregate results for order status
rem transactions.
rem These results are from the measurement
rem interval only.
rem
CREATE TABLE tpcc_ord_res
(
run_name VARCHAR2(20),
rep1 NUMBER,
rep2 NUMBER,
rep3 NUMBER,
rep4 NUMBER,
rep5 NUMBER,
rep6 NUMBER,
rep7 NUMBER,
rep8 NUMBER,
rep9 NUMBER,
rep10 NUMBER,
rep11 NUMBER,
rep12 NUMBER,
rep13 NUMBER,
rep14 NUMBER,
rep15 NUMBER,
rep16 NUMBER,
rep17 NUMBER,
rep18 NUMBER,
rep19 NUMBER,
rep20 NUMBER,
rep21 NUMBER,
rep22 NUMBER,
rep23 NUMBER,
rep24 NUMBER,
rep25 NUMBER,
rep26 NUMBER,
rep27 NUMBER,
rep28 NUMBER,
rep29 NUMBER,
rep30 NUMBER,
rep31 NUMBER,
rep32 NUMBER,
rep33 NUMBER,
rep34 NUMBER,
rep35 NUMBER,
rep36 NUMBER,
rep37 NUMBER,
rep38 NUMBER,
rep39 NUMBER,
rep40 NUMBER,
rep41 NUMBER,
rep42 NUMBER,
rep43 NUMBER,
rep44 NUMBER,
rep45 NUMBER,
rep46 NUMBER,
rep47 NUMBER,
rep48 NUMBER,
rep49 NUMBER,
rep50 NUMBER,
rep51 NUMBER,

```

```

rep52 NUMBER,
rep53 NUMBER,
rep54 NUMBER,
rep55 NUMBER,
rep56 NUMBER,
rep57 NUMBER,
rep58 NUMBER,
rep59 NUMBER,
rep60 NUMBER,
rep61 NUMBER,
rep62 NUMBER,
rep63 NUMBER,
rep64 NUMBER,
rep65 NUMBER,
rep66 NUMBER,
rep67 NUMBER,
rep68 NUMBER,
rep69 NUMBER,
rep70 NUMBER,
rep71 NUMBER,
rep72 NUMBER,
rep73 NUMBER,
rep74 NUMBER,
rep75 NUMBER,
rep76 NUMBER,
rep77 NUMBER,
rep78 NUMBER,
rep79 NUMBER,
rep80 NUMBER,
rep81 NUMBER,
rep82 NUMBER,
rep83 NUMBER,
rep84 NUMBER,
rep85 NUMBER,
rep86 NUMBER,
rep87 NUMBER,
rep88 NUMBER,
rep89 NUMBER,
rep90 NUMBER,
rep91 NUMBER,
rep92 NUMBER,
rep93 NUMBER,
rep94 NUMBER,
rep95 NUMBER,
rep96 NUMBER,
rep97 NUMBER,
rep98 NUMBER,
rep99 NUMBER,
rep100 NUMBER,
thk1 NUMBER,
thk2 NUMBER,
thk3 NUMBER,
thk4 NUMBER,
thk5 NUMBER,
thk6 NUMBER,
thk7 NUMBER,
thk8 NUMBER,
thk9 NUMBER,
thk10 NUMBER,
thk11 NUMBER,
thk12 NUMBER,
thk13 NUMBER,
thk14 NUMBER,
thk15 NUMBER,
thk16 NUMBER,
thk17 NUMBER,
thk18 NUMBER,
thk19 NUMBER,
thk20 NUMBER,

```

```

thk21 NUMBER,
thk22 NUMBER,
thk23 NUMBER,
thk24 NUMBER,
thk25 NUMBER,
key1 NUMBER,
key2 NUMBER,
key3 NUMBER,
key4 NUMBER,
key5 NUMBER,
key6 NUMBER,
key7 NUMBER,
key8 NUMBER,
key9 NUMBER,
key10 NUMBER
);
rem
rem Results for order status transactions.
rem These results are from the measurement
rem interval only.
rem
CREATE TABLE bench_ord_res
(
run_name VARCHAR2(20),
audit_str VARCHAR2(10),
proc_no NUMBER,
rep1 NUMBER,
rep2 NUMBER,
rep3 NUMBER,
rep4 NUMBER,
rep5 NUMBER,
rep6 NUMBER,
rep7 NUMBER,
rep8 NUMBER,
rep9 NUMBER,
rep10 NUMBER,
rep11 NUMBER,
rep12 NUMBER,
rep13 NUMBER,
rep14 NUMBER,
rep15 NUMBER,
rep16 NUMBER,
rep17 NUMBER,
rep18 NUMBER,
rep19 NUMBER,
rep20 NUMBER,
rep21 NUMBER,
rep22 NUMBER,
rep23 NUMBER,
rep24 NUMBER,
rep25 NUMBER,
rep26 NUMBER,
rep27 NUMBER,
rep28 NUMBER,
rep29 NUMBER,
rep30 NUMBER,
rep31 NUMBER,
rep32 NUMBER,
rep33 NUMBER,
rep34 NUMBER,
rep35 NUMBER,
rep36 NUMBER,
rep37 NUMBER,
rep38 NUMBER,
rep39 NUMBER,
rep40 NUMBER,
rep41 NUMBER,
rep42 NUMBER,
rep43 NUMBER,

```

```

rep44 NUMBER,
rep45 NUMBER,
rep46 NUMBER,
rep47 NUMBER,
rep48 NUMBER,
rep49 NUMBER,
rep50 NUMBER,
rep51 NUMBER,
rep52 NUMBER,
rep53 NUMBER,
rep54 NUMBER,
rep55 NUMBER,
rep56 NUMBER,
rep57 NUMBER,
rep58 NUMBER,
rep59 NUMBER,
rep60 NUMBER,
rep61 NUMBER,
rep62 NUMBER,
rep63 NUMBER,
rep64 NUMBER,
rep65 NUMBER,
rep66 NUMBER,
rep67 NUMBER,
rep68 NUMBER,
rep69 NUMBER,
rep70 NUMBER,
rep71 NUMBER,
rep72 NUMBER,
rep73 NUMBER,
rep74 NUMBER,
rep75 NUMBER,
rep76 NUMBER,
rep77 NUMBER,
rep78 NUMBER,
rep79 NUMBER,
rep80 NUMBER,
rep81 NUMBER,
rep82 NUMBER,
rep83 NUMBER,
rep84 NUMBER,
rep85 NUMBER,
rep86 NUMBER,
rep87 NUMBER,
rep88 NUMBER,
rep89 NUMBER,
rep90 NUMBER,
rep91 NUMBER,
rep92 NUMBER,
rep93 NUMBER,
rep94 NUMBER,
rep95 NUMBER,
rep96 NUMBER,
rep97 NUMBER,
rep98 NUMBER,
rep99 NUMBER,
rep100 NUMBER,
thk1 NUMBER,
thk2 NUMBER,
thk3 NUMBER,
thk4 NUMBER,
thk5 NUMBER,
thk6 NUMBER,
thk7 NUMBER,
thk8 NUMBER,
thk9 NUMBER,
thk10 NUMBER,
thk11 NUMBER,
thk12 NUMBER,

```

```

thk13    NUMBER,
thk14    NUMBER,
thk15    NUMBER,
thk16    NUMBER,
thk17    NUMBER,
thk18    NUMBER,
thk19    NUMBER,
thk20    NUMBER,
thk21    NUMBER,
thk22    NUMBER,
thk23    NUMBER,
thk24    NUMBER,
thk25    NUMBER,
key1     NUMBER,
key2     NUMBER,
key3     NUMBER,
key4     NUMBER,
key5     NUMBER,
key6     NUMBER,
key7     NUMBER,
key8     NUMBER,
key9     NUMBER,
key10    NUMBER
)
partition by range (proc_no) (
partition restbl4_0 values less than ( 120)
tablespace restbl_0,
partition restbl4_1 values less than ( 240)
tablespace restbl_1,
partition restbl4_2 values less than ( 360)
tablespace restbl_2,
partition restbl4_3 values less than ( 480)
tablespace restbl_3,
partition restbl4_4 values less than ( 600)
tablespace restbl_4,
partition restbl4_5 values less than ( 720)
tablespace restbl_5,
partition restbl4_6 values less than ( 840)
tablespace restbl_6,
partition restbl4_7 values less than ( 960)
tablespace restbl_7,
partition restbl4_8 values less than ( 1080)
tablespace restbl_8,
partition restbl4_9 values less than ( 1200)
tablespace restbl_9,
partition restbl4_10 values less than
( 1320) tablespace restbl_10,
partition restbl4_11 values less than
( 1440) tablespace restbl_11,
partition restbl4_12 values less than
( 1560) tablespace restbl_12,
partition restbl4_13 values less than
( 1680) tablespace restbl_13,
partition restbl4_14 values less than
( 1800) tablespace restbl_14,
partition restbl4_15 values less than
( 1920) tablespace restbl_15,
partition restbl4_16 values less than
( 2040) tablespace restbl_16,
partition restbl4_17 values less than
( 2160) tablespace restbl_17,
partition restbl4_18 values less than
( 2280) tablespace restbl_18,
partition restbl4_19 values less than
(maxvalue) tablespace restbl_19
);
rem
rem Aggregate results for delivery transactions.

```

```

rem These results are from the measurement
interval only.

```

```
rem
```

```

CREATE TABLE tpcc_del_res
(
run_name  VARCHAR2(20),
rep1     NUMBER,
rep2     NUMBER,
rep3     NUMBER,
rep4     NUMBER,
rep5     NUMBER,
rep6     NUMBER,
rep7     NUMBER,
rep8     NUMBER,
rep9     NUMBER,
rep10    NUMBER,
rep11    NUMBER,
rep12    NUMBER,
rep13    NUMBER,
rep14    NUMBER,
rep15    NUMBER,
rep16    NUMBER,
rep17    NUMBER,
rep18    NUMBER,
rep19    NUMBER,
rep20    NUMBER,
rep21    NUMBER,
rep22    NUMBER,
rep23    NUMBER,
rep24    NUMBER,
rep25    NUMBER,
rep26    NUMBER,
rep27    NUMBER,
rep28    NUMBER,
rep29    NUMBER,
rep30    NUMBER,
rep31    NUMBER,
rep32    NUMBER,
rep33    NUMBER,
rep34    NUMBER,
rep35    NUMBER,
rep36    NUMBER,
rep37    NUMBER,
rep38    NUMBER,
rep39    NUMBER,
rep40    NUMBER,
rep41    NUMBER,
rep42    NUMBER,
rep43    NUMBER,
rep44    NUMBER,
rep45    NUMBER,
rep46    NUMBER,
rep47    NUMBER,
rep48    NUMBER,
rep49    NUMBER,
rep50    NUMBER,
rep51    NUMBER,
rep52    NUMBER,
rep53    NUMBER,
rep54    NUMBER,
rep55    NUMBER,
rep56    NUMBER,
rep57    NUMBER,
rep58    NUMBER,
rep59    NUMBER,
rep60    NUMBER,
rep61    NUMBER,
rep62    NUMBER,
rep63    NUMBER,

```

```

rep64    NUMBER,
rep65    NUMBER,
rep66    NUMBER,
rep67    NUMBER,
rep68    NUMBER,
rep69    NUMBER,
rep70    NUMBER,
rep71    NUMBER,
rep72    NUMBER,
rep73    NUMBER,
rep74    NUMBER,
rep75    NUMBER,
rep76    NUMBER,
rep77    NUMBER,
rep78    NUMBER,
rep79    NUMBER,
rep80    NUMBER,
rep81    NUMBER,
rep82    NUMBER,
rep83    NUMBER,
rep84    NUMBER,
rep85    NUMBER,
rep86    NUMBER,
rep87    NUMBER,
rep88    NUMBER,
rep89    NUMBER,
rep90    NUMBER,
rep91    NUMBER,
rep92    NUMBER,
rep93    NUMBER,
rep94    NUMBER,
rep95    NUMBER,
rep96    NUMBER,
rep97    NUMBER,
rep98    NUMBER,
rep99    NUMBER,
rep100   NUMBER,
thk1     NUMBER,
thk2     NUMBER,
thk3     NUMBER,
thk4     NUMBER,
thk5     NUMBER,
thk6     NUMBER,
thk7     NUMBER,
thk8     NUMBER,
thk9     NUMBER,
thk10    NUMBER,
thk11    NUMBER,
thk12    NUMBER,
thk13    NUMBER,
thk14    NUMBER,
thk15    NUMBER,
thk16    NUMBER,
thk17    NUMBER,
thk18    NUMBER,
thk19    NUMBER,
thk20    NUMBER,
thk21    NUMBER,
thk22    NUMBER,
thk23    NUMBER,
thk24    NUMBER,
thk25    NUMBER,
key1     NUMBER,
key2     NUMBER,
key3     NUMBER,
key4     NUMBER,
key5     NUMBER,
key6     NUMBER,
key7     NUMBER,

```

```

        key8    NUMBER,
        key9    NUMBER,
        key10   NUMBER
    );
rem
rem Results for delivery transactions.
rem These results are from the measurement
rem interval only.
rem
CREATE TABLE bench_del_res
(
    run_name    VARCHAR2(20),
    audit_str   VARCHAR2(10),
    proc_no     NUMBER,
    rep1        NUMBER,
    rep2        NUMBER,
    rep3        NUMBER,
    rep4        NUMBER,
    rep5        NUMBER,
    rep6        NUMBER,
    rep7        NUMBER,
    rep8        NUMBER,
    rep9        NUMBER,
    rep10       NUMBER,
    rep11       NUMBER,
    rep12       NUMBER,
    rep13       NUMBER,
    rep14       NUMBER,
    rep15       NUMBER,
    rep16       NUMBER,
    rep17       NUMBER,
    rep18       NUMBER,
    rep19       NUMBER,
    rep20       NUMBER,
    rep21       NUMBER,
    rep22       NUMBER,
    rep23       NUMBER,
    rep24       NUMBER,
    rep25       NUMBER,
    rep26       NUMBER,
    rep27       NUMBER,
    rep28       NUMBER,
    rep29       NUMBER,
    rep30       NUMBER,
    rep31       NUMBER,
    rep32       NUMBER,
    rep33       NUMBER,
    rep34       NUMBER,
    rep35       NUMBER,
    rep36       NUMBER,
    rep37       NUMBER,
    rep38       NUMBER,
    rep39       NUMBER,
    rep40       NUMBER,
    rep41       NUMBER,
    rep42       NUMBER,
    rep43       NUMBER,
    rep44       NUMBER,
    rep45       NUMBER,
    rep46       NUMBER,
    rep47       NUMBER,
    rep48       NUMBER,
    rep49       NUMBER,
    rep50       NUMBER,
    rep51       NUMBER,
    rep52       NUMBER,
    rep53       NUMBER,
    rep54       NUMBER,
    rep55       NUMBER,

```

```

    rep56       NUMBER,
    rep57       NUMBER,
    rep58       NUMBER,
    rep59       NUMBER,
    rep60       NUMBER,
    rep61       NUMBER,
    rep62       NUMBER,
    rep63       NUMBER,
    rep64       NUMBER,
    rep65       NUMBER,
    rep66       NUMBER,
    rep67       NUMBER,
    rep68       NUMBER,
    rep69       NUMBER,
    rep70       NUMBER,
    rep71       NUMBER,
    rep72       NUMBER,
    rep73       NUMBER,
    rep74       NUMBER,
    rep75       NUMBER,
    rep76       NUMBER,
    rep77       NUMBER,
    rep78       NUMBER,
    rep79       NUMBER,
    rep80       NUMBER,
    rep81       NUMBER,
    rep82       NUMBER,
    rep83       NUMBER,
    rep84       NUMBER,
    rep85       NUMBER,
    rep86       NUMBER,
    rep87       NUMBER,
    rep88       NUMBER,
    rep89       NUMBER,
    rep90       NUMBER,
    rep91       NUMBER,
    rep92       NUMBER,
    rep93       NUMBER,
    rep94       NUMBER,
    rep95       NUMBER,
    rep96       NUMBER,
    rep97       NUMBER,
    rep98       NUMBER,
    rep99       NUMBER,
    rep100      NUMBER,
    thk1        NUMBER,
    thk2        NUMBER,
    thk3        NUMBER,
    thk4        NUMBER,
    thk5        NUMBER,
    thk6        NUMBER,
    thk7        NUMBER,
    thk8        NUMBER,
    thk9        NUMBER,
    thk10       NUMBER,
    thk11       NUMBER,
    thk12       NUMBER,
    thk13       NUMBER,
    thk14       NUMBER,
    thk15       NUMBER,
    thk16       NUMBER,
    thk17       NUMBER,
    thk18       NUMBER,
    thk19       NUMBER,
    thk20       NUMBER,
    thk21       NUMBER,
    thk22       NUMBER,
    thk23       NUMBER,
    thk24       NUMBER,

```

```

    thk25       NUMBER,
    key1        NUMBER,
    key2        NUMBER,
    key3        NUMBER,
    key4        NUMBER,
    key5        NUMBER,
    key6        NUMBER,
    key7        NUMBER,
    key8        NUMBER,
    key9        NUMBER,
    key10       NUMBER
)
partition by range (proc_no) (
    partition restbl5_0 values less than ( 120 )
    tablespace restbl_0,
    partition restbl5_1 values less than ( 240 )
    tablespace restbl_1,
    partition restbl5_2 values less than ( 360 )
    tablespace restbl_2,
    partition restbl5_3 values less than ( 480 )
    tablespace restbl_3,
    partition restbl5_4 values less than ( 600 )
    tablespace restbl_4,
    partition restbl5_5 values less than ( 720 )
    tablespace restbl_5,
    partition restbl5_6 values less than ( 840 )
    tablespace restbl_6,
    partition restbl5_7 values less than ( 960 )
    tablespace restbl_7,
    partition restbl5_8 values less than ( 1080 )
    tablespace restbl_8,
    partition restbl5_9 values less than ( 1200 )
    tablespace restbl_9,
    partition restbl5_10 values less than
    ( 1320 ) tablespace restbl_10,
    partition restbl5_11 values less than
    ( 1440 ) tablespace restbl_11,
    partition restbl5_12 values less than
    ( 1560 ) tablespace restbl_12,
    partition restbl5_13 values less than
    ( 1680 ) tablespace restbl_13,
    partition restbl5_14 values less than
    ( 1800 ) tablespace restbl_14,
    partition restbl5_15 values less than
    ( 1920 ) tablespace restbl_15,
    partition restbl5_16 values less than
    ( 2040 ) tablespace restbl_16,
    partition restbl5_17 values less than
    ( 2160 ) tablespace restbl_17,
    partition restbl5_18 values less than
    ( 2280 ) tablespace restbl_18,
    partition restbl5_19 values less than
    ( maxvalue ) tablespace restbl_19
);
rem
rem Aggregate results for stock level transactions.
rem These results are from the measurement
rem interval only.
rem
CREATE TABLE tpcc_sto_res
(
    run_name    VARCHAR2(20),
    rep1        NUMBER,
    rep2        NUMBER,
    rep3        NUMBER,
    rep4        NUMBER,
    rep5        NUMBER,
    rep6        NUMBER,

```

```

rep7      NUMBER,
rep8      NUMBER,
rep9      NUMBER,
rep10     NUMBER,
rep11     NUMBER,
rep12     NUMBER,
rep13     NUMBER,
rep14     NUMBER,
rep15     NUMBER,
rep16     NUMBER,
rep17     NUMBER,
rep18     NUMBER,
rep19     NUMBER,
rep20     NUMBER,
rep21     NUMBER,
rep22     NUMBER,
rep23     NUMBER,
rep24     NUMBER,
rep25     NUMBER,
rep26     NUMBER,
rep27     NUMBER,
rep28     NUMBER,
rep29     NUMBER,
rep30     NUMBER,
rep31     NUMBER,
rep32     NUMBER,
rep33     NUMBER,
rep34     NUMBER,
rep35     NUMBER,
rep36     NUMBER,
rep37     NUMBER,
rep38     NUMBER,
rep39     NUMBER,
rep40     NUMBER,
rep41     NUMBER,
rep42     NUMBER,
rep43     NUMBER,
rep44     NUMBER,
rep45     NUMBER,
rep46     NUMBER,
rep47     NUMBER,
rep48     NUMBER,
rep49     NUMBER,
rep50     NUMBER,
rep51     NUMBER,
rep52     NUMBER,
rep53     NUMBER,
rep54     NUMBER,
rep55     NUMBER,
rep56     NUMBER,
rep57     NUMBER,
rep58     NUMBER,
rep59     NUMBER,
rep60     NUMBER,
rep61     NUMBER,
rep62     NUMBER,
rep63     NUMBER,
rep64     NUMBER,
rep65     NUMBER,
rep66     NUMBER,
rep67     NUMBER,
rep68     NUMBER,
rep69     NUMBER,
rep70     NUMBER,
rep71     NUMBER,
rep72     NUMBER,
rep73     NUMBER,
rep74     NUMBER,
rep75     NUMBER,

```

```

rep76     NUMBER,
rep77     NUMBER,
rep78     NUMBER,
rep79     NUMBER,
rep80     NUMBER,
rep81     NUMBER,
rep82     NUMBER,
rep83     NUMBER,
rep84     NUMBER,
rep85     NUMBER,
rep86     NUMBER,
rep87     NUMBER,
rep88     NUMBER,
rep89     NUMBER,
rep90     NUMBER,
rep91     NUMBER,
rep92     NUMBER,
rep93     NUMBER,
rep94     NUMBER,
rep95     NUMBER,
rep96     NUMBER,
rep97     NUMBER,
rep98     NUMBER,
rep99     NUMBER,
rep100    NUMBER,
thk1      NUMBER,
thk2      NUMBER,
thk3      NUMBER,
thk4      NUMBER,
thk5      NUMBER,
thk6      NUMBER,
thk7      NUMBER,
thk8      NUMBER,
thk9      NUMBER,
thk10     NUMBER,
thk11     NUMBER,
thk12     NUMBER,
thk13     NUMBER,
thk14     NUMBER,
thk15     NUMBER,
thk16     NUMBER,
thk17     NUMBER,
thk18     NUMBER,
thk19     NUMBER,
thk20     NUMBER,
thk21     NUMBER,
thk22     NUMBER,
thk23     NUMBER,
thk24     NUMBER,
thk25     NUMBER,
key1      NUMBER,
key2      NUMBER,
key3      NUMBER,
key4      NUMBER,
key5      NUMBER,
key6      NUMBER,
key7      NUMBER,
key8      NUMBER,
key9      NUMBER,
key10     NUMBER

```

);

```

rem
rem Results for stock level transactions.
rem These results are from the measurement
rem interval only.
rem

```

```

CREATE TABLE bench_sto_res
(

```

```

run_name  VARCHAR2(20),
audit_str VARCHAR2(10),
proc_no   NUMBER,
rep1      NUMBER,
rep2      NUMBER,
rep3      NUMBER,
rep4      NUMBER,
rep5      NUMBER,
rep6      NUMBER,
rep7      NUMBER,
rep8      NUMBER,
rep9      NUMBER,
rep10     NUMBER,
rep11     NUMBER,
rep12     NUMBER,
rep13     NUMBER,
rep14     NUMBER,
rep15     NUMBER,
rep16     NUMBER,
rep17     NUMBER,
rep18     NUMBER,
rep19     NUMBER,
rep20     NUMBER,
rep21     NUMBER,
rep22     NUMBER,
rep23     NUMBER,
rep24     NUMBER,
rep25     NUMBER,
rep26     NUMBER,
rep27     NUMBER,
rep28     NUMBER,
rep29     NUMBER,
rep30     NUMBER,
rep31     NUMBER,
rep32     NUMBER,
rep33     NUMBER,
rep34     NUMBER,
rep35     NUMBER,
rep36     NUMBER,
rep37     NUMBER,
rep38     NUMBER,
rep39     NUMBER,
rep40     NUMBER,
rep41     NUMBER,
rep42     NUMBER,
rep43     NUMBER,
rep44     NUMBER,
rep45     NUMBER,
rep46     NUMBER,
rep47     NUMBER,
rep48     NUMBER,
rep49     NUMBER,
rep50     NUMBER,
rep51     NUMBER,
rep52     NUMBER,
rep53     NUMBER,
rep54     NUMBER,
rep55     NUMBER,
rep56     NUMBER,
rep57     NUMBER,
rep58     NUMBER,
rep59     NUMBER,
rep60     NUMBER,
rep61     NUMBER,
rep62     NUMBER,
rep63     NUMBER,
rep64     NUMBER,
rep65     NUMBER,
rep66     NUMBER,

```

```

rep67 NUMBER,
rep68 NUMBER,
rep69 NUMBER,
rep70 NUMBER,
rep71 NUMBER,
rep72 NUMBER,
rep73 NUMBER,
rep74 NUMBER,
rep75 NUMBER,
rep76 NUMBER,
rep77 NUMBER,
rep78 NUMBER,
rep79 NUMBER,
rep80 NUMBER,
rep81 NUMBER,
rep82 NUMBER,
rep83 NUMBER,
rep84 NUMBER,
rep85 NUMBER,
rep86 NUMBER,
rep87 NUMBER,
rep88 NUMBER,
rep89 NUMBER,
rep90 NUMBER,
rep91 NUMBER,
rep92 NUMBER,
rep93 NUMBER,
rep94 NUMBER,
rep95 NUMBER,
rep96 NUMBER,
rep97 NUMBER,
rep98 NUMBER,
rep99 NUMBER,
rep100 NUMBER,
thk1 NUMBER,
thk2 NUMBER,
thk3 NUMBER,
thk4 NUMBER,
thk5 NUMBER,
thk6 NUMBER,
thk7 NUMBER,
thk8 NUMBER,
thk9 NUMBER,
thk10 NUMBER,
thk11 NUMBER,
thk12 NUMBER,
thk13 NUMBER,
thk14 NUMBER,
thk15 NUMBER,
thk16 NUMBER,
thk17 NUMBER,
thk18 NUMBER,
thk19 NUMBER,
thk20 NUMBER,
thk21 NUMBER,
thk22 NUMBER,
thk23 NUMBER,
thk24 NUMBER,
thk25 NUMBER,
key1 NUMBER,
key2 NUMBER,
key3 NUMBER,
key4 NUMBER,
key5 NUMBER,
key6 NUMBER,
key7 NUMBER,
key8 NUMBER,
key9 NUMBER,
key10 NUMBER

```

```

)
partition by range (proc_no) (
partition restbl6_0 values less than ( 120 )
tablespace restbl_0,
partition restbl6_1 values less than ( 240 )
tablespace restbl_1,
partition restbl6_2 values less than ( 360 )
tablespace restbl_2,
partition restbl6_3 values less than ( 480 )
tablespace restbl_3,
partition restbl6_4 values less than ( 600 )
tablespace restbl_4,
partition restbl6_5 values less than ( 720 )
tablespace restbl_5,
partition restbl6_6 values less than ( 840 )
tablespace restbl_6,
partition restbl6_7 values less than ( 960 )
tablespace restbl_7,
partition restbl6_8 values less than ( 1080 )
tablespace restbl_8,
partition restbl6_9 values less than ( 1200 )
tablespace restbl_9,
partition restbl6_10 values less than
( 1320 ) tablespace restbl_10,
partition restbl6_11 values less than
( 1440 ) tablespace restbl_11,
partition restbl6_12 values less than
( 1560 ) tablespace restbl_12,
partition restbl6_13 values less than
( 1680 ) tablespace restbl_13,
partition restbl6_14 values less than
( 1800 ) tablespace restbl_14,
partition restbl6_15 values less than
( 1920 ) tablespace restbl_15,
partition restbl6_16 values less than
( 2040 ) tablespace restbl_16,
partition restbl6_17 values less than
( 2160 ) tablespace restbl_17,
partition restbl6_18 values less than
( 2280 ) tablespace restbl_18,
partition restbl6_19 values less than
( maxvalue ) tablespace restbl_19
);
commit;
set echo off;
rem spool off;
rem exit;
:cre_tab.sql
rem
rem
=====
rem Copyright (c) 1995 Oracle Corp,
Redwood Shores, CA |
rem OPEN SYSTEMS
PERFORMANCE GROUP |
rem All Rights Reserved
|
rem
=====
rem FILENAME
rem cre_tab.sql
rem DESCRIPTION
rem Create temporary tables for consistency
tests.

```

```

rem
=====
rem
rem Usage: sqlplus tpcc/tpcc @cre_tab
rem
connect tpcc/tpcc;
set echo on;
drop table temp_o1;
drop table temp_no;
drop table temp_o2;
drop table temp_ol;
drop table tpcc_audit_tab;
create table temp_o1 (
o_w_id integer,
o_d_id integer,
o_o_id integer);
create table temp_no (
no_w_id integer,
no_d_id integer,
no_o_id integer);
create table temp_o2 (
o_w_id integer,
o_d_id integer,
o_count integer);
create table temp_ol (
ol_w_id integer,
ol_d_id integer,
ol_count integer);
create table tpcc_audit_tab (starttime date);
delete from tpcc_audit_tab;
set echo off;
:cache_views.sql
rem
rem This script creates four views that when
queried will return
rem the total number of buffers in the buffer cache
and the total
rem number of cloned buffers from each of the
database's tablespaces.
rem
rem This assumes that each table and index is in
its own tablespace.
rem If this is not the case, another query can be
used which uses the
rem database's object tables to decipher the
different objects. However,
rem this query is slower.
rem
rem This script assumes 7.3.x. If you are using
V7.2.x or below, please
rem replace svrmgr1 with sqldbmode=y.
rem
rem Modification History:
rem
rem wbattist 16-Jun-1996 Create two
additional views to keep
rem track of the number of
clones in each
rem tablespace.
rem
rem wbattist 24-May-1995 Add the state
check for the cbf view
rem to ensure that cloned
blocks are not
rem counted.
rem
connect $oracle_dba/$oracle_dba_password;

```

```

set echo on;
drop view cbf;
create view cbf as
select distinct(dbarfil) file#, count(1) blocks
from x$bh
  where dbarfil > 0 and state <> 3
group by dbarfil;
drop view cbt;
create view cbt as
select ts$.name name,sum(cbf.blocks) buffers
from cbf, file$, ts$
  where cbf.file#=file$.file# and file$.ts#=ts$.ts#
group by file$.ts#, ts$.name;
drop view cbfcln;
create view cbfcln as
select distinct(dbarfil) file#, count(1) blocks
from x$bh
  where dbarfil > 0
group by dbarfil;
drop view cbtn;
create view cbtn as
select ts$.name name,sum(cbfcln.blocks) buffers
from cbfcln, file$, ts$
  where cbfcln.file#=file$.file# and
file$.ts#=ts$.ts#
group by file$.ts#, ts$.name;
set echo off;
create_table.sh
#sqlplus tpcc/tpcc @sql/createtable_cust &
#sqlplus tpcc/tpcc @sql/createtable_ware &
#sqlplus tpcc/tpcc @sql/createtable_dist &
#sqlplus tpcc/tpcc @sql/createtable_item &
#sqlplus tpcc/tpcc @sql/createtable_stok &
#sqlplus tpcc/tpcc @sql/createtable_hist &
#sqlplus tpcc/tpcc @sql/createtable_nord &
#sqlplus tpcc/tpcc @sql/createtable_ordr
#sqlplus tpcc/tpcc @sql/createtable_ordl &
wait
create_ts.sh
CPU=20
## space.sh name number size1 size2 CPU
#sh space_2.sh ware 1 255M 258944K $CPU &
#sh space.sh cust 96 16220M 300944K $CPU
#sh space_2.sh dist 1 2500M 1276768K $CPU &
#sh space.sh hist 8 15140M 100965K $CPU
#sh space.sh stok 109 16280M 339173K $CPU
sh space.sh item 1 32M 30464K $CPU
wait
#sh space_16.sh ordr 23 64980M 101342K
$CPU
wait
sh space.sh nord 3 15370M 101169K $CPU
wait
sh space_16.sh istok 3 61240M 14544K $CPU
sh space_16.sh icust1 2 43130M 10243K $CPU
sh space_16.sh icust2 6 53810M 12780K $CPU
sh space_16.sh iordr2 3 33450M 7944K $CPU
sh space_temp.sh temp 36 16310M 202639K
$CPU
wait
sh space.sh iware 1 75M 18K $CPU
sh space.sh idist 1 255M 61K $CPU
sh space.sh iitem 1 11M 38K $CPU
wait

```

```

createdb.sql
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreatedb.sh Fri Aug 29 11:50:56
JST 2003 */
spool createdb.log
set echo on
shutdown abort
startup pfile=p_create.ora nomount
create database tpcc
controlfile reuse
maxinstances 1
datafile
'ora_dev/SYS' size 600M reuse
logfile 'ora_dev/LOG_0' size 4096M reuse,
'ora_dev/LOG_1' size 4096M reuse
sysaux datafile 'ora_dev/aux.df' size 120M
reuse ;

create undo tablespace undo_1 datafile
'ora_dev/roll_0_0' size 8096M reuse blocksize
8K;
set echo off
exit sql.sqlcode
createindex_icust1.sql
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreateindex.sh Fri Aug 29 11:51:23
JST 2003 */
set timing on
  set sqlblanklines on
  spool createindex_icust1.log ;
  set echo on ;
  drop index icust1 ;
  create unique index icust1 on cust ( c_w_id
, c_d_id
, c_id )
pctfree 1 intrans 3
storage ( buffer_pool default )
parallel 256
tablespace icust1_0 ;
  set echo off
  spool off
  exit sql.sqlcode;
createindex_icust2.sql
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreateindex.sh Fri Aug 29 11:51:23
JST 2003 */
set timing on
  set sqlblanklines on
  spool createindex_icust2.log ;
  set echo on ;
  drop index icust2 ;
  create unique index icust2 on cust ( c_last
, c_w_id
, c_d_id
, c_first
, c_id )
pctfree 1 intrans 3
storage ( buffer_pool default )
parallel 256
tablespace icust2_0 ;
  set echo off
  spool off
  exit sql.sqlcode;

```

```

createindex_idist.sql
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreateindex.sh Fri Aug 29 11:51:24
JST 2003 */
set timing on
  set sqlblanklines on
  spool createindex_idist.log ;
  set echo on ;
  drop index idist ;
  create unique index idist on dist ( d_w_id
, d_id )
pctfree 5 intrans 3
storage ( buffer_pool default )
parallel 256
tablespace idist_0 ;
  set echo off
  spool off
  exit sql.sqlcode;
createindex_iitem.sql
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreateindex.sh Fri Aug 29 11:51:25
JST 2003 */
set timing on
  set sqlblanklines on
  spool createindex_iitem.log ;
  set echo on ;
  drop index iitem ;
  create unique index iitem on item ( i_id )
pctfree 5 intrans 4
storage ( buffer_pool default )
parallel 256
tablespace iitem_0 ;
  set echo off
  spool off
  exit sql.sqlcode;
createindex_inordr.sql
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreateindex.sh Fri Aug 29 11:51:27
JST 2003 */
set timing on
  exit 0;
createindex_iordl.sql
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreateindex.sh Fri Aug 29 11:51:26
JST 2003 */
set timing on
  exit 0;
createindex_iordr1.sql
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreateindex.sh Fri Aug 29 11:51:25
JST 2003 */
set timing on
  exit 0;
createindex_iordr2.sql
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreateindex.sh Fri Aug 29 11:51:26
JST 2003 */
set timing on

```

```

set sqlblanklines on
spool createindex_iordr2.log ;
set echo on ;
drop index iordr2 ;
create unique index iordr2 on order ( o_c_id
, o_d_id
, o_w_id
, o_id )
pctfree 25 initrans 4
storage ( buffer_pool default )
parallel 256
tablespace iordr2_0 ;
set echo off
spool off
exit sql.sqlcode;
:~::~~::~~:
createindex_istok.sql
:~::~~::~~:
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreateindex.sh Fri Aug 29 11:51:24
JST 2003 */
set timing on
set sqlblanklines on
spool createindex_istok.log ;
set echo on ;
drop index istok ;
create unique index istok on stok ( s_i_id
, s_w_id )
pctfree 1 initrans 3
storage ( buffer_pool default )
parallel 256
tablespace istok_0 ;
set echo off
spool off
exit sql.sqlcode;
:~::~~::~~:
createindex_iware.sql
:~::~~::~~:
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreateindex.sh Fri Aug 29 11:51:22
JST 2003 */
set timing on
set sqlblanklines on
spool createindex_iware.log ;
set echo on ;
drop index iware ;
create unique index iware on ware ( w_id )
pctfree 1 initrans 3
storage ( buffer_pool default )
parallel 256
tablespace iware_0 ;
set echo off
spool off
exit sql.sqlcode;
:~::~~::~~:
createmisc.sh
:~::~~::~~:
#!/bin/sh
$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect
spool createmisc.log
set echo on;
alter user tpcc temporary tablespace system;
grant execute on dbms_lock to public;
grant execute on dbms_pipe to public;
grant select on v_\\$parameter to public;

REM
REM begin plsql_mon.sql

```

```

REM
connect tpcc/tpcc;
set echo on;
CREATE OR REPLACE PACKAGE
plsql_mon_pack
IS
PROCEDURE print
(
info VARCHAR2
);
END;
/
show errors;
CREATE OR REPLACE PACKAGE BODY
plsql_mon_pack
IS
PROCEDURE print
(
info VARCHAR2
)
IS
s NUMBER;
BEGIN
dbms_pipe.pack_message (info);
s := dbms_pipe.send_message ('plsql_mon');
IF (s <> 0) THEN
raise_application_error (-20000, 'Error: ' ||
to_char(s) ||
' sending on pipe');
END IF;
END;
END;
/
show errors;
set echo off;
REM
REM end plsql_mon.sql
REM
REM
REM begin cre_tab.sql
REM
connect tpcc/tpcc;
set echo on;
drop table temp_o1;
drop table temp_no;
drop table temp_o2;
drop table temp_ol;
drop table tpcc_audit_tab;
create table temp_o1 (
o_w_id integer,
o_d_id integer,
o_o_id integer);
create table temp_no (
no_w_id integer,
no_d_id integer,
no_o_id integer);
create table temp_o2 (
o_w_id integer,
o_d_id integer,
o_count integer);
create table temp_ol (
ol_w_id integer,
ol_d_id integer,
ol_count integer);
create table tpcc_audit_tab (starttime date);
delete from tpcc_audit_tab;
set echo off;
REM
REM
REM end cre_tab.sql

```

```

REM
REM
REM begin views.sql
REM
connect tpcc/tpcc;
set echo on;
create or replace view wh_cust
(w_id, w_tax, c_id, c_d_id, c_w_id, c_discount,
c_last, c_credit)
as select w.w_id, w.w_tax,
c.c_id, c.c_d_id, c.c_w_id, c.c_discount,
c.c_last, c.c_credit
from cust c, ware w
where w.w_id = c.c_w_id;
create or replace view wh_dist
(w_id, d_id, d_tax, d_next_o_id, w_tax)
as select w.w_id, d.d_id, d.d_tax, d.d_next_o_id,
w.w_tax
from dist d, ware w
where w.w_id = d.d_w_id;
create or replace view stock_item
(i_id, s_w_id, i_price, i_name, i_data, s_data,
s_quantity,
s_order_cnt, s_ytd, s_remote_cnt,
s_dist_01, s_dist_02, s_dist_03, s_dist_04,
s_dist_05,
s_dist_06, s_dist_07, s_dist_08, s_dist_09,
s_dist_10)
as
select i.i_id, s.w_id, i.i_price, i.i_name, i.i_data,
s_data, s_quantity,
s_order_cnt, s_ytd, s_remote_cnt,
s_dist_01, s_dist_02, s_dist_03, s_dist_04,
s_dist_05,
s_dist_06, s_dist_07, s_dist_08, s_dist_09,
s_dist_10
from stok s, item i
where i.i_id = s.s_i_id;
set echo off;
REM
REM end views.sql
REM
REM
REM begin dml.sql
REM
connect tpcc/tpcc;
set echo on;
alter table ware disable table lock;
alter table dist disable table lock;
alter table cust disable table lock;
alter table hist disable table lock;
alter table item disable table lock;
alter table stok disable table lock;
alter table order disable table lock;
alter table nord disable table lock;
alter table ordl disable table lock;
set echo off;
REM
REM end dml.sql
REM
REM
REM begin extent.sql
REM

$SYS_CONNECTION_STRING

@$tpcc_sql_dir/extent
@$tpcc_sql_dir/freeext
exit sql.sqlcode;

```

```

!
:
:
createspacestats.sh
:
#!/bin/sh
$tpcc_sqlplus $tpcc_dba_user_pass
@$tpcc_genscripts_dir/createspacestats > junk
2>&1
if test $? -ne 0
then
  exit 1;
else
  exit 0;
fi
:
createspacestats.sql
:
@/ora/Ora10G/tpcc-kit/scripts/sql/space_init
@/ora/Ora10G/tpcc-kit/scripts/sql/space_get 10
@/ora/Ora10G/tpcc-kit/scripts/sql/space_rpt
spool off
exit sql.sqlcode;
:
createstats.sh
:
#!/bin/sh
cstat=c_stat
if test $tpcc_np -gt 1 ; then
  cstat=c_stat_rac
fi
$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect
REM
REM create tablespace for statspack user sp
begin
REM
spool createstats.log
set echo on
  drop tablespace sp including contents;
  create tablespace sp_0 datafile
'($tpcc_disks_location)sp_0' size
$tpcc_statspack_size reuse autoextend on extent
management local uniform size 1M nologging ;
  spool off
REM
REM create tablespace for statspack user sp end
REM
REM
REM begin now call spcreate to create statspack
sp package
REM
$tpcc_internal_connect
define default_tablespace='sp_0'
define temporary_tablespace='temp_0'
@$ORACLE_HOME/rdbms/admin/spcreate
perfstat
REM note that the last thing (after spcreate) is the
perfstat password.
REM since we're not worried about security,
perfstat will do.
REM
REM tpcc stat table for NT, it is not working so I
comment it out
REM shui.lau@oracle.com it is better to use
perfmon
REM
@$tpcc_sql_dir/cs_tpcc

```

```

@$tpcc_sql_dir/cs_cpu
@$tpcc_sql_dir/cs_os
@$tpcc_sql_dir/cs_proc
@$tpcc_sql_dir/cs_thread
REM
REM tpcc result table for unix and NT
REM
@$tpcc_sql_dir/$(cstat)
@$tpcc_sql_dir/pst_c
!
:
createstoreprocs.sh
:
#!/bin/sh
$tpcc_sqlplus $tpcc_user_pass
@$tpcc_genscripts_dir/createstoreprocs > junk
2>&1
if test $? -ne 0
then
  exit 1;
else
  exit 0;
fi
:
createstoreprocs.sql
:
spool createstoreprocs.log
@/ora/Ora10G/tpcc-kit/scripts/sql/tkvcin.sql
spool off
exit sql.sqlcode;
:
createtable_cust.sql
:
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreatetable.sh Fri Aug 29 11:50:57
JST 2003 */
set timing on
  set sqlblanklines on
  spool createtable_cust.log
  set echo on
  drop cluster custcluster including tables ;
create cluster custcluster (
  c_id number
  , c_d_id number
  , c_w_id number
)
  single table
  hashkeys 1530000000
  hash is ((c_id * (51000 * 10) + c_w_id * 10 +
c_d_id) )
  size 180
  pctfree 0  intrans 3
  storage ( buffer_pool recycle )
tablespace cust_0;
create table cust (
  c_id number
  , c_d_id number
  , c_w_id number
  , c_discount number
  , c_credit char(2)
  , c_last varchar2(16)
  , c_first varchar2(16)
  , c_credit_lim number
  , c_balance binary_double
  , c_ytd_payment binary_float
  , c_payment_cnt binary_float
  , c_delivery_cnt binary_float
  , c_street_1 varchar2(20)
  , c_street_2 varchar2(20)

```

```

  , c_city varchar2(20)
  , c_state char(2)
  , c_zip char(9)
  , c_phone char(16)
  , c_since date
  , c_middle char(2)
  , c_data varchar2(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 /ora/Ora10G/tpcc-
kit/scripts/buildcreatetable.sh Fri Aug 29 11:51:00
JST 2003 */
set timing on
  set sqlblanklines on
  spool createtable_dist.log
  set echo on
  drop cluster distcluster including tables ;

create cluster distcluster (
  d_id number
  , d_w_id number
)
  single table
  hashkeys 510000
  hash is (((d_w_id * 10) + d_id) )
  size 1448
  intrans 4
  storage ( buffer_pool default )
tablespace dist_0;

create table dist (
  d_id number
  , d_w_id number
  , d_ytd binary_double
  , d_next_o_id number
  , d_tax number
  , d_name varchar2(10)
  , d_street_1 varchar2(20)
  , d_street_2 varchar2(20)
  , d_city varchar2(20)
  , d_state char(2)
  , d_zip char(9)
)
cluster distcluster (
  d_id
  , d_w_id
);
  set echo off
  spool off
  exit sql.sqlcode;
:
createtable_hist.sql
:
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreatetable.sh Fri Aug 29 11:51:02
JST 2003 */
set timing on
  set sqlblanklines on
  spool createtable_hist.log

```

```

set echo on
  drop table hist ;
create table hist (
  , h_c_id number
  , h_c_d_id number
  , h_c_w_id number
  , h_d_id number
  , h_w_id number
  , h_date date
  , h_amount binary_float
  , h_data varchar2(24)
)
pctfree 5 initrans 4
storage ( buffer_pool recycle )
tablespace hist_0 ;
set echo off
spool off
exit sql.sqlcode;
:
:
createtable_item.sql
:
:
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreatetable.sh Fri Aug 29 11:51:05
JST 2003 */
set timing on
set sqlblanklines on
spool createtable_item.log
set echo on
  drop cluster itemcluster including tables ;
create cluster itemcluster (
  i_id number(6,0)
)
single table
hashkeys 100000
hash is ((i_id))
size 120
pctfree 0 initrans 3
storage ( buffer_pool keep )
tablespace item_0;
create table item (
  i_id number(6,0)
  , i_name varchar2(24)
  , i_price binary_float
  , i_data varchar2(50)
  , i_im_id number
)
cluster itemcluster (
  i_id
);
set echo off
spool off
exit sql.sqlcode;
:
:
createtable_nord.sql
:
:
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreatetable.sh Fri Aug 29 11:51:09
JST 2003 */
set timing on
set sqlblanklines on
spool createtable_nord.log
set echo on
  drop cluster nordcluster_queue including
tables ;
create cluster nordcluster_queue (
  no_w_id number
  , no_d_id number
  , no_o_id number SORT
)

```

```

hashkeys 510000
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 /ora/Ora10G/tpcc-
kit/scripts/buildcreatetable.sh Fri Aug 29 11:51:08
JST 2003 */
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 binary_float
  , ol_supply_w_id number
  , ol_quantity binary_float
  , ol_dist_info char(24)
  , constraint ordl_uk primary key (ol_w_id, ol_d_id,
ol_o_id, ol_number )) CLUSTER
ordrcluster_queue(ol_w_id, ol_d_id, ol_o_id,
ol_number) ;
set echo off
spool off
exit sql.sqlcode;
:
:
createtable_ordr.sql
:
:
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreatetable.sh Fri Aug 29 11:51:06
JST 2003 */
set timing on
set sqlblanklines on
spool createtable_ordr.log
set echo on
  drop cluster ordrcluster_queue including
tables ;
create cluster ordrcluster_queue (
  o_w_id number
  , o_d_id number
  , o_id number SORT
  , o_number number SORT
)
hashkeys 510000
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 /ora/Ora10G/tpcc-
kit/scripts/buildcreatetable.sh Fri Aug 29 11:51:03
JST 2003 */
set timing on
set sqlblanklines on
spool createtable_stok.log
set echo on
  drop cluster stokcluster including tables ;
create cluster stokcluster (
  s_i_id number
  , s_w_id number
)
single table
hashkeys 5100000000
hash is ((s_i_id * 51000 + s_w_id))
size 290
pctfree 0 initrans 3
storage ( buffer_pool keep )
tablespace stok_0;
create table stok (
  s_i_id number
  , s_w_id number
  , s_quantity binary_float
  , s_ytd binary_double
  , s_order_cnt binary_float
  , s_remote_cnt binary_float
  , s_data varchar2(50)
  , s_dist_01 varchar2(24)
  , s_dist_02 varchar2(24)
  , s_dist_03 varchar2(24)
  , s_dist_04 varchar2(24)
  , s_dist_05 varchar2(24)
  , s_dist_06 varchar2(24)
  , s_dist_07 varchar2(24)
  , s_dist_08 varchar2(24)
  , s_dist_09 varchar2(24)
  , s_dist_10 varchar2(24)
)
cluster stokcluster (
  s_i_id
  , s_w_id
);
set echo off
spool off

```

```

exit sql.sqlcode;
:
:
create table ware.sql
:
/* created automatically by /ora/Ora10G/tpcc-
kit/scripts/buildcreatetable.sh Fri Aug 29 11:50:56
JST 2003 */
set timing on
set sqlblanklines on
spool createtable_ware.log
set echo on
drop cluster warecluster including tables ;

create cluster warecluster (
w_id number(5,0)
)
single table
hashkeys 51000
hash is ((w_id))
size 1448
initrans 2
storage (buffer_pool default)
tablespace ware_0;
create table ware (
w_id number(5,0)
,w_ytd binary_double
,w_tax number
,w_name varchar2(10)
,w_street_1 varchar2(20)
,w_street_2 varchar2(20)
,w_city varchar2(20)
,w_state char(2)
,w_zip char(9)
)
cluster warecluster (
w_id
);
set echo off
spool off
exit sql.sqlcode;
:
:
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
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$name" =
"xitem" -o "x$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
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
$stemp \& >> $createtsout
        rac_count=`expr $rac_count + 1`
        if test "$rac_count" = "$para" ; then
            rac_count=0
            echo wait >> $createtsout
        fi
    fi
fi
else
    $addfile $name\_ $i
$fileaddr$name\_ $i\_ `expr $j + 1` $filesize
$stemp \> junk$filecount 2\>\&1 \& ;
fi
eval "proc$filecount=$i"

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`

if test $out -ne 0
then
    exit 1 ;
else
    exit 0 ;
fi
:
createuser.sh
:
#!/bin/sh

echo Creating user tpcc...
$tpcc_sqlplus $tpcc_dba_user_pass
@$tpcc_sql_dir/createuser > junk 2>&1
if test $? -ne 0
then
    exit 1 ;
else
    exit 0 ;
fi
:
createuser.sql
:
spool createusertpcc.log;

set echo on;

create user tpcc identified by tpcc;

grant dba to tpcc;

set echo off;
spool off;

exit ;
:
cs_proc.sql
:
rem
rem
rem=====
rem      Copyright (c) 1997 Oracle Corp,
rem      Redwood Shores, CA  |
rem      All Rights Reserved
rem=====
rem FILENAME
rem cs_proc.sql
rem DESCRIPTION
rem Create Table for OS Specific Process Stats
rem=====
rem Usage: sqlplus tpcc/tpcc @$cs_proc.sql

connect tpcc/tpcc

```

```

set echo on

DROP TABLE process_stats;
DROP TABLE pre_process_stats;
DROP TABLE post_process_stats;

rem
rem Resource usage for a process.
rem

CREATE TABLE process_stats
(
    runname VARCHAR2(20),
    user_cpu NUMBER,
    priv_cpu NUMBER,
    processor_cpu NUMBER,
    pagefaults NUMBER
);

rem
rem Save Beginning Resource Values for a
process.
rem

CREATE TABLE pre_process_stats
(
    runname VARCHAR2(20),
    user_cpu NUMBER,
    priv_cpu NUMBER,
    processor_cpu NUMBER,
    pagefaults NUMBER
);

rem
rem Save Ending Resource Values for a process.
rem

CREATE TABLE post_process_stats
(
    runname VARCHAR2(20),
    user_cpu NUMBER,
    priv_cpu NUMBER,
    processor_cpu NUMBER,
    pagefaults NUMBER
);

commit;
set echo off
:
cs_thread.sql
:
rem
rem
rem=====
rem      Copyright (c) 1997 Oracle Corp,
rem      Redwood Shores, CA  |
rem      All Rights Reserved
rem=====
rem FILENAME
rem cs_thread.sql
rem DESCRIPTION
rem Create Table for thread statistics
rem=====

```

```
rem Usage: sqlplus tpcc/tpcc @cs_thread.sql
```

```
connect tpcc/tpcc
set echo on
```

```
DROP TABLE thread_stats;
DROP TABLE pre_thread_stats;
DROP TABLE post_thread_stats;
```

```
rem
rem Resource usage for a thread.
rem
```

```
CREATE TABLE thread_stats
(
  runname VARCHAR2(20),
  thread_id VARCHAR2(10),
  user_cpu NUMBER,
  priv_cpu NUMBER,
  processor_cpu NUMBER,
  ctxswitch NUMBER
);
```

```
rem
rem Save Beginning Resource Values for a thread.
rem
```

```
CREATE TABLE pre_thread_stats
(
  runname VARCHAR2(20),
  thread_id VARCHAR2(10),
  user_cpu NUMBER,
  priv_cpu NUMBER,
  processor_cpu NUMBER,
  ctxswitch NUMBER
);
```

```
rem
rem Save Ending Resource Values for a thread.
rem
```

```
CREATE TABLE post_thread_stats
(
  runname VARCHAR2(20),
  thread_id VARCHAR2(10),
  user_cpu NUMBER,
  priv_cpu NUMBER,
  processor_cpu NUMBER,
  ctxswitch NUMBER
);
```

```
commit;
set echo off
:
:
ddview.sh
:
:
#!/bin/sh
```

```
$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect
```

```
spool ddview.log
```

```
REM
REM In an ade/nde view we might need to run
standard.sql and dbmsstdx manually
```

```
REM catalog and catproc suppose to take care of
it
REM
```

```
@$ORACLE_HOME/plsql/admin/standard
@$ORACLE_HOME/rdbms/admin/dbmsstdx
```

```
@$ORACLE_HOME/rdbms/admin/catalog
@$ORACLE_HOME/rdbms/admin/catproc
```

```
REM
REM In an ade/nde view we might need to run
pubbld manually
REM catalog and catproc suppose to take care of
it
REM
```

```
connect system/manager
REM @$ORACLE_HOME/sqlplus/admin/pubbld
```

```
REM
REM Oracle
REM
```

```
REM if test $NUMBER_ORACLE_NODE -qt 1
REM then
```

```
REM @$ORACLE_HOME/rdbms/admin/catparr
```

```
REM fi
```

```
spool off
!
```

```
#sh $tpcc_scripts/queue.sh
```

```
:
:
defaultopts.sh
:
:
# configurable vars
tpcc_os=unix
tpcc_version=ttt
tpcc_ldrive=1
tpcc_scale=10
tpcc_np=1
tpcc_cpu=1
#megabytes
tpcc_memsize=512
#minutes
tpcc_runlen=24
```

```
tpcc_temp_imp=temp
tpcc_temp_size=10M
tpcc_temp_ext=calc
tpcc_temp_nf=calc
tpcc_temp_bs=2K
```

```
tpcc_ware_imp=iot
tpcc_ware_size=10M
tpcc_ware_ext=calc
tpcc_ware_nf=calc
tpcc_ware_bs=auto
```

```
tpcc_ware_used=-1
tpcc_ware_free=-1
tpcc_ware_trans=-1
```

```
tpcc_ware_indices=1-
```

```
tpcc_ware_autospace=t
tpcc_ware_flg=30
tpcc_ware_fl=22
```

```
tpcc_iware_imp=none
tpcc_iware_size=1M
tpcc_iware_ext=calc
tpcc_iware_nf=calc
tpcc_iware_bs=2K
```

```
tpcc_iware_used=-1
tpcc_iware_free=-1
tpcc_iware_trans=-1
```

```
tpcc_iware_autospace=t
tpcc_iware_flg=30
tpcc_iware_fl=22
```

```
tpcc_iware_indices=1-
```

```
tpcc_dist_imp=cluster
tpcc_dist_size=10M
tpcc_dist_ext=calc
tpcc_dist_nf=calc
tpcc_dist_bs=auto
```

```
tpcc_dist_used=-1
tpcc_dist_free=-1
tpcc_dist_trans=-1
```

```
tpcc_dist_indices=2-1-
```

```
tpcc_dist_autospace=t
tpcc_dist_flg=30
tpcc_dist_fl=22
```

```
tpcc_idist_imp=index
tpcc_idist_size=1M
tpcc_idist_ext=calc
tpcc_idist_nf=calc
tpcc_idist_bs=2K
```

```
tpcc_idist_used=-1
tpcc_idist_free=-1
tpcc_idist_trans=-1
```

```
tpcc_idist_autospace=t
tpcc_idist_flg=30
tpcc_idist_fl=22
```

```
tpcc_idist_indices=2-1-
```

```
tpcc_item_imp=cluster
tpcc_item_size=15M
tpcc_item_ext=calc
tpcc_item_nf=calc
tpcc_item_bs=auto
```

```
tpcc_item_used=-1
tpcc_item_free=-1
tpcc_item_trans=-1
```

```
tpcc_item_indices=1-
```

```
tpcc_item_autospace=t
tpcc_item_flg=30
```

tpcc_item_fl=22
 tpcc_item_imp=index
 tpcc_item_size=1M
 tpcc_item_ext=calc
 tpcc_item_nf=calc
 tpcc_item_bs=2K
 tpcc_item_used=-1
 tpcc_item_free=-1
 tpcc_item_trans=-1
 tpcc_item_autospace=t
 tpcc_item_flg=30
 tpcc_item_fl=22
 tpcc_item_indices=1-
 tpcc_nord_imp=iot
 tpcc_nord_size=10M
 tpcc_nord_ext=calc
 tpcc_nord_nf=calc
 tpcc_nord_bs=auto
 tpcc_nord_used=-1
 tpcc_nord_free=-1
 tpcc_nord_trans=-1
 tpcc_nord_indices=1-2-3-
 tpcc_nord_autospace=t
 tpcc_nord_flg=30
 tpcc_nord_fl=22
 tpcc_inord_imp=none
 tpcc_inord_size=1M
 tpcc_inord_ext=calc
 tpcc_inord_nf=calc
 tpcc_inord_bs=2K
 tpcc_inord_used=-1
 tpcc_inord_free=-1
 tpcc_inord_trans=-1
 tpcc_inord_autospace=t
 tpcc_inord_flg=30
 tpcc_inord_fl=22
 tpcc_inord_indices=1-2-3-
 tpcc_ordl_imp=iot
 tpcc_ordl_size=10M
 tpcc_ordl_ext=calc
 tpcc_ordl_nf=calc
 tpcc_ordl_bs=auto
 tpcc_ordl_used=-1
 tpcc_ordl_free=-1
 tpcc_ordl_trans=-1
 tpcc_ordl_indices=1-2-3-4-
 tpcc_ordl_autospace=t
 tpcc_ordl_flg=30
 tpcc_ordl_fl=22
 tpcc_iordl_imp=none

tpcc_iordl_size=1M
 tpcc_iordl_ext=calc
 tpcc_iordl_nf=calc
 tpcc_iordl_bs=2K
 tpcc_iordl_used=-1
 tpcc_iordl_free=-1
 tpcc_iordl_trans=-1
 tpcc_iordl_autospace=t
 tpcc_iordl_flg=30
 tpcc_iordl_fl=22
 tpcc_iordl_indices=1-2-3-4-
 tpcc_ordr_imp=table
 tpcc_ordr_size=10M
 tpcc_ordr_ext=calc
 tpcc_ordr_nf=calc
 tpcc_ordr_bs=auto
 tpcc_ordr_used=-1
 tpcc_ordr_free=-1
 tpcc_ordr_trans=-1
 tpcc_ordr_indices=2-3-1
 tpcc_ordr_autospace=t
 tpcc_ordr_flg=30
 tpcc_ordr_fl=22
 tpcc_iordr1_imp=index
 tpcc_iordr1_size=1M
 tpcc_iordr1_ext=calc
 tpcc_iordr1_nf=calc
 tpcc_iordr1_bs=2K
 tpcc_iordr1_used=-1
 tpcc_iordr1_free=-1
 tpcc_iordr1_trans=-1
 tpcc_iordr1_autospace=t
 tpcc_iordr1_flg=30
 tpcc_iordr1_fl=22
 tpcc_iordr1_indices=2-3-1
 tpcc_iordr2_imp=index
 tpcc_iordr2_size=1M
 tpcc_iordr2_ext=calc
 tpcc_iordr2_nf=calc
 tpcc_iordr2_bs=2K
 tpcc_iordr2_used=-1
 tpcc_iordr2_free=-1
 tpcc_iordr2_trans=-1
 tpcc_iordr2_autospace=t
 tpcc_iordr2_flg=30
 tpcc_iordr2_fl=22
 tpcc_iordr2_indices=2-3-4-1
 tpcc_stok_imp=cluster
 tpcc_stok_size=35M
 tpcc_stok_ext=calc

tpcc_stok_nf=calc
 tpcc_stok_bs=auto
 tpcc_stok_used=-1
 tpcc_stok_free=-1
 tpcc_stok_trans=-1
 tpcc_stok_indices=1-2-
 tpcc_stok_autospace=t
 tpcc_stok_flg=30
 tpcc_stok_fl=22
 tpcc_istok_imp=index
 tpcc_istok_size=1M
 tpcc_istok_ext=calc
 tpcc_istok_nf=calc
 tpcc_istok_bs=2K
 tpcc_istok_used=-1
 tpcc_istok_free=-1
 tpcc_istok_trans=-1
 tpcc_istok_autospace=t
 tpcc_istok_flg=30
 tpcc_istok_fl=22
 tpcc_istok_indices=1-2-
 tpcc_cust_imp=cluster
 tpcc_cust_size=25M
 tpcc_cust_ext=calc
 tpcc_cust_nf=calc
 #bs
 tpcc_cust_bs=2K
 tpcc_cust_used=-1
 tpcc_cust_free=-1
 tpcc_cust_trans=-1
 tpcc_cust_indices=1-2-3-
 tpcc_cust_autospace=t
 tpcc_cust_flg=30
 tpcc_cust_fl=22
 tpcc_icust1_imp=index
 tpcc_icust1_size=1M
 tpcc_icust1_ext=calc
 tpcc_icust1_nf=calc
 tpcc_icust1_bs=2K
 tpcc_icust1_used=-1
 tpcc_icust1_free=-1
 tpcc_icust1_trans=-1
 tpcc_icust1_autospace=t
 tpcc_icust1_flg=30
 tpcc_icust1_fl=22
 tpcc_icust1_indices=1-2-3-
 tpcc_icust2_imp=index
 tpcc_icust2_size=1M
 tpcc_icust2_ext=calc
 tpcc_icust2_nf=calc
 tpcc_icust2_bs=2K

```

tpcc_icust2_used=-1
tpcc_icust2_free=-1
tpcc_icust2_trans=-1

tpcc_icust2_autospace=t
tpcc_icust2_flg=30
tpcc_icust2_fl=22

tpcc_icust2_indices=6-7-1-2-3-

tpcc_hist_imp=table
tpcc_hist_size=10M
tpcc_hist_ext=calc
tpcc_hist_nf=calc
tpcc_hist_bs=auto

tpcc_hist_used=-1
tpcc_hist_free=-1
tpcc_hist_trans=-1

tpcc_hist_indices=no

tpcc_hist_autospace=t
tpcc_hist_flg=30
:
:
dml.sql
:
REM=====
REM Copyright (c) 1996 Oracle Corp,
Redwood Shores, CA |
REM OPEN SYSTEMS
PERFORMANCE GROUP |
REM All Rights Reserved
|
REM=====
REM FILENAME
REM dml.sql
REM DESCRIPTION
REM Disable table locks for TPC-C tables.
REM USAGE
REM sqlplus tpcc/tpcc dml.sql
REM=====

connect tpcc/tpcc;
set echo on;

alter table ware disable table lock;
alter table dist disable table lock;
alter table cust disable table lock;
alter table hist disable table lock;
alter table item disable table lock;
alter table stok disable table lock;
alter table ordr disable table lock;
alter table nord disable table lock;
alter table ordl disable table lock;

set echo off;

connect $oracle_dba/$oracle_dba_password;
:
:
driver.sh
:
:
#!/bin/sh

./stepenv.sh

```

```

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

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

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" = xithrough; 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/`/`'
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

:
:
estsize.sh
:
:
#!/bin/sh
# round down closest k or m from number of
kilobytes.

# fairly small, doesn't really matter
amount=$1
if $tpcc_isneg ` $tpcc_bcexpr $amount - 10000 `;
then
echo ${amount}K
exit 0;
fi;

# convert to megs, then trunc to nearest 100
amount=` $tpcc_bcexpr \($amount + 1023\) /
1024 `
amount=` $tpcc_bcexpr \($amount + 9\) / 10 `
amount=` $tpcc_bcexpr $amount \^ 10 `

echo ${amount}M
exit 0;
:
:

```

```

evenload.sh
:
:
#!/bin/sh
#evenly load using tpcc load, following
parameters:
#$1 name of the table to load- this is used to
choose where to log.
#$2 the number of things to load
#$3 the starting flag (usually b or j)
#$4 the ending flag (usually e or k)
#$5 the flag to load (h for history , c for cust, S for
stock, etc.
#$6 if true, add dummy (only used for -o so far.)
#$7 the command to be used, if not $tpcc_load

command=$7
if test -z "$command"; then
command='$tpcc_load'
fi

tablename=$1
# write out to file to load later
if expr "x$tpcc_rac_load" = "xt" > /dev/null ; then

loadout=$tpcc_genscripts_dir/load${tablename}_n
ode${tpcc_rac_node}.sh
else

loadout=$tpcc_genscripts_dir/load${tablename}.s
h
fi
rm -f ${loadout}
echo "#created automatically by $0 `date` >
$loadout
echo "rm -f load${tablename}.log" >> $loadout
echo "cd $tpcc_bench" >> $loadout

numloaders=`$tpcc_bcexpr 2 \ ` $tpcc_cpu`

if expr "x$tpcc_rac_load" = "xt" > /dev/null ; then
numloaders=`$tpcc_bcexpr $tpcc_np \ `
$tpcc_cpu \ ` 2`
fi

if expr $numloaders > $2 > /dev/null; then
numloaders=$2
fi

numloaders=`$tpcc_bcexpr $tpcc_np \ `
$tpcc_cpu \ ` 2`

echo "allprocs=" >> $loadout
curstuff=1
stuffextra=`expr $2 \% $numloaders`
stuffinc=`expr $2 / $numloaders`
curloader=0

if expr "x$tpcc_rac_load" = "xt" > /dev/null ; then
warepermode=`$tpcc_bcexpr $2 / $tpcc_np`
procpemode=`$tpcc_bcexpr $tpcc_cpu \ ` 2`
curstuff=`$tpcc_bcexpr $warepermode \ `
\ ` $tpcc_rac_node - 1 \ ` + 1`
stuffinc=`expr $warepermode / $procpemode`
stuffextra=`expr $warepermode \%
$procpemode`
curloader=`$tpcc_bcexpr $procpemode \ `
\ ` $tpcc_rac_node - 1 \ ` + 1`
endloader=`$tpcc_bcexpr $procpemode \ `
$tpcc_rac_node + 1`

```

```

while expr $curloader \< $endloader > /dev/null ;
do

newstuff=`expr $curstuff + $stuffinc +
\ ` $stuffextra / $procpemode \ `
if expr x$6 = xt > /dev/null; then
if expr $tpcc_os = unix > /dev/null; then

adddummy=\${tpcc_disks_location}dummy${curlo
ader}.dat
else
# is this what we actually want to do? check
nt stuff
adddummy=\\llll\\llldummy${curloader}.dat
fi
else
adddummy=
fi
echo "$command -M $tpcc_scale -$5
$adddummy -$3 $curstuff -$4 `expr $newstuff - 1`
>> load${tablename}${curloader}.log 2>&1 &" >>
$loadout
echo 'allprocs="$allprocs ${!}" >> $loadout

curstuff=$newstuff
stuffextra=`expr $stuffextra + 1`
curloader=`expr 1 + $curloader`
done

else
while expr $curloader \< $numloaders >
/dev/null; do
newstuff=`expr $curstuff + $stuffinc +
\ ` $stuffextra / $numloaders \ `
if expr x$6 = xt > /dev/null; then
if expr $tpcc_os = unix > /dev/null; then

adddummy=\${tpcc_disks_location}dummy${curlo
ader}.dat
else
# is this what we actually want to do? check
nt stuff
adddummy=\\llll\\llldummy${curloader}.dat
fi
else
adddummy=
fi
echo "$command -M $tpcc_scale -$5
$adddummy -$3 $curstuff -$4 `expr $newstuff - 1`
>> load${tablename}${curloader}.log 2>&1 &" >>
$loadout
echo 'allprocs="$allprocs ${!}" >> $loadout
curstuff=$newstuff

stuffextra=`expr $stuffextra + 1`
curloader=`expr 1 + $curloader`
done
fi

cat >> $loadout <<!
error=0
for curproc in \\$allprocs; do
wait \\$curproc
error=`expr $? + $error`
done
exit `expr \\$error != 0`

```

```

!
exit 0
:
:
extent.sql
:
:
REM      Copyright (c) 1994 Oracle Corp,
Belmont, CA      |
REM      OPEN SYSTEMS
PERFORMANCE GROUP      |
REM      All Rights Reserved
|
REM=====
=====+
REM FILENAME
REM      extent.sql
REM DESCRIPTION
REM      List all extents in all the TPCC
tablespaces.
REM
REM Usage: sqlplus 'sys/change_on_install as
sysdba' @extent
REM=====
=====*/

set space 2
set pagesize 2000
set echo off
set termout off
set verify off
set feedback off
spool extent.rpt
select substr(e.tablespace_name,1,8) tspace,
substr(segment_name,1,11) segment,
substr(segment_type,1,15) type,
substr(extent_id,1,5) eid, substr(file_id,1,5)
fid, blocks,
blocks * t.block_size / 1048576 size_MB
from dba_extents e, dba_tablespaces t
where owner = 'TPCC' AND ( segment_type
= 'INDEX' OR
segment_type = 'INDEX PARTITION' OR
segment_type = 'CLUSTER'
OR segment_type = 'TABLE' OR
segment_type = 'TABLE PARTITION')
AND e.tablespace_name <> 'SYSTEM'
AND e.tablespace_name =
t.tablespace_name
order by e.tablespace_name, segment_name,
extent_id, file_id;

select substr(e.tablespace_name,1,8) tspace,
substr(segment_name,1,11) segment,
sum(blocks) tot_blk, sum(blocks) *
t.block_size / 1048576 size_MB
from dba_extents e, dba_tablespaces t
where owner = 'TPCC' AND ( segment_type
= 'INDEX' OR
segment_type = 'INDEX PARTITION' OR
segment_type = 'CLUSTER'
OR segment_type = 'TABLE' OR
segment_type = 'TABLE PARTITION')
AND e.tablespace_name <> 'SYSTEM'
AND e.tablespace_name =
t.tablespace_name
group by e.tablespace_name, segment_name,
t.block_size
order by e.tablespace_name, segment_name;
spool off;

```

```

:.....:
extractcols.sh
:.....:
#!/bin/sh

extractcols(){
table=$1
tablecols=`tp $table cols`

lines=`echo "$tablecols" | sed -e's/^ */' | cut -d'-' -
f1 |
sed -e's/(.*)'\1',/' | tr -s '\n' '\n'`
echo "tablecols[${table}] = [$lines" | sed -e's/,
$/;/'
}

defaultcols(){
table=$1
tableinds=`tp $table indices`

indarr=`echo "$tableinds" | sed -e's/^([0-9][0-
9]*)\1/g`
echo "tableinds[${table}] = [$indarr];" | sed -e's/-
//g' | sed -e's/,//g' | sed -e's/\n//g'
}

.../stepenv.sh
.$tpcc_scripts/tabledata.sh
for table in $tpcc_table_list; do
extractcols $table
done

for table in $tpcc_table_list $tpcc_index_list; do
defaultcols $table
:.....:
freeext.sql
:.....:
REM=====
=====+
REM      Copyright (c) 1994 Oracle Corp,
Belmont, CA      |
REM              OPEN SYSTEMS
PERFORMANCE GROUP |
REM              All Rights Reserved
|
REM=====
=====+
REM FILENAME
REM freeext.sql
REM DESCRIPTION
REM List all free extents in all the TPCC
tablespace
REM
REM Usage: sqlplus 'sys/change_on_install as
sysdba' @freeext
REM=====
=====*/
set space 2
set pagesize 2000
set echo off
set termout off
set verify off
set feedback off
spool freeextent.rpt
select substr(e.tablespace_name,1,8) tspace,
file_id, block_id, blocks,
blocks * t.block_size / 1048576 size_MB
from dba_free_space e, dba_tablespaces t

```

```

where e.tablespace_name =
t.tablespace_name
order by e.tablespace_name, file_id, block_id;

select substr(e.tablespace_name,1,8) tspace,
sum(blocks) tot_blk,
sum(blocks) * t.block_size / 1048576
size_MB
from dba_free_space e, dba_tablespaces t
where e.tablespace_name =
t.tablespace_name
group by e.tablespace_name, t.block_size
order by e.tablespace_name;

:.....:
fromkilobytes.sh
:.....:
#!/bin/sh
# round up to k, m, g, t from number of kilobytes.

amount=$1
if $tpcc_isneg `$tpcc_bcepr $amount - 1024`;
then
echo ${amount}K
exit 0;
fi;
amount=`$tpcc_bcepr `($amount + 1023) /
1024`
if $tpcc_isneg `$tpcc_bcepr $amount - 1024`;
then
echo ${amount}M
exit 0;
fi;
amount=`$tpcc_bcepr `($amount + 1023) /
1024`
echo ${amount}G
:.....:
hardanalyze.sql
:.....:
spool analyze.log;
set echo on;

connect tpcc/tpcc;

ANALYZE TABLE stok COMPUTE STATISTICS;
ANALYZE TABLE cust COMPUTE STATISTICS;
ANALYZE TABLE ordr COMPUTE STATISTICS;
ANALYZE TABLE ordl COMPUTE STATISTICS;
ANALYZE TABLE hist COMPUTE STATISTICS;
ANALYZE TABLE dist COMPUTE STATISTICS;
ANALYZE TABLE item COMPUTE STATISTICS;
ANALYZE TABLE ware COMPUTE STATISTICS;
ANALYZE TABLE nord COMPUTE STATISTICS;
ANALYZE index iware COMPUTE STATISTICS;
ANALYZE index idist COMPUTE STATISTICS;
ANALYZE index iitem COMPUTE STATISTICS;
ANALYZE index icust1 COMPUTE STATISTICS;
ANALYZE index icust2 COMPUTE STATISTICS;
ANALYZE index istok COMPUTE STATISTICS;
ANALYZE index iordr1 COMPUTE STATISTICS;
ANALYZE index iordr2 COMPUTE STATISTICS;

set echo off;
spool off;

exit sql.sqlcode;
:.....:
isneg.sh
:.....:

```

```

#!/bin/sh
# exit true if negative, else false

if test ` $tpcc_bcepr "$*" | cut -b1` = -; then
exit 0
else
exit 1
fi
:.....:
lcm.sh
:.....:
#!/bin/sh
# echo the lcm of two numbers

if expr $2 \> $1 > /dev/null; then
set $2 $1
# now $1 is guaranteed to be bigger
fi

lcm=$1
while expr `(` $lcm % $1 `) + `(` $lcm % $2 `) \>
0 > /dev/null; do
lcm=`expr $lcm + $1`
done

echo $lcm
:.....:
listfiles.sh
:.....:
#!/bin/sh
# Write out tablespace files to files.dat so the
person in
# charge of the test can relink them.

#automatically export
set -a
tpcc_listfiles=t

rm -f $tpcc_bench/files.dat
cd $tpcc_bench
echo "#generated automatically by files.dat `date`"
>> $tpcc_bench/files.dat
$tpcc_scripts/runscript.sh createts
cat >> $tpcc_bench/files.dat <<!
${tpcc_disks_location}aux.df $tpcc_sysaux_size
${tpcc_disks_location}control_001 25M
${tpcc_disks_location}control_002 25M
${tpcc_disks_location}sp_0 $tpcc_statspack_size
!

i=1
while test $i -le $tpcc_np; do
cat >> $tpcc_bench/files.dat <<!
${tpcc_disks_location}log_${i}_1
$tpcc_logfile_size
${tpcc_disks_location}log_${i}_2
$tpcc_logfile_size
!
i=`expr $i + 1`;
done

i=1
while test $i -le $tpcc_np; do
cat >> $tpcc_bench/files.dat <<!
${tpcc_disks_location}system_${i}
$tpcc_system_size
!
i=`expr $i + 1`;

```

```

done

i=1
while test $i -le $tpcc_np; do
  cat >> $tpcc_bench/files.dat <<!
  $(tpcc_disks_location)roll${i} $tpcc_undo_size
  !
  i=`expr $i + 1`;
done

tpcc_listfiles=f
#automatic export off
set +a

echo List of files written to $tpcc_bench/files.dat.
Stopping driver.

exit 99
.....
load_data.sh
.....

./stepenv.sh

sh loadware.sh
sh loaddist.sh
sh loaditem.sh
sh step24_tpcc.sh
sh step25_tpcc.sh
sh step23_tpcc.sh
sh step26_tpcc.sh

sh step27_tpcc.sh
.....
loadcust.sh
.....

cd $tpcc_bench
I=1
J=1
SW=1
EW=1700
INC=1700
while [ $I -le 30 ]
do

$tpcc_load -M 51000 -c -b $SW -e $EW >
step26loadcust${I}.log 2>&1 &
  I=`expr $I + 1`
  SW=`expr $SW + $INC`
  EW=`expr $EW + $INC`
done
wait

sh dummy.sh
.....
loaddist.sh
.....
cd $tpcc_bench
$tpcc_load -M $tpcc_scale -d > loaddist.log 2>&1
.....
loadfixordrordl.sh
.....
#created automatically by /ora/Ora10G/tpcc-
kit/scripts/evencustload.sh Fri Aug 29 11:51:20 JST
2003
rm -f loadfixordrordl*.log
cd $tpcc_bench
allprocs=

```

```

$tpcc_updateordrordl -M 48000 -o -b 1 -e 375 >>
loadfixordrordl0.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 376 -e 750
>> loadfixordrordl1.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 751 -e 1125
>> loadfixordrordl2.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 1126 -e
1500 >> loadfixordrordl3.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 1501 -e
1875 >> loadfixordrordl4.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 1876 -e
2250 >> loadfixordrordl5.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 2251 -e
2625 >> loadfixordrordl6.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 2626 -e
3000 >> loadfixordrordl7.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 3001 -e
3375 >> loadfixordrordl8.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 3376 -e
3750 >> loadfixordrordl9.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 3751 -e
4125 >> loadfixordrordl10.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 4126 -e
4500 >> loadfixordrordl11.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 4501 -e
4875 >> loadfixordrordl12.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 4876 -e
5250 >> loadfixordrordl13.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 5251 -e
5625 >> loadfixordrordl14.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 5626 -e
6000 >> loadfixordrordl15.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 6001 -e
6375 >> loadfixordrordl16.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 6376 -e
6750 >> loadfixordrordl17.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 6751 -e
7125 >> loadfixordrordl18.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 7126 -e
7500 >> loadfixordrordl19.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 7501 -e
7875 >> loadfixordrordl20.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 7876 -e
8250 >> loadfixordrordl21.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 8251 -e
8625 >> loadfixordrordl22.log 2>&1 &
allprocs="$allprocs ${I}"

```

```

$tpcc_updateordrordl -M 48000 -o -b 8626 -e
9000 >> loadfixordrordl23.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 9001 -e
9375 >> loadfixordrordl24.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 9376 -e
9750 >> loadfixordrordl25.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 9751 -e
10125 >> loadfixordrordl26.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 10126 -e
10500 >> loadfixordrordl27.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 10501 -e
10875 >> loadfixordrordl28.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 10876 -e
11250 >> loadfixordrordl29.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 11251 -e
11625 >> loadfixordrordl30.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 11626 -e
12000 >> loadfixordrordl31.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 12001 -e
12375 >> loadfixordrordl32.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 12376 -e
12750 >> loadfixordrordl33.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 12751 -e
13125 >> loadfixordrordl34.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 13126 -e
13500 >> loadfixordrordl35.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 13501 -e
13875 >> loadfixordrordl36.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 13876 -e
14250 >> loadfixordrordl37.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 14251 -e
14625 >> loadfixordrordl38.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 14626 -e
15000 >> loadfixordrordl39.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 15001 -e
15375 >> loadfixordrordl40.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 15376 -e
15750 >> loadfixordrordl41.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 15751 -e
16125 >> loadfixordrordl42.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 16126 -e
16500 >> loadfixordrordl43.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 16501 -e
16875 >> loadfixordrordl44.log 2>&1 &
allprocs="$allprocs ${I}"
$tpcc_updateordrordl -M 48000 -o -b 16876 -e
17250 >> loadfixordrordl45.log 2>&1 &
allprocs="$allprocs ${I}"

```



```

$tpcc_updateordrordl -M 48000 -o -b 43126 -e
43500 >> loadfixordrordl115.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 48000 -o -b 43501 -e
43875 >> loadfixordrordl116.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 48000 -o -b 43876 -e
44250 >> loadfixordrordl117.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 48000 -o -b 44251 -e
44625 >> loadfixordrordl118.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 48000 -o -b 44626 -e
45000 >> loadfixordrordl119.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 48000 -o -b 45001 -e
45375 >> loadfixordrordl120.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 48000 -o -b 45376 -e
45750 >> loadfixordrordl121.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 48000 -o -b 45751 -e
46125 >> loadfixordrordl122.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 48000 -o -b 46126 -e
46500 >> loadfixordrordl123.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 48000 -o -b 46501 -e
46875 >> loadfixordrordl124.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 48000 -o -b 46876 -e
47250 >> loadfixordrordl125.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 48000 -o -b 47251 -e
47625 >> loadfixordrordl126.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 48000 -o -b 47626 -e
48000 >> loadfixordrordl127.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
    wait $curproc
    error=`expr $? + $error`
done
exit `expr $error != 0`
loadhist.sh
cd $tpcc_bench
l=1
SW=1
EW=1700
INC=1700
while [ $l -le 30 ]
do
$tpcc_load -M 51000 -h -b $SW -e $EW >
step23loadhist${!}.log0 2>&1 &
l=`expr $l + 1`
SW=`expr $SW + $INC`
EW=`expr $EW + $INC`
done
wait
loaditem.sh
cd $tpcc_bench
$tpcc_load -M $tpcc_scale -i > loaditem.log 2>&1

```

```

loadnord.sh
cd $tpcc_bench
l=1
SW=1
EW=1700
INC=1700
while [ $l -le 30 ]
do
$tpcc_load -M $tpcc_scale -n -b $SW -e $EW
>step24loadnord${!}.log0 2>&1 &
l=`expr $l + 1`
SW=`expr $SW + $INC`
EW=`expr $EW + $INC`
done
wait
loadordrordl.sh
cd $tpcc_bench
l=1
SW=1
EW=1700
INC=1700
while [ $l -le 30 ]
do
$tpcc_load -M 51000 -o /tmp/dummy0.dat -b $SW
-e $EW >> step25loadordrordl1.log${!} 2>&1 &
l=`expr $l + 1`
SW=`expr $SW + $INC`
EW=`expr $EW + $INC`
done
wait
sh dummy.sh
loadstock.sh
cd $tpcc_bench
l=1
SI=1
EI=4000
INC=4000
while [ $l -le 25 ]
do
$tpcc_load -M 51000 -S -j $SI -k $EI
>step27loadstok${!}.log0 2>&1 &
l=`expr $l + 1`
SI=`expr $SI + $INC`
EI=`expr $EI + $INC`
done
wait
sh dummy.sh
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=/location/of/db/files

#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
notneg.sh
#!bin/sh
# echo the first and second params provided $2 is
greater than -1.

if expr $2 \> -1 > /dev/null; then
    echo $1 $2
else
    echo
fi
options.sh
tpcc_os='unix'
tpcc_version='ttt'
tpcc_ldrive='1'
tpcc_scale='51000'
tpcc_np='1'
tpcc_cpu='64'
tpcc_memsize='256000'
tpcc_runlen='25'
tpcc_compress='t'
tpcc_overflow='t'
tpcc_defbs='4'

```

```
tpcc_jeec_number='t'

tpcc_cust_imp='cluster'
tpcc_cust_size='calc'
tpcc_cust_ext='calc'
tpcc_cust_nf='calc'
tpcc_cust_bs='auto'
tpcc_cust_used='-1'
tpcc_cust_free='0'
tpcc_cust_trans='3'
tpcc_cust_autospace='t'
tpcc_cust_flg='43'
tpcc_cust_fl='22'
tpcc_cust_rsize='auto'
tpcc_cust_hkey='auto'
tpcc_cust_hash='auto'
tpcc_cust_bpool='recycle'
tpcc_cust_indices=3-2-1-
```

```
tpcc_dist_imp='cluster'
tpcc_dist_size='calc'
tpcc_dist_ext='calc'
tpcc_dist_nf='calc'
tpcc_dist_bs='2K'
tpcc_dist_used='-1'
tpcc_dist_free='-1'
tpcc_dist_trans='4'
tpcc_dist_autospace='t'
tpcc_dist_flg='43'
tpcc_dist_fl='22'
tpcc_dist_rsize='auto'
tpcc_dist_hkey='auto'
tpcc_dist_hash='auto'
tpcc_dist_bpool='default'
tpcc_dist_indices=2-1-
```

```
tpcc_hist_imp='table'
tpcc_hist_size='1791'
tpcc_hist_ext='calc'
tpcc_hist_nf='calc'
tpcc_hist_bs='auto'
tpcc_hist_used='-1'
tpcc_hist_free='5'
tpcc_hist_trans='4'
tpcc_hist_autospace='t'
tpcc_hist_flg='43'
tpcc_hist_fl='22'
tpcc_hist_rsize='auto'
tpcc_hist_hkey='auto'
tpcc_hist_hash='auto'
tpcc_hist_bpool='recycle'
tpcc_hist_indices=1-1-1-1-1-1-1-
```

```
tpcc_item_imp='cluster'
tpcc_item_size='calc'
tpcc_item_ext='calc'
tpcc_item_nf='calc'
tpcc_item_bs='auto'
tpcc_item_used='-1'
tpcc_item_free='0'
tpcc_item_trans='3'
tpcc_item_autospace='t'
tpcc_item_flg='43'
tpcc_item_fl='22'
tpcc_item_rsize='auto'
tpcc_item_hkey='auto'
tpcc_item_hash='auto'
tpcc_item_bpool='keep'
tpcc_item_indices=1-
```

```
tpcc_nord_imp='queue'
tpcc_nord_size='178'
tpcc_nord_ext='calc'
tpcc_nord_nf='calc'
tpcc_nord_bs='auto'
tpcc_nord_used='-1'
tpcc_nord_free='5'
tpcc_nord_trans='4'
tpcc_nord_autospace='t'
tpcc_nord_flg='43'
tpcc_nord_fl='22'
tpcc_nord_rsize='auto'
tpcc_nord_hkey='auto'
tpcc_nord_hash='auto'
tpcc_nord_bpool='default'
tpcc_nord_indices=1-2-3-
```

```
tpcc_ordl_imp='queue'
tpcc_ordl_size='21775'
tpcc_ordl_ext='calc'
tpcc_ordl_nf='calc'
tpcc_ordl_bs='16K'
tpcc_ordl_used='-1'
tpcc_ordl_free='5'
tpcc_ordl_trans='4'
tpcc_ordl_autospace='t'
tpcc_ordl_flg='43'
tpcc_ordl_fl='22'
tpcc_ordl_rsize='auto'
tpcc_ordl_hkey='auto'
tpcc_ordl_hash='auto'
tpcc_ordl_bpool='default'
tpcc_ordl_indices=1-2-3-4-
```

```
tpcc_ordr_imp='queue'
tpcc_ordr_size='1206'
tpcc_ordr_ext='calc'
tpcc_ordr_nf='calc'
tpcc_ordr_bs='16K'
tpcc_ordr_used='-1'
tpcc_ordr_free='5'
tpcc_ordr_trans='4'
tpcc_ordr_autospace='t'
tpcc_ordr_flg='43'
tpcc_ordr_fl='22'
tpcc_ordr_rsize='auto'
tpcc_ordr_hkey='auto'
tpcc_ordr_hash='auto'
tpcc_ordr_bpool='default'
tpcc_ordr_indices=2-3-1-
```

```
tpcc_stok_imp='cluster'
tpcc_stok_size='calc'
tpcc_stok_ext='calc'
tpcc_stok_nf='calc'
tpcc_stok_bs='auto'
tpcc_stok_used='-1'
tpcc_stok_free='0'
tpcc_stok_trans='3'
tpcc_stok_autospace='t'
tpcc_stok_flg='43'
tpcc_stok_fl='22'
tpcc_stok_rsize='auto'
tpcc_stok_hkey='auto'
tpcc_stok_hash='auto'
tpcc_stok_bpool='keep'
tpcc_stok_indices=1-2-
```

```
tpcc_ware_imp='cluster'
tpcc_ware_size='calc'
tpcc_ware_ext='calc'
tpcc_ware_nf='calc'
tpcc_ware_bs='2K'
tpcc_ware_used='-1'
tpcc_ware_free='-1'
tpcc_ware_trans='2'
tpcc_ware_autospace='t'
tpcc_ware_flg='43'
tpcc_ware_fl='22'
tpcc_ware_rsize='auto'
tpcc_ware_hkey='auto'
tpcc_ware_hash='auto'
tpcc_ware_bpool='default'
tpcc_ware_indices=1-
```

```
tpcc_icust1_imp='index'
tpcc_icust1_size='736'
tpcc_icust1_ext='calc'
tpcc_icust1_nf='calc'
tpcc_icust1_bs='16K'
tpcc_icust1_used='-1'
tpcc_icust1_free='1'
tpcc_icust1_trans='3'
tpcc_icust1_autospace='t'
tpcc_icust1_flg='43'
tpcc_icust1_fl='22'
tpcc_icust1_rsize='auto'
tpcc_icust1_hkey='auto'
tpcc_icust1_hash='auto'
tpcc_icust1_bpool='default'
tpcc_icust1_indices=3-2-1-
```

```
tpcc_icust2_imp='index'
tpcc_icust2_size='4591'
tpcc_icust2_ext='calc'
tpcc_icust2_nf='calc'
tpcc_icust2_bs='16K'
tpcc_icust2_used='-1'
tpcc_icust2_free='1'
tpcc_icust2_trans='3'
tpcc_icust2_autospace='t'
tpcc_icust2_flg='43'
tpcc_icust2_fl='22'
tpcc_icust2_rsize='auto'
tpcc_icust2_hkey='auto'
tpcc_icust2_hash='auto'
tpcc_icust2_bpool='default'
tpcc_icust2_indices=6-3-2-7-1-
```

```
tpcc_idist_imp='index'
tpcc_idist_size='4'
tpcc_idist_ext='calc'
tpcc_idist_nf='calc'
tpcc_idist_bs='auto'
tpcc_idist_used='-1'
tpcc_idist_free='5'
tpcc_idist_trans='3'
tpcc_idist_autospace='t'
tpcc_idist_flg='43'
tpcc_idist_fl='22'
tpcc_idist_rsize='auto'
tpcc_idist_hkey='auto'
tpcc_idist_hash='auto'
tpcc_idist_bpool='default'
tpcc_idist_indices=2-1-
```

```
tpcc_iitem_imp='index'
```

```
tpcc_item_size='2048'
tpcc_item_ext='calc'
tpcc_item_nf='calc'
tpcc_item_bs='auto'
tpcc_item_used='-1'
tpcc_item_free='5'
tpcc_item_trans='4'
tpcc_item_autospace='t'
tpcc_item_flg='43'
tpcc_item_fl='22'
tpcc_item_rsize='auto'
tpcc_item_hkey='auto'
tpcc_item_hash='auto'
tpcc_item_bpool='default'
tpcc_item_indices=1-
```

```
tpcc_inord_imp='none'
tpcc_inord_size='229'
tpcc_inord_ext='calc'
tpcc_inord_nf='calc'
tpcc_inord_bs='auto'
tpcc_inord_used='-1'
tpcc_inord_free='5'
tpcc_inord_trans='4'
tpcc_inord_autospace='t'
tpcc_inord_flg='43'
tpcc_inord_fl='22'
tpcc_inord_rsize='auto'
tpcc_inord_hkey='auto'
tpcc_inord_hash='auto'
tpcc_inord_bpool='default'
tpcc_inord_indices=1-2-3-
```

```
tpcc_iordl_imp='none'
tpcc_iordl_size='8072'
tpcc_iordl_ext='calc'
tpcc_iordl_nf='calc'
tpcc_iordl_bs='auto'
tpcc_iordl_used='-1'
tpcc_iordl_free='5'
tpcc_iordl_trans='4'
tpcc_iordl_autospace='t'
tpcc_iordl_flg='43'
tpcc_iordl_fl='22'
tpcc_iordl_rsize='auto'
tpcc_iordl_hkey='auto'
tpcc_iordl_hash='auto'
tpcc_iordl_bpool='default'
tpcc_iordl_indices=1-2-3-4-
```

```
tpcc_iordr1_imp='none'
tpcc_iordr1_size='703'
tpcc_iordr1_ext='calc'
tpcc_iordr1_nf='calc'
tpcc_iordr1_bs='auto'
tpcc_iordr1_used='-1'
tpcc_iordr1_free='1'
tpcc_iordr1_trans='3'
tpcc_iordr1_autospace='t'
tpcc_iordr1_flg='43'
tpcc_iordr1_fl='22'
tpcc_iordr1_rsize='auto'
tpcc_iordr1_hkey='auto'
tpcc_iordr1_hash='auto'
tpcc_iordr1_bpool='default'
tpcc_iordr1_indices=2-3-1-
```

```
tpcc_iordr2_imp='index'
tpcc_iordr2_size='1135'
```

```
tpcc_iordr2_ext='calc'
tpcc_iordr2_nf='calc'
tpcc_iordr2_bs='16K'
tpcc_iordr2_used='-1'
tpcc_iordr2_free='25'
tpcc_iordr2_trans='4'
tpcc_iordr2_autospace='t'
tpcc_iordr2_flg='43'
tpcc_iordr2_fl='22'
tpcc_iordr2_rsize='auto'
tpcc_iordr2_hkey='auto'
tpcc_iordr2_hash='auto'
tpcc_iordr2_bpool='default'
tpcc_iordr2_indices=2-3-4-1-
```

```
tpcc_istok_imp='index'
tpcc_istok_size='2090'
tpcc_istok_ext='calc'
tpcc_istok_nf='calc'
tpcc_istok_bs='16K'
tpcc_istok_used='-1'
tpcc_istok_free='1'
tpcc_istok_trans='3'
tpcc_istok_autospace='t'
tpcc_istok_flg='43'
tpcc_istok_fl='22'
tpcc_istok_rsize='auto'
tpcc_istok_hkey='auto'
tpcc_istok_hash='auto'
tpcc_istok_bpool='default'
tpcc_istok_indices=1-2-
```

```
tpcc_iware_imp='index'
tpcc_iware_size='1'
tpcc_iware_ext='calc'
tpcc_iware_nf='calc'
tpcc_iware_bs='auto'
tpcc_iware_used='-1'
tpcc_iware_free='1'
tpcc_iware_trans='3'
tpcc_iware_autospace='t'
tpcc_iware_flg='43'
tpcc_iware_fl='22'
tpcc_iware_rsize='auto'
tpcc_iware_hkey='auto'
tpcc_iware_hash='auto'
tpcc_iware_bpool='default'
tpcc_iware_indices=1-
```

```
tpcc_temp_imp='temp'
tpcc_temp_size='16145'
tpcc_temp_ext='calc'
tpcc_temp_nf='calc'
tpcc_temp_bs='auto'
tpcc_temp_used='-1'
tpcc_temp_free='0'
tpcc_temp_trans='3'
tpcc_temp_autospace='t'
tpcc_temp_flg='43'
tpcc_temp_fl='22'
tpcc_temp_rsize='auto'
tpcc_temp_hkey='auto'
tpcc_temp_hash='auto'
tpcc_temp_bpool='default'
tpcc_temp_indices=no
```

```
.....
orst_cre.sql
.....
```

```
rem
rem
rem
=====+
=====+
rem Copyright (c) 1996 Oracle Corp,
Redwood Shores, CA |
rem OPEN SYSTEMS
PERFORMANCE GROUP |
rem All Rights Reserved
|
rem
=====+
=====+
rem FILENAME
rem orst_cre.sql
rem DESCRIPTION
rem Drop and Create Tables for Oracle
Statistics
rem
=====+
=====*/
rem
rem Usage: sqlplus internal/internal
@orst_cre.sql
rem
connect
$oracle_dba/$oracle_dba_password:
SET ECHO ON;
rem SET TERMOUT OFF;

DROP TABLE save_sysstat;
DROP TABLE save_latch;
DROP TABLE save_rollback;
DROP TABLE save_filestat;
DROP TABLE save_rowcache;
DROP TABLE save_parameter;
DROP TABLE save_wait;
DROP TABLE save_fwait;
DROP TABLE save_event;
DROP TABLE save_lockact;
DROP TABLE save_fping;
DROP TABLE save_fping2;
DROP TABLE save_ping;
DROP TABLE save_ping2;
DROP TABLE save_blkping;
DROP TABLE save_blkping2;
DROP TABLE save_kclwait;
DROP TABLE save_sqllarea;
DROP TABLE save_time;
DROP TABLE save_dfile;
DROP TABLE save_rsrc;
DROP TABLE save_circuit;
DROP TABLE save_dispatcher;
DROP TABLE save_queue;
DROP TABLE save_server;

DROP TABLE tmp_sysstat;
DROP TABLE tmp_latch;
DROP TABLE tmp_filestat;
DROP TABLE tmp_rollback;
DROP TABLE tmp_rowcache;
DROP TABLE tmp_wait;
DROP TABLE tmp_fwait;
DROP TABLE tmp_event;
DROP TABLE tmp_fping;
DROP TABLE tmp_fping2;
DROP TABLE tmp_kclwait;
DROP TABLE blockclass;
```

```

DROP TABLE tmp_lockact;
DROP TABLE zero_lockact;
DROP TABLE tmp_sqlarea;
DROP TABLE tmp_time;
DROP TABLE tmp_circuit;
DROP TABLE tmp_dispatcher;
DROP TABLE tmp_queue;
DROP TABLE tmp_server;
DROP TABLE tmp_rsrc;

rem
rem save_sysstat corresponds to v$sysstat and
v$statname
rem
CREATE TABLE save_sysstat
(
  hid      NUMBER,
  run      NUMBER,
  name     VARCHAR2(64),
  statistic# NUMBER,
  value    NUMBER
);

rem
rem save_latch corresponds to v$latch and
v$latchname
rem
CREATE TABLE save_latch
(
  hid      NUMBER,
  run      NUMBER,
  name     VARCHAR2(64),
  latch#   NUMBER,
  gets     NUMBER,
  misses   NUMBER,
  sleeps   NUMBER,
  immediate_gets NUMBER,
  immediate_misses NUMBER
);

rem
rem save_rollstat corresponds to v$rollstat and
v$rollname
rem
CREATE TABLE save_rollstat
(
  hid      NUMBER,
  run      NUMBER,
  name     VARCHAR2(30),
  USN      NUMBER,
  EXTENTS  NUMBER,
  RSSIZE   NUMBER,
  WRITES   NUMBER,
  XACTS    NUMBER,
  GETS     NUMBER,
  WAITS    NUMBER,
  OPTSIZE  NUMBER,
  HWMSIZE  NUMBER,
  SHRINKS  NUMBER,
  WRAPS    NUMBER,
  EXTENDS  NUMBER,
  AVESHRINK NUMBER,
  AVEACTIVE NUMBER
);

rem
rem save_filestat corresponds to v$filestat and
v$dbfile;

```

```

rem
CREATE TABLE save_filestat
(
  hid      NUMBER,
  run      NUMBER,
  FILE#    NUMBER,
  PHYRDS   NUMBER,
  PHYWRTS  NUMBER,
  PHYBLKRD NUMBER,
  PHYBLKWRT NUMBER,
  READTIM  NUMBER,
  WRITETIM NUMBER,
  NAME     VARCHAR2(257)
);

rem
rem save_rowcache corresponds to v$rowcache
rem
CREATE TABLE save_rowcache
(
  hid      NUMBER,
  run      NUMBER,
  cache#   NUMBER,
  type     VARCHAR2(11),
  subordinate# NUMBER,
  parameter VARCHAR2(32),
  count    NUMBER,
  usage    NUMBER,
  fixed    NUMBER,
  gets     NUMBER,
  getmisses NUMBER,
  scans    NUMBER,
  scanmisses NUMBER,
  scancompletes NUMBER,
  modifications NUMBER,
  flushes  NUMBER
);

rem
rem Create table to hold values in v$parameter
rem
CREATE TABLE save_parameter
(
  hid      NUMBER,
  run      NUMBER,
  NAME     VARCHAR2(64),
  VALUE    VARCHAR2(512)
);

rem
rem save_wait corresponds to v$wait_stat
rem
CREATE TABLE save_wait
(
  hid      NUMBER,
  run      NUMBER,
  class    VARCHAR2(18),
  count    NUMBER,
  time     NUMBER
);

rem
rem save_fwait corresponds to X$KCBFWAIT
rem
CREATE TABLE save_fwait
(
  hid      NUMBER,
  run      NUMBER,

```

```

  addr     VARCHAR2(20),
  indx     NUMBER,
  count    NUMBER,
  time     NUMBER
);

rem
rem save_event corresponds to v$system_event
rem
CREATE TABLE save_event
(
  hid      NUMBER,
  run      NUMBER,
  event     VARCHAR2(64),
  total_waits NUMBER,
  time_waited NUMBER,
  average_wait NUMBER
);

rem
rem save_lockact corresponds to v$lock_activity
rem
CREATE TABLE save_lockact
(
  hid      NUMBER,
  run      NUMBER,
  from_val  VARCHAR2(4),
  to_val    VARCHAR2(4),
  action_val VARCHAR2(51),
  counter   NUMBER
);

rem
rem save_fping corresponds to file_ping
rem
CREATE TABLE save_fping
(
  hid      NUMBER,
  run      NUMBER,
  file_id   NUMBER,
  file_name VARCHAR2(257),
  ts_name   VARCHAR2(30),
  x_to_n    NUMBER
);

rem
rem save_fping2 corresponds to file_ping with
extended ping stats
rem
CREATE TABLE save_fping2
(
  hid      NUMBER,
  run      NUMBER,
  file_id   NUMBER,
  file_name VARCHAR2(257),
  ts_name   VARCHAR2(30),
  x2n       NUMBER,
  x2s       NUMBER,
  x2ssx     NUMBER,
  s2n       NUMBER,
  cic       NUMBER,
  crt       NUMBER,
  hping     NUMBER,
  sping     NUMBER
);

rem
rem save_ping corresponds to v$ping
rem

```

```

CREATE TABLE save_ping
(
  hid      NUMBER,
  run      NUMBER,
  tablespace_name VARCHAR2(30),
  file_name VARCHAR2(257),
  kind     VARCHAR2(12),
  status   VARCHAR2(4),
  xnc      NUMBER
);

rem
rem save_ping2 corresponds to v$ping with
rem extended ping stats
rem
CREATE TABLE save_ping2
(
  hid      NUMBER,
  run      NUMBER,
  tablespace_name VARCHAR2(30),
  file#    NUMBER,
  kind     VARCHAR2(12),
  status   VARCHAR2(4),
  hpings   NUMBER,
  spings   NUMBER
);

rem
rem save_blkping corresponds to v$ping
rem
CREATE TABLE save_blkping
(
  hid      NUMBER,
  run      NUMBER,
  tablespace_name VARCHAR2(30),
  file_name VARCHAR2(257),
  kind     VARCHAR2(12),
  block#   NUMBER,
  status   VARCHAR2(4),
  xnc      NUMBER
);

rem
rem save_blkping2 corresponds to v$ping with
rem extended ping stats
rem
CREATE TABLE save_blkping2
(
  hid      NUMBER,
  run      NUMBER,
  tablespace_name VARCHAR2(30),
  file#    NUMBER,
  kind     VARCHAR2(12),
  block#   NUMBER,
  status   VARCHAR2(4),
  hpings   NUMBER,
  spings   NUMBER,
  lock_element_addr RAW(4)
);

rem
rem save_kclwait corresponds to v$kclwait
rem
CREATE TABLE save_kclwait
(
  hid      NUMBER,
  run      NUMBER,
  indx     NUMBER,
  pings    NUMBER,
  hpings   NUMBER,
  spings   NUMBER,
  wpings   NUMBER
);

rem
rem save_sqlarea corresponds to v$sqlarea
rem
CREATE TABLE save_sqlarea
(
  hid      NUMBER,
  run      NUMBER,
  sql_text VARCHAR2(1000),
  executions NUMBER,
  buffer_gets NUMBER,
  disk_reads NUMBER,
  serializable_aborts NUMBER
);

rem
rem save_time records duration of each run
rem
CREATE TABLE save_time
(
  hid      NUMBER,
  run      NUMBER,
  rtime    NUMBER
);

rem
rem save_dfile maps oracle datafile to physical
rem disks and nodes
rem
CREATE TABLE save_dfile
(
  file#    NUMBER,
  group#   NUMBER,
  gname    VARCHAR2(20),
  node#    NUMBER,
  nname    VARCHAR2(20),
  disk#    NUMBER,
  dname    VARCHAR2(20)
);

rem
rem save_rsrc corresponds to
rem v$rsrc_consumer_group
rem
CREATE TABLE save_rsrc
(
  hid      NUMBER,
  run      NUMBER,
  NAME     VARCHAR2(32),
  ACTIVE_SESSIONS NUMBER,
  EXECUTION_WAITERS NUMBER,
  REQUESTS NUMBER,
  CPU_WAIT_TIME NUMBER,
  CPU_WAITS NUMBER,
  CONSUMED_CPU_TIME NUMBER,
  YIELDS NUMBER,
  SESSIONS_QUEUED NUMBER
);

rem
rem save_circuit
rem
CREATE TABLE save_circuit
(
  hid      NUMBER,
  run      NUMBER,
  circuit  raw(4),
  msg0     NUMBER,
  msg1     NUMBER,
  msgs     NUMBER,
  bytes    NUMBER,
  breaks   NUMBER
);

rem
rem save_dispatcher
rem
CREATE TABLE save_dispatcher
(
  hid      NUMBER,
  run      NUMBER,
  paddr    raw(4),
  msgs     NUMBER,
  bytes    NUMBER,
  breaks   NUMBER,
  idle     NUMBER,
  busy     NUMBER
);

rem
rem save_server
rem
CREATE TABLE save_server
(
  hid      NUMBER,
  run      NUMBER,
  name     VARCHAR2(20),
  msgs     NUMBER,
  bytes    NUMBER,
  breaks   NUMBER,
  idle     NUMBER,
  busy     NUMBER,
  requests NUMBER
);

rem
rem save_queue
rem
CREATE TABLE save_queue
(
  hid      NUMBER,
  run      NUMBER,
  paddr    raw(4),
  wait     NUMBER,
  totalq   NUMBER
);

rem
rem tmp_sysstat corresponds to v$sysstat
rem
CREATE TABLE tmp_sysstat
(
  hid      NUMBER,
  state    VARCHAR2(10),
  statistic# NUMBER,
  value    NUMBER
);

rem
rem tmp_latch corresponds to v$latch
rem
CREATE TABLE tmp_latch
(
  hid      NUMBER,
  state    VARCHAR2(10),
  latch#   NUMBER,
  gets     NUMBER,
  misses   NUMBER,
  sleeps   NUMBER,
  immediate_gets NUMBER,
  immediate_misses NUMBER
);

rem
rem tmp_filestat corresponds to v$filestat
rem
CREATE TABLE tmp_filestat
(
  hid      NUMBER,
  state    VARCHAR2(10),
  FILE#    NUMBER,
  PHYRDS   NUMBER,

```

```

PHYWRTS    NUMBER,
PHYBLKRD   NUMBER,
PHYBLKWRT  NUMBER,
READTIM    NUMBER,
WRITETIM   NUMBER
);
rem
rem tmp_rollstat corresponds to v$rollstat
rem
CREATE TABLE tmp_rollstat
(
  hid        NUMBER,
  state      VARCHAR2(10),
  USN        NUMBER,
  EXTENTS    NUMBER,
  RSSIZE     NUMBER,
  WRITES     NUMBER,
  XACTS      NUMBER,
  GETS       NUMBER,
  WAITS      NUMBER,
  OPTSIZE    NUMBER,
  HWMSIZE    NUMBER,
  SHRINKS    NUMBER,
  WRAPS      NUMBER,
  EXTENDS    NUMBER,
  AVESHINK   NUMBER,
  AVEACTIVE  NUMBER
);
rem
rem tmp_rowcache corresponds to v$rowcache
rem
CREATE TABLE tmp_rowcache
(
  hid        NUMBER,
  state      VARCHAR2(10),
  cache#     NUMBER,
  type       VARCHAR2(11),
  subordinate# NUMBER,
  parameter  VARCHAR2(32),
  count      NUMBER,
  usage      NUMBER,
  fixed      NUMBER,
  gets       NUMBER,
  getmisses  NUMBER,
  scans      NUMBER,
  scanmisses NUMBER,
  scancompletes NUMBER,
  modifications NUMBER,
  flushes    NUMBER
);
rem
rem tmp_wait corresponds to v$wait_stat
rem
CREATE TABLE tmp_wait
(
  hid        NUMBER,
  state      VARCHAR2(10),
  class     VARCHAR2(18),
  count     NUMBER,
  time      NUMBER
);
rem
rem tmp_fwait corresponds to X$KCBFWAIT
rem
CREATE TABLE tmp_fwait
(
  hid        NUMBER,
  state      VARCHAR2(10),
  addr       VARCHAR2(20),
  indx       NUMBER,
  count      NUMBER,
  time       NUMBER
);
rem
rem tmp_event corresponds to v$system_event
rem
CREATE TABLE tmp_event
(
  hid        NUMBER,
  state      VARCHAR2(10),
  event      VARCHAR2(64),
  total_waits NUMBER,
  time_waited NUMBER,
  average_wait NUMBER
);
rem
rem tmp_fping corresponds to file_ping
rem
CREATE TABLE tmp_fping
(
  hid        NUMBER,
  state      VARCHAR2(10),
  file_id    NUMBER,
  file_name  VARCHAR2(257),
  ts_name    VARCHAR2(30),
  x_to_n     NUMBER
);
rem
rem tmp_fping2 corresponds to file_ping with
extended ping stats
rem
CREATE TABLE tmp_fping2
(
  hid        NUMBER,
  state      VARCHAR2(10),
  file_id    NUMBER,
  file_name  VARCHAR2(257),
  ts_name    VARCHAR2(30),
  x2n        NUMBER,
  x2s        NUMBER,
  x2ssx      NUMBER,
  s2n        NUMBER,
  cic        NUMBER,
  crt        NUMBER,
  hping      NUMBER,
  sping      NUMBER
);
rem
rem tmp_kclwait corresponds to v$kclwait
rem
CREATE TABLE tmp_kclwait
(
  hid        NUMBER,
  state      VARCHAR2(10),
  indx       NUMBER,
  pings      NUMBER,
  hpings     NUMBER,
  spings     NUMBER,
  wpings     NUMBER
);
rem
rem blockclass contains mapping from KCBC
index to text
rem
CREATE TABLE blockclass
(
  indx       NUMBER,
  name       VARCHAR2(24)
);
INSERT INTO blockclass VALUES (0, 'NONE');
INSERT INTO blockclass VALUES (1, 'DATA');
INSERT INTO blockclass VALUES (2, 'SORT');
INSERT INTO blockclass VALUES (3, 'SAVE
UNDO');
INSERT INTO blockclass VALUES (4, 'SEG
HDR');
INSERT INTO blockclass VALUES (5, 'SAVE
UNDO SEG HDR');
INSERT INTO blockclass VALUES (6, 'FREE
LIST|EXTENT MAP');
INSERT INTO blockclass VALUES (7, 'UNDO
HDR');
INSERT INTO blockclass VALUES (8, 'UNDO');
INSERT INTO blockclass VALUES (9, 'UNDO
HDR');
INSERT INTO blockclass VALUES (10, 'UNDO');
rem Are there more...??
rem
rem tmp_lockact corresponds to v$lock_activity
rem
CREATE TABLE tmp_lockact
(
  hid        NUMBER,
  state      VARCHAR2(10),
  from_val   VARCHAR2(4),
  to_val     VARCHAR2(4),
  action_val VARCHAR2(51),
  counter    NUMBER
);
rem
rem zero_lockact corresponds to v$lock_activity
with no activity
rem
CREATE TABLE zero_lockact
(
  from_val   VARCHAR2(4),
  to_val     VARCHAR2(4),
  action_val VARCHAR2(51),
  counter    NUMBER
);
INSERT INTO zero_lockact (from_val, to_val,
action_val, counter) VALUES
('NULL', 'S', 'Lock buffers for read',
0);
INSERT INTO zero_lockact (from_val, to_val,
action_val, counter) VALUES
('NULL', 'X', 'Lock buffers for write',
0);
INSERT INTO zero_lockact (from_val, to_val,
action_val, counter) VALUES
('S', 'NULL', 'Make buffers CR (no write)',
0);
INSERT INTO zero_lockact (from_val, to_val,
action_val, counter) VALUES
('S', 'X', 'Upgrade read lock to write',
0);
INSERT INTO zero_lockact (from_val, to_val,
action_val, counter) VALUES
('X', 'NULL', 'Make buffers CR (write dirty
buffers)', 0);
INSERT INTO zero_lockact (from_val, to_val,
action_val, counter) VALUES
('X', 'S', 'Downgrade write lock to read (write
dirty buffers)', 0);
INSERT INTO zero_lockact (from_val, to_val,
action_val, counter) VALUES

```

```

('X', 'SSX', 'Write transaction table/undo
blocks', 0);
INSERT INTO zero_lockact (from_val, to_val,
action_val, counter) VALUES
('SSX', 'NULL', 'Transaction table/undo blocks
(write dirty buffers)', 0);
INSERT INTO zero_lockact (from_val, to_val,
action_val, counter) VALUES
('SSX', 'S', 'Make transaction table/undo
block available share', 0);
INSERT INTO zero_lockact (from_val, to_val,
action_val, counter) VALUES
('SSX', 'X', 'Rearm transaction table write
mechanism', 0);
rem
rem tmp_sqlarea corresponds to v$sqlarea
rem
CREATE TABLE tmp_sqlarea
(
  hid          NUMBER,
  state        VARCHAR2(10),
  sql_text     VARCHAR2(1000),
  executions   NUMBER,
  buffer_gets  NUMBER,
  disk_reads  NUMBER,
  serializable_aborts  NUMBER
);
rem
rem tmp_time records begin and end time
rem
CREATE TABLE tmp_time
(
  hid          NUMBER,
  state        VARCHAR2(10),
  timestamp    DATE
);
CREATE TABLE tmp_circuit
(
  hid          NUMBER,
  state        VARCHAR2(10),
  circuit      raw(4),
  msg0         NUMBER,
  msg1         NUMBER,
  msgs         NUMBER,
  bytes        NUMBER,
  breaks       NUMBER
);
CREATE TABLE tmp_dispatcher
(
  hid          NUMBER,
  state        VARCHAR2(10),
  paddr        raw(4),
  msgs         NUMBER,
  bytes        NUMBER,
  breaks       NUMBER,
  idle         NUMBER,
  busy         NUMBER
);
CREATE TABLE tmp_server
(
  hid          NUMBER,
  state        VARCHAR2(10),
  name         VARCHAR2(20),
  msgs         NUMBER,
  bytes        NUMBER,
  breaks       NUMBER,
  idle         NUMBER,
  busy         NUMBER,
  requests     NUMBER

```

```

);
CREATE TABLE tmp_queue
(
  hid          NUMBER,
  state        VARCHAR2(10),
  paddr        raw(4),
  wait         NUMBER,
  totalq       NUMBER
);
rem
rem tmp_rsrc corresponds to
v$rsrc_consumer_group
rem
CREATE TABLE tmp_rsrc
(
  hid          NUMBER,
  state        VARCHAR2(10),
  NAME         VARCHAR2(32),
  ACTIVE_SESSIONS  NUMBER,
  EXECUTION_WAITERS  NUMBER,
  REQUESTS         NUMBER,
  CPU_WAIT_TIME    NUMBER,
  CPU_WAITS        NUMBER,
  CONSUMED_CPU_TIME  NUMBER,
  YIELDS           NUMBER,
  SESSIONS_QUEUED  NUMBER
);
COMMIT;
SET ECHO OFF;
:
:
plsq_mon.sql
:
:
rem
rem
=====+
rem Copyright (c) 1995 Oracle Corp,
Redwood Shores, CA |
rem OPEN SYSTEMS
PERFORMANCE GROUP |
rem All Rights Reserved
|
rem
=====+
rem FILENAME
rem plsq_mon.sql
rem DESCRIPTION
rem SQL script to create a stored package for
PL/SQL stored
rem procedures to dump messages.
rem
=====+
rem
rem Usage: sqlplus tpcc/tpcc @plsq_mon
rem
connect tpcc/tpcc;
set echo on;
CREATE OR REPLACE PACKAGE
plsq_mon_pack
IS
  PROCEDURE print
  (
    info  VARCHAR2
  );
END;
/
show errors;

```

```

CREATE OR REPLACE PACKAGE BODY
plsq_mon_pack
IS
  PROCEDURE print
  (
    info  VARCHAR2
  )
  IS
    s     NUMBER;
  BEGIN
    dbms_pipe.pack_message (info);
    s := dbms_pipe.send_message ('plsq_mon');
    IF (s <> 0) THEN
      raise_application_error (-20000, 'Error: ||
to_char(s) ||
      ' sending on pipe');
    END IF;
  END;
END;
/
show errors;
set echo off;
:
:
pst_c.sql
:
:
rem
rem
rem
=====+
rem Copyright (c) 1992 Oracle Corp,
Belmont, CA |
rem OPEN SYSTEMS
PERFORMANCE GROUP |
rem All Rights Reserved
|
rem
=====+
rem FILENAME
rem pst_c.sql
rem DESCRIPTION
rem Create Table for OS Specific Process
Stats
rem
=====+
rem
rem Tables for Unix-specific process statistics
rem
rem Usage: sqlplus internal/internal @pst_c
rem
connect tpcc/tpcc;
set echo on;
DROP TABLE proc_resource;
DROP TABLE os_stat;
rem
rem Resource usage for a process.
rem
CREATE TABLE proc_resource
(
  config  VARCHAR2(10),
  run     NUMBER,
  proc    NUMBER,
  child   NUMBER,
  user_cpu_ms  NUMBER,
  system_cpu_ms  NUMBER,
  maxrss  NUMBER,
  pagein  NUMBER,

```

```

reclaim    NUMBER,
zerofill  NUMBER,
pffincr   NUMBER,
pffdecr   NUMBER,
swap      NUMBER,
syscall   NUMBER,
volcsw    NUMBER,
involcsw  NUMBER,
signal    NUMBER,
lread     NUMBER,
lwrite    NUMBER,
bread     NUMBER,
bwrite    NUMBER,
phread    NUMBER,
phwrite   NUMBER
);
rem
rem OS statistics.
rem These results are from the measurement
interval only.
rem
CREATE TABLE os_stat
(
  config    VARCHAR2(10),
  run       NUMBER,
  hid       NUMBER,
  syscall   NUMBER,
  intr      NUMBER,
  cswitch   NUMBER,
  pagefault NUMBER,
  usr       NUMBER,
  sys       NUMBER,
  idl       NUMBER,
  wio       NUMBER
);
set echo off;
:-----:
require_vars.sh
:-----:
#!/bin/sh
# make sure each environment variable argument
is defined.
for var in $* ; do
  if test -z "$var" ; then
    echo Missing variable: $var - exiting.
    exit 0
  fi
done
exit 1
:-----:
shutdowndb.sh
:-----:
#!/bin/sh
echo "Shutting down database..."
$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect
spool shutdowndb.log;
set echo on;
alter system switch logfile;
alter system switch logfile;
shutdown immediate;
set echo off;
spool off;
exit
!
:-----:
space.sh
:-----:
NAME=$1

```

```

CNT=$2
SIZE1=$3
SIZE2=$4
CPU=$5
#drop tablespace ${NAME} ;
sqlplus tpcc/tpcc <<! &
spool ${NAME}.0.log ;
create tablespace ${NAME}_0 datafile
'${tpcc_disks_location}/${NAME}_0_0'
size ${SIZE1} reuse extent management local
uniform size ${SIZE2}
segment space management auto nologging ;
exit;
!
wait
l=1
J=0
while [ $l -lt $CNT ]
do
  sqlplus tpcc/tpcc <<! &
  spool ${NAME}.$l.log ;
  alter tablespace ${NAME}_0 add datafile
  '${tpcc_disks_location}/${NAME}_0_0_0' size
  ${SIZE1} reuse autoextend on;
  exit;
  !
  l=`expr $l + 1`
  J=`expr $J + 1`
  if test $J -eq $CPU
  then
    wait
    J=0
  fi
done
wait
:-----:
space_16.sh
:-----:
NAME=$1
CNT=$2
SIZE1=$3
SIZE2=$4
CPU=$5
#drop tablespace ${NAME} ;
sqlplus tpcc/tpcc <<! &
spool ${NAME}.0.log ;
create tablespace ${NAME}_0 datafile
'${tpcc_disks_location}/${NAME}_0_0'
size ${SIZE1} reuse extent management local
uniform size ${SIZE2}
blocksize 16K segment space management auto
nologging ;
exit;
!
wait
l=1
J=0
while [ $l -lt $CNT ]
do
  sqlplus tpcc/tpcc <<! &
  spool ${NAME}.$l.log ;
  alter tablespace ${NAME}_0 add datafile
  '${tpcc_disks_location}/${NAME}_0_0_0' size
  ${SIZE1} reuse autoextend on;
  exit;
  !
  l=`expr $l + 1`
  J=`expr $J + 1`
  if test $J -eq $CPU

```

```

then
  wait
  J=0
fi
done
wait
:-----:
space_2.sh
:-----:
NAME=$1
CNT=$2
SIZE1=$3
SIZE2=$4
CPU=$5
#drop tablespace ${NAME} ;
sqlplus tpcc/tpcc <<! &
spool ${NAME}.0.log ;
create tablespace ${NAME}_0 datafile
'${tpcc_disks_location}/${NAME}_0_0'
size ${SIZE1} reuse extent management local
uniform size ${SIZE2}
blocksize 2K segment space management auto
nologging ;
exit;
!
wait
l=1
J=0
while [ $l -lt $CNT ]
do
  sqlplus tpcc/tpcc <<! &
  spool ${NAME}.$l.log ;
  alter tablespace ${NAME}_0 add datafile
  '${tpcc_disks_location}/${NAME}_0_0_0' size
  ${SIZE1} reuse autoextend on;
  exit;
  !
  l=`expr $l + 1`
  J=`expr $J + 1`
  if test $J -eq $CPU
  then
    wait
    J=0
  fi
done
wait
:-----:
space_get.sql
:-----:
REM=====
=====+
REM      Copyright (c) 1995 Oracle Corp,
Redwood Shores, CA |
REM      OPEN SYSTEMS
PERFORMANCE GROUP |
REM      All Rights Reserved
|
REM=====
=====+
REM FILENAME
REM space_get.sql
REM DESCRIPTION
REM Get sizes of tables, indexes and
tablespaces.
REM Usage: sqlplus 'sys/change_on_install as
sysdba' @space_get [<tpm> <# of warehouses>]
REM=====
=====*/
set echo on;

```

```

delete from tpcc_data;
delete from tpcc_space;
delete from tpcc_totospace;
insert into tpcc_data
select substr(segment_name,1,18),
substr(segment_type,1,15),
sum(blocks), t.block_size,
round(sum(blocks) * 0.05), 0,
sum(blocks) + round(sum(blocks) * 0.05)
from dba_extents e, dba_tablespaces t
where owner = 'TPCC' AND ( segment_type =
'INDEX' OR
segment_type = 'INDEX PARTITION' OR
segment_type = 'CLUSTER'
OR segment_type = 'TABLE' OR
segment_type = 'TABLE PARTITION')
AND e.tablespace_name <> 'SYSTEM'
AND e.tablespace_name <> 'SP_0'
AND e.tablespace_name =
t.tablespace_name
group by segment_name, segment_type,
t.block_size;
insert into tpcc_data
select 'SYSTEM', 'SYS', sum(blocks),
t.block_size, 0, 0, sum(blocks)
from dba_data_files f, dba_tablespaces t
where f.tablespace_name = 'SYSTEM' and
t.tablespace_name = f.tablespace_name
group by t.block_size;
insert into tpcc_data
select 'SYSAUX', 'SYS', sum(blocks),
t.block_size, 0, 0, sum(blocks)
from dba_data_files f, dba_tablespaces t
where f.tablespace_name = 'SYSAUX' and
t.tablespace_name = f.tablespace_name
group by t.block_size;
insert into tpcc_data
select 'ROLL_SEG', 'SYS', sum(blocks),
t.block_size, 0, 0, sum(blocks)
from dba_data_files f, dba_tablespaces t
where f.tablespace_name like
'%UNDO_TS%' and f.tablespace_name =
t.tablespace_name
group by f.tablespace_name, t.block_size;
insert into tpcc_data
select 'DB_STAT', 'SYS', sum(blocks),
t.block_size, 0, 0, sum(blocks)
from dba_data_files f, dba_tablespaces t
where f.tablespace_name like '%SP_0%' and
f.tablespace_name = t.tablespace_name
group by f.tablespace_name, t.block_size;
update tpcc_data
set five_pct = 0,
daily_grow = round(blocks * &&1 / 62.5 /
&&2),
total = blocks + round(blocks * &&1 / 62.5 /
&&2)
where segment = 'HIST' OR segment =
'ORDRCLUSTER_QUEUE' OR
segment = 'IORDL';
insert into tpcc_space
select substr(ex$.name,1,18),
sum(sp$.sz_blocks), sp$.block_size, 0, 0, 0
from
(select f.tablespace_name , sum(blocks)
sz_blocks, t.block_size block_size
from dba_data_files f, dba_tablespaces t
where f.tablespace_name <> 'SYSTEM' and
f.tablespace_name = t.tablespace_name

```

```

group by f.tablespace_name, t.block_size
) sp$,
(select distinct tablespace_name,
segment_name name
from dba_extents
where owner = 'TPCC'
and (segment_type = 'CLUSTER' or
segment_type = 'TABLE'
or segment_type = 'TABLE PARTITION' or
segment_type = 'INDEX'
or segment_type = 'INDEX PARTITION')
and tablespace_name <> 'SYSTEM'
) ex$
where sp$.tablespace_name =
ex$.tablespace_name
group by ex$.name, sp$.block_size;
insert into tpcc_space
select substr(f.tablespace_name,1,18),
sum(blocks), t.block_size, 0, 0, 0
from dba_data_files f, dba_tablespaces t
where (f.tablespace_name = 'SYSTEM' or
f.tablespace_name = 'SYSAUX')
and f.tablespace_name =
t.tablespace_name
group by f.tablespace_name, t.block_size;
insert into tpcc_space
select 'ROLL_SEG', sum(blocks), t.block_size, 0,
0, 0
from dba_data_files f, dba_tablespaces t
where f.tablespace_name = 'UNDO_TS' and
f.tablespace_name = t.tablespace_name
group by f.tablespace_name, t.block_size;
insert into tpcc_space
update tpcc_space
set required =
(
select sum(total)
from tpcc_data
where tpcc_data.segment =
tpcc_space.segment
)
where segment in
(
select segment from tpcc_data
);
update tpcc_space
set static =
(
select sum(total)
from tpcc_data
where tpcc_data.segment =
tpcc_space.segment
)
where segment in
(
select segment from tpcc_data
);
update tpcc_space
set static = 0,
dynamic =
(
select sum(blocks)
from tpcc_data

```

```

where tpcc_data.segment =
tpcc_space.segment
)
where segment in ('HIST',
'ORDRCLUSTER_QUEUE', 'IORDL');
update tpcc_space
set oversize = blocks - required;
insert into tpcc_totospace
select &&1, &&2, sum(static *
block_size)/1024, sum(dynamic *
block_size)/1024, sum(oversize *
block_size)/1024, 0, 0, 0
from tpcc_space;
update tpcc_totospace
set daily_grow =
(
select sum(daily_grow * block_size)/1024
from tpcc_data
);
update tpcc_totospace
set space60 = static + 60 * daily_grow;
set echo off;
:-----:
space_init.sql
:-----:
REM=====
=====+
REM FILENAME
REM space_init.sql
REM DESCRIPTION
REM Create tables for space calculations.
REM Usage: sqlplus 'sys/change_on_install as
sysdba' @space_init.sql
REM=====
=====*/
set echo on;
drop table tpcc_data;
drop table tpcc_space;
drop table tpcc_totospace;
create table tpcc_data (
segment varchar2(18),
type varchar2(15),
blocks number,
block_size number,
five_pct number,
daily_grow number,
total number
);
create table tpcc_space (
segment varchar2(18),
blocks number,
block_size number,
required number,
static number,
dynamic number,
oversize number
);
create table tpcc_totospace (
tpm number,
nware number,
static number,
dynamic number,
oversize number,
daily_grow number,
daily_spre number,
space60 number
);
create unique index tpcc_data on tpcc_data
(segment);

```

```

    create unique index tpcc_space on tpcc_space
(segment);
    set echo off;

:.....:
space_temp.sh
:.....:

NAME=$1
CNT=$2
SIZE1=$3
SIZE2=$4
CPU=$5

#drop tablespace ${NAME} ;
sqlplus tpcc/tpcc <<! &
spool ${NAME}.0.log ;
create temporary tablespace ${NAME} tempfile
'${tpcc_disks_location}/${NAME}_0_0'
size ${SIZE1} reuse extent management local
uniform size ${SIZE2} ;
exit;
!
wait

I=1
J=0
while [ $I -lt $CNT ]
do
    sqlplus tpcc/tpcc <<! &
    spool ${NAME}.$I.log ;
    alter tablespace ${NAME} add tempfile
'${tpcc_disks_location}/${NAME}_0_$I' size
${SIZE1} reuse ;
    exit;
!
I=`expr $I + 1`
J=`expr $J + 1`
if test $J -eq $CPU
then
    wait
    J=0
fi
done
wait
:.....:
startpdb.sh
:.....:
#!/bin/sh
echo "Starting up database using $1..."
init_file=${1}.ora
if test $tpcc_np -gt 1 ; then
    init_file=build_init_${tpcc_rac_id}.ora
fi
$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect
spool startpdb.log
set echo on
startup pfile=$init_file open
spool off
set echo off
exit sql.sqlcode
!
:.....:
stepenv.sh
:.....:
# forces any env variables we set to be exported
set -a
tpcc_kit=t

```

```

tpcc_bench=/ora/Ora10G/tpcc-kit
tpcc_scripts=$tpcc_bench/scripts
tpcc_require=$tpcc_scripts/require_vars.sh
tpcc_lcm=$tpcc_scripts/lcm.sh
tpcc_tokilobytes=$tpcc_scripts/tokilobytes.sh
tpcc_fromkilobytes=$tpcc_scripts/fromkilobytes.sh
tpcc_estsize=$tpcc_scripts/estsize.sh
tpcc_notneg=$tpcc_scripts/notneg.sh
tpcc_isneg=$tpcc_scripts/isneg.sh
# need a better way to check for bc, may
# resort to checking each directory in path
# if this doesn't work
#11/7/02 - alex.ni this is causing too many
problems
#because systems have bc in some odd place.
typically
#mangled cygwin installs w/ mksnt/cygwin mixes
#if test -x /usr/bin/bc -o -x /bin/bc; then
tpcc_bcexpr=$tpcc_scripts/bcexpr.sh
#else
#tpcc_bcexpr=expr
#fi
# the ksh version is a bit faster, so we want
# to use it if we have ksh. Otherwise we have
# a compatible version.
#if test -x /bin/ksh; then
#tpcc_createts=$tpcc_scripts/createts.ksh
#else
tpcc_createts=$tpcc_scripts/createts.sh
#fi
tpcc_tabledata=$tpcc_scripts/taledata.sh
tpcc_load=$tpcc_bench/benchrun/bin/tpccload.ex
e
tpcc_createtablespace=$tpcc_scripts/createtable
spaces.sh
##
tpcc_sqlplus=cat
tpcc_sqlplus_args="/nolog"
tpcc_internal_connect='connect / as sysdba'
tpcc_user_pass='tpcc/tpcc'
tpcc_dba_user_pass='system/manager'
oracle_dba=system
oracle_dba_password=manager
tpcc_sqlplus_args=
tpcc_user_pass=
tpcc_sqlplus=sqlplus
tpcc_user_pass='tpcc/tpcc'
# import options generated by gui
. $(tpcc_bench)/options.sh
#8gb oracle filesize limit (in k)
tpcc_fsize_limit_k=8388608
#2gb - 1k oracle extent limit (in k)
tpcc_extent_limit_k=2097151
# Runlen calculations should be in hours, but
# this was the old calculation, which assumed
# minutes, and also 8 times:
# tpcc_runlen= $tpcc_bcexpr 8 \* 60 \*
$tpcc_runlen
# we just want to keep the value as it is.
tpcc_system_size=200M
tpcc_logfile_size=$tpcc_bcexpr 20 +
\($tpcc_scale\)M
tpcc_undo_size=$tpcc_bcexpr 2 \* $tpcc_scale
if test $tpcc_undo_size -gt 8096; then
    tpcc_undo_size=8096
fi
tpcc_undo_size="$tpcc_undo_size)M"
tpcc_undo_bs=8K

```

```

tpcc_statspack_size= $tpcc_bcexpr 1 \*
$tpcc_scale
if test $tpcc_statspack_size -gt 2048; then
    tpcc_statspack_size=2048
fi
tpcc_statspack_size="$tpcc_statspack_size)M"
tpcc_sysaux_size=120M
# fixed table params
#table list (note temp is always at the end since it
may use numbers from other tables, and it's not
included in these lists)
tpcc_table_list='ware cust dist hist stok item ordr
ordl nord'
tpcc_index_list='ware icust1 icust2 idist istok iitem
iordr1 iordr2 iordl inord'
#for these I use average row length, calculated
from multi-blocksize stats.
#we figure out how many new rows we will gain in
a run (in createtablespace.sh)
#and add that much to the base tablespace size.
tpcc_hist_growth=51
tpcc_ordr_growth=35
tpcc_nord_growth=13
#tpcc_ordl_growth=660
tpcc_ordl_growth=900
#i started indices at 1/10th... need an exact figure
tpcc_iordr1_growth=20
tpcc_iordr2_growth=20
tpcc_iordl_growth=66
tpcc_inord_growth=2
tpcc_item_growth=0
tpcc_iitem_growth=0
tpcc_temp_growth=0
tpcc_cust_growth=regular
tpcc_icust1_growth=regular
tpcc_icust2_growth=regular
tpcc_stok_growth=regular
tpcc_istok_growth=regular
tpcc_ware_growth=regular
tpcc_iware_growth=regular
tpcc_dist_growth=regular
tpcc_idist_growth=regular
# minimum size of temp tablespace
tpcc_tempts_min=10240
# for Linux, set appropriate tablespace heuristics
# to set high io tables to have 64 files, and
minimize
# others.
if expr $tpcc_os = linux > /dev/null; then
# for table in $tpcc_table_list $tpcc_index_list
temp; do
# eval "tpcc_${table}_tsfileinc=1"
# done
    tpcc_os=unix
# tpcc_stok_tsfileinc=64
# tpcc_cust_tsfileinc=64
# tpcc_iordl2_tsfileinc=16
# tpcc_icust2_tsfileinc=16
# tpcc_iordl_tsfileinc=16
else
#in case someone changes out of linux, and the
shell is stuck
    for table in $tpcc_table_list $tpcc_index_list
temp; do
        eval "tpcc_${table}_tsfileinc="
    done
fi
tpcc_stok_tsfileinc=
tpcc_cust_tsfileinc=

```

```

tpcc_iordl2_tsfileinc=
tpcc_icust2_tsfileinc=
tpcc_iordl_tsfileinc=
#fi
# import local options
. ${tpcc_bench}/localoptions.sh
if expr `echo x${tpcc_no_options}` = xt > /dev/null;
then
    echo Please modify
    ${tpcc_bench}/localoptions.sh to configure the
    generator.
    exit 1
fi
tpcc_fixordrordl=${tpcc_genscripts_dir}/loadfixordr
ordl.sh
tpcc_updateordrordl=${tpcc_scripts}/updateordrordl
dl.sh
#tp- get table param. (that is,
$tpcc_tablename_tableparam)
tp(){
    eval echo `"$tpcc_$1_$2"`
}
# automatically generated variables
if expr `echo $tpcc_version | cut -b1` = t >
/dev/null; then
    tpcc_auto_undo=t
else
    tpcc_auto_undo=f
fi
if expr `echo $tpcc_version | cut -b2` = t >
/dev/null; then
    tpcc_autospace_avail=t
else
    tpcc_autospace_avail=f
fi
if expr `echo $tpcc_version | cut -b3` = t >
/dev/null; then
    tpcc_queue_avail=t
    tpcc_use_sysaux=t
else
    tpcc_queue_avail=f
    tpcc_use_sysaux=f
fi
# for NT, ORACLE does not like $variables in sql
scripts, so we must
# hardcode these things for it.
if test x${tpcc_os} = xnt; then
    tpcc_hardcode=t
else
    tpcc_hardcode=f
fi
# if this is unset we need to make sure it's
something anyway
if test x${tpcc_defbs} = x; then
    tpcc_defbs=2
fi
# used for loading program
if test x${tpcc_hash_overflow} = xt; then
    tpcc_hash_overflow=t
else
    unset tpcc_hash_overflow
fi
if test x${tpcc_overflow} = xt; then
    tpcc_hash_overflow=t
else
    unset tpcc_hash_overflow
fi
tpcc_create_steps="buildtpccflags buildora
buildcreatets buildcreatedb \

```

```

buildcreatetable-ware buildcreatetable-cust
buildcreatetable-dist buildcreatetable-hist
buildcreatetable-stok buildcreatetable-item
buildcreatetable-ordr buildcreatetable-ordl
buildcreatetable-nord \
buildloadware buildloadaddit buildloaditem
buildloadhist buildloadnord buildloadordrordl
buildloadcust buildloadstok buildfixoo \
buildcreateindex-iware buildcreateindex-icust1
buildcreateindex-icust2 buildcreateindex-idist
buildcreateindex-istok buildcreateindex-iitem
buildcreateindex-iordr1 buildcreateindex-iordr2
buildcreateindex-iordl buildcreateindex-inord \
buildstoreprosql buildspacestats listfiles
"
# remove runscript-loadfixordrordl - shuang,
030626
tpcc_steps="runsqllocal-createdb shutdownnb
startupdb-p_build createuser runscript-createts
assigntemp ddview \
    runsql-createtable_ware runsql-createtable_cust
runsql-createtable_dist runsql-createtable_hist
runsql-createtable_stok runsql-createtable_item
runsql-createtable_ordr runsql-createtable_ordl
runsql-createtable_nord \
runscript-loadware runscript-loadaddit runscript-
loaditem runscript-loadhist runscript-loadnord
runscript-loadordrordl runscript-loadcust runscript-
loadstok \
runsql-createindex_iware runsql-
createindex_icust1 runsql-createindex_icust2
runsql-createindex_idist runsql-createindex_istok
runsql-createindex_iitem runsql-
createindex_iordr1 runsql-createindex_iordr2
runsql-createindex_iordl runsql-createindex_inord
\
analyze createstats createstoredprocs
createspacestats createmisc"
# no longer automatically exports env variables
set +a
# check for problems with configuration
badconf=
for table in $tpcc_table_list; do
    if expr `tp $table imp` = queue > /dev/null; then
        if expr $tpcc_queue_avail = f > /dev/null; then
            echo Table $table may not be a queue,
            since queues are
            echo are unavailable in the selected Oracle
            version.
            badconf=t
        fi
    fi
    if expr $tpcc_autospace_avail = f \& `tp $table
autospace` = t > /dev/null; then
        echo Table $table may not use bitmapped
        space management
        echo since it is not available in the selected
        Oracle version.
        badconf=t
    fi
done
if test -n "$badconf"; then
    exit 1
fi
# make sure we have everything
if $tpcc_require ORACLE_SID \
tpcc_tokilobytes tpcc_createts tpcc_lcm\
tpcc_sqlplus tpcc_internal_connect\

```

```

tpcc_np tpcc_cpu tpcc_os tpcc_runlen tpcc_ldrive
tpcc_scale tpcc_disks_location tpcc_auto_undo
tpcc_tempt_min\
tpcc_system_size tpcc_logfile_size\
tpcc_undo_size tpcc_undo_bs\
oracle_dba oracle_dba_password
tpcc_dba_user_pass
then exit 1; fi
if test x${tpcc_hardcode} != xt; then
    tpcc_disks_location=${tpcc_disks_location}/
# tpcc_sql_dir=${tpcc_sql_dir}
# tpcc_statspack_size=${tpcc_statspack_size}
# tpcc_genscripts_dir=${tpcc_genscripts_dir}
fi
:
:
:
tabledata.sh
:
:
# tabledata- (not a script) data and functions for
# getting columns and info for creating tables and
indices.
# forces any env variables we set to be exported
set -a
# note- clustcols will be an option- move to
stepenv file!
#tp- get table param. (that is,
$tpcc_tablename_tableparam)
tp(){
    eval echo `"$tpcc_$1_$2"`
}
#nullauto- check if $1 null or auto, if not echo it,
otherwise echo $2.
nullauto(){
    if expr "x$1" = x \ | "x$1" = xauto \ | "x$1" = xno >
/dev/null; then
        echo "$2"
    else
        echo "$1"
    fi
}
tpcc_item_warecol=none
tpcc_item_clustcols=t
tpcc_item_hkey="nullauto "$tpcc_item_hkey"
100000"
if test $tpcc_np -gt 1; then
    tpcc_item_hash_def="(i_id + 1)"
else
    tpcc_item_hash_def="(i_id)"
fi
tpcc_item_hash="nullauto "$tpcc_item_hash"
"$tpcc_item_hash_def"
tpcc_item_rsize="nullauto "$tpcc_item_rsize"
120"
#The actual size of item needs to be a little bigger
just because
#it's around boundary size for hash clusters.
tpcc_item_crsz=130
tpcc_item_cols="cat <<|
    i_id-number(6,0)
    i_name-varchar2(24)
    i_price-number
    i_data-varchar2(50)
    i_im_id-number
!
~
tpcc_iitem_cols=$tpcc_item_cols
tpcc_iitem_warecol=none
tpcc_iitem_indexon=item
tpcc_stok_row_factor=100000

```

```

tpcc_stok_rac_storage=t
tpcc_stok_2_RowPerBlock=5
if expr x$tpcc_compress = xt > /dev/null; then
  tpcc_stok_2_RowPerBlock=7
fi
tpcc_stok_4_RowPerBlock=11
if expr x$tpcc_overflow = xt > /dev/null; then
  if expr x$tpcc_compress = xt > /dev/null; then
    tpcc_stok_4_RowPerBlock=15
  fi
fi
tpcc_stok_8_RowPerBlock=23
if expr x$tpcc_overflow = xt > /dev/null; then
  if expr x$tpcc_compress = xt > /dev/null; then
    tpcc_stok_8_RowPerBlock=32
  fi
fi
if expr x$tpcc_compress = xt > /dev/null; then
  tpcc_stok_compress=ffffffffffffff
  tpcc_stok_rsize="nullauto "$tpcc_stok_rsize"
  256"
  if expr "x$tpcc_ieeee_number" = "xt" >/dev/null;
then
    tpcc_stok_rsize=290
  fi
tpcc_stok_cols= cat <<!
  s_i_id-number
  s_w_id-number
  s_quantity-number
  s_ytd-number
  s_order_cnt-number
  s_remote_cnt-number
  s_data-varchar2(50)
  s_dist_01-varchar2(24)
  s_dist_02-varchar2(24)
  s_dist_03-varchar2(24)
  s_dist_04-varchar2(24)
  s_dist_05-varchar2(24)
  s_dist_06-varchar2(24)
  s_dist_07-varchar2(24)
  s_dist_08-varchar2(24)
  s_dist_09-varchar2(24)
  s_dist_10-varchar2(24)
!
,
else
tpcc_stok_rsize="nullauto "$tpcc_stok_rsize"
350"
tpcc_stok_cols= cat <<!
  s_i_id-number
  s_w_id-number
  s_quantity-number
  s_ytd-number
  s_order_cnt-number
  s_remote_cnt-number
  s_data-varchar2(50)
  s_dist_01-char(24)
  s_dist_02-char(24)
  s_dist_03-char(24)
  s_dist_04-char(24)
  s_dist_05-char(24)
  s_dist_06-char(24)
  s_dist_07-char(24)
  s_dist_08-char(24)
  s_dist_09-char(24)
  s_dist_10-char(24)
!
,
fi

```

```

tpcc_stok_clustcols=tt
tpcc_stok_hkey_def= $tpcc_bcexpr 100000 \\\
$tpcc_scale`
tpcc_stok_hkey="nullauto "$tpcc_stok_hkey"
"$tpcc_stok_hkey_def""
if expr $tpcc_stok_imp = parttable \\\
$tpcc_stok_imp = partiot \\\ $tpcc_np \\\> 1 >
/dev/null; then
  tmpdiv= expr $tpcc_scale / $tpcc_np`
  tpcc_stok_hash_def="(abs(s_i_id - 1) * $tmpdiv +
mod((s_w_id - 1), $tmpdiv) + trunc ((s_w_id - 1) /
$tmpdiv) * $tmpdiv * 100000)"
else
  tpcc_stok_hash_def="(s_i_id * $tpcc_scale +
s_w_id)"
fi
tpcc_stok_hash="nullauto "$tpcc_stok_hash"
"$tpcc_stok_hash_def""
tpcc_stok_warecol=s_w_id
tpcc_istok_cols=$tpcc_stok_cols
tpcc_istok_warecol=$tpcc_stok_warecol
tpcc_istok_indexon=stok
tpcc_ware_row_factor=1
tpcc_ware_2_RowPerBlock=1
tpcc_ware_4_RowPerBlock=1
tpcc_ware_8_RowPerBlock=1
tpcc_ware_rac_storage=t
tpcc_ware_clustcols=t
tpcc_ware_hkey="nullauto "$tpcc_ware_hkey"
"$tpcc_scale""
if test $tpcc_np -gt 1 ; then
  tpcc_ware_hash_def="(w_id - 1)"
else
  tpcc_ware_hash_def="(w_id)"
fi
tpcc_ware_hash="nullauto "$tpcc_ware_hash"
"$tpcc_ware_hash_def""
#now we want default for ware to be auto (one
block always)
#tpcc_ware_rsize="nullauto "$tpcc_ware_rsize"
1536"
tpcc_ware_rsize="nullauto "$tpcc_ware_rsize"
auto"
tpcc_ware_warecol=none
tpcc_ware_cols= cat <<!
  w_id-number(5,0)
  w_ytd-number
  w_tax-number
  w_name-varchar2(10)
  w_street_1-varchar2(20)
  w_street_2-varchar2(20)
  w_city-varchar2(20)
  w_state-char(2)
  w_zip-char(9)
!
,
tpcc_iware_cols=$tpcc_ware_cols
tpcc_iware_warecol=none
tpcc_iware_indexon=ware
tpcc_dist_row_factor=10
tpcc_dist_2_RowPerBlock=1
tpcc_dist_4_RowPerBlock=1
tpcc_dist_8_RowPerBlock=1
tpcc_dist_rac_storage=t
tpcc_dist_clustcols=tt
tpcc_dist_hkey_def= $tpcc_bcexpr 10 \\\
$tpcc_scale`
tpcc_dist_hkey="nullauto "$tpcc_dist_hkey"
"$tpcc_dist_hkey_def"

```

```

if test $tpcc_np -gt 1 ; then
  tpcc_dist_hash_def="(((d_w_id - 1) * 10) +
d_id)"
else
  tpcc_dist_hash_def="((d_w_id * 10) + d_id)"
fi
tpcc_dist_hash="nullauto "$tpcc_dist_hash"
"$tpcc_dist_hash_def""
#tpcc_dist_rsize="nullauto "$tpcc_dist_rsize"
1536"
tpcc_dist_rsize="nullauto "$tpcc_dist_rsize"
auto"
tpcc_dist_warecol=d_w_id
tpcc_dist_cols= cat <<!
  d_id-number
  d_w_id-number
  d_ytd-number
  d_next_o_id-number
  d_tax-number
  d_name-varchar2(10)
  d_street_1-varchar2(20)
  d_street_2-varchar2(20)
  d_city-varchar2(20)
  d_state-char(2)
  d_zip-char(9)
!
,
tpcc_idist_cols=$tpcc_dist_cols
tpcc_idist_warecol=d_w_id
tpcc_idist_indexon=dist
#NOTE- needs to change SQL code
tpcc_ordr_clustcols=tttttttt
tpcc_ordr_queuesort=2t-3t-1t-9tt-
tpcc_ordr_queuenames=2-3-1-
#NOTE- check if this is right for hash keys
tpcc_ordr_hkey_def= $tpcc_bcexpr 10 \\\
$tpcc_scale`
tpcc_ordr_hkey="nullauto "$tpcc_ordr_hkey"
"$tpcc_ordr_hkey_def""
tpcc_ordr_hash="nullauto "$tpcc_ordr_hash"
'(o_w_id - 1) * 10 + o_d_id - 1'"
tpcc_ordr_rsize="nullauto "$tpcc_ordr_rsize"
1490"
tpcc_ordr_warecol=o_w_id
if expr $tpcc_ordr_imp = queue > /dev/null; then
# buildcreatetable sets cols to this after tablecols
# are fetched from normal ord_r_cols, so we can
get the
# o_number field out for queuesort. (bit of a hack,
but
# it would be difficult to redo it another way)
tpcc_ordr_qcols= cat <<!
  o_id-number
  o_w_id-number
  o_d_id-number
  o_c_id-number
  o_carrier_id-number
  o_ol_cnt-number
  o_all_local-number
  o_entry_d-date
  o_number-number
!
,
tpcc_ordr_cols= cat <<!
  o_id-number-sort
  o_w_id-number
  o_d_id-number
  o_c_id-number
  o_carrier_id-number

```

```

o_ol_cnt-number
o_all_local-number
o_entry_d-date
!
.
else
tpcc_ordr_cols=`cat <<!
o_id-number
o_w_id-number
o_d_id-number
o_c_id-number
o_carrier_id-number
o_ol_cnt-number
o_all_local-number
o_entry_d-date
!
.
fi
tpcc_iordr1_warecol=$tpcc_ordr_warecol
tpcc_iordr1_cols=$tpcc_ordr_cols
tpcc_iordr1_indexon=ordr
tpcc_iordr2_warecol=$tpcc_ordr_warecol
tpcc_iordr2_cols=$tpcc_ordr_cols
tpcc_iordr2_indexon=ordr
#ordl uses ordr's cluster, so we represent this
#with another variable
tpcc_ordl_usecluster=', constraint ordl_uk primary
key (ol_w_id, ol_d_id, ol_o_id, ol_number )
CLUSTER ordrcluster_queue(ol_w_id, ol_d_id,
ol_o_id, ol_number)'
tpcc_ordl_othercluster=ordr
tpcc_ordl_warecol=ol_w_id
if test $tpcc_ordl_imp = queue; then
tpcc_ordl_cols=`cat <<!
ol_w_id-number
ol_d_id-number
ol_o_id-number-sort
ol_number-number-sort
ol_i_id-number
ol_delivery_d-date
ol_amount-number
ol_supply_w_id-number
ol_quantity-number
ol_dist_info-char(24)
!
.
else
tpcc_ordl_cols=`cat <<!
ol_w_id-number
ol_d_id-number
ol_o_id-number
ol_number-number
ol_i_id-number
ol_delivery_d-date
ol_amount-number
ol_supply_w_id-number
ol_quantity-number
ol_dist_info-char(24)
!
.
fi
tpcc_iordl_cols=$tpcc_ordl_cols
tpcc_iordl_warecol=$tpcc_ordl_warecol
tpcc_iordl_indexon=ordl
tpcc_nord_clustcols=ttt
tpcc_nord_queuesort=1t-2t-3t-
tpcc_nord_queuenames=1-2-3-
#NOTE- check if this is right for hash keys

```

```

tpcc_nord_hkey_def=`$tpcc_bcexpr 10 \`
$tpcc_scale`
tpcc_nord_hkey="nullauto "$tpcc_nord_hkey"
"$tpcc_nord_hkey_def""
tpcc_nord_hash="nullauto "$tpcc_nord_hash"
'(no_w_id - 1) * 10 + no_d_id - 1'"
tpcc_nord_rsize="nullauto "$tpcc_nord_rsize"
190"
tpcc_nord_warecol=no_w_id
tpcc_nord_qcols=`cat <<!
no_w_id-number
no_d_id-number
no_o_id-number
!
.
tpcc_nord_cols=`cat <<!
no_w_id-number
no_d_id-number
no_o_id-number-sort
!
.
tpcc_inord_cols=$tpcc_nord_cols
tpcc_inord_warecol=$tpcc_nord_warecol
tpcc_inord_indexon=nord
#we regardless of overflow or compress we
always want the
#size calculation for cust to be the same.
tpcc_cust_row_factor=30000
tpcc_cust_2_RowPerBlock=2
if expr x$tpcc_overflow = xt > /dev/null; then
if expr x$tpcc_compress = xt > /dev/null; then
tpcc_cust_2_RowPerBlock=11
else
tpcc_cust_2_RowPerBlock=5
fi
fi
tpcc_cust_4_RowPerBlock=4
if expr x$tpcc_overflow = xt > /dev/null; then
if expr x$tpcc_compress = xt > /dev/null; then
tpcc_cust_4_RowPerBlock=23
else
tpcc_cust_4_RowPerBlock=11
fi
fi
tpcc_cust_8_RowPerBlock=9
if expr x$tpcc_overflow = xt > /dev/null; then
if expr x$tpcc_compress = xt > /dev/null; then
tpcc_cust_8_RowPerBlock=47
else
tpcc_cust_8_RowPerBlock=23
fi
fi
tpcc_cust_rac_storage=t
tpcc_cust_crsz=850
if expr x$tpcc_overflow = xt > /dev/null; then
if expr x$tpcc_compress = xt > /dev/null; then
tpcc_cust_compress=ffffffffttttfffff
tpcc_cust_rsize="nullauto "$tpcc_cust_rsize"
180"
else
tpcc_cust_rsize="nullauto "$tpcc_cust_rsize"
350"
fi
tpcc_cust_cols=`cat <<!
c_id-number
c_d_id-number
c_w_id-number
c_discount-number
c_credit-char(2)

```

```

c_last-varchar2(16)
c_first-varchar2(16)
c_credit_lim-number
c_balance-number
c_ytd_payment-number
c_payment_cnt-number
c_delivery_cnt-number
c_street_1-varchar2(20)
c_street_2-varchar2(20)
c_city-varchar2(20)
c_state-char(2)
c_zip-char(9)
c_phone-char(16)
c_since-date
c_middle-char(2)
c_data-varchar2(500)
!
.
else
tpcc_cust_rsize="nullauto "$tpcc_cust_rsize"
850"
tpcc_cust_cols=`cat <<!
c_id-number
c_d_id-number
c_w_id-number
c_discount-number
c_credit-char(2)
c_last-varchar2(16)
c_first-varchar2(16)
c_credit_lim-number
c_balance-number
c_ytd_payment-number
c_payment_cnt-number
c_delivery_cnt-number
c_street_1-varchar2(20)
c_street_2-varchar2(20)
c_city-varchar2(20)
c_state-char(2)
c_zip-char(9)
c_phone-char(16)
c_since-date
c_middle-char(2)
c_data-varchar2(500)
!
.
fi
tpcc_cust_clustcols=ttt
tpcc_cust_hkey_def=`$tpcc_bcexpr 30000 \`
$tpcc_scale`
tpcc_cust_hkey="nullauto "$tpcc_cust_hkey"
"$tpcc_cust_hkey_def""
if expr $tpcc_np \> 1 > /dev/null; then
tpcc_cust_hash_def="(c_w_id * 30000 + c_id *
10 + c_d_id - 30011)"
else
tpcc_cust_hash_def="(c_id * ($tpcc_scale * 10)
+ c_w_id * 10 + c_d_id)"
fi
tpcc_cust_hash="nullauto "$tpcc_cust_hash"
"$tpcc_cust_hash_def""
tpcc_cust_warecol=c_w_id
tpcc_icust1_cols=$tpcc_cust_cols
tpcc_icust1_warecol=$tpcc_cust_warecol
tpcc_icust1_indexon=cust
tpcc_icust2_cols=$tpcc_cust_cols
tpcc_icust2_warecol=$tpcc_cust_warecol
tpcc_icust2_indexon=cust
tpcc_hist_warecol=h_w_id
tpcc_hist_cols=`cat <<!

```

```

h_c_id-number
h_c_d_id-number
h_c_w_id-number
h_d_id-number
h_w_id-number
h_date-date
h_amount-number
h_data-varchar2(24)
!
.
set +a
#indexnames- with columns for table $1, pull out
the desired indices
#and echo each. Also allowed in format, two
characters after the
#index specifying if type and SORT should be
listed as well-
#example 1t-4t-2tt-3tt gives the type on each, but
only SORT on the
#two entries.
#echos blank if cannot index.
indexnames(){
  tabcols=`tp $1 cols`
  if expr x$2 = xno \|| x$2 = x > /dev/null; then
    echo ""
  else
    colslft=$2
    addcomma=
    while test -n "$colslft"; do
      desiredcol=`echo $colslft | cut -f1 -d.`
      yestype=`x` echo $desiredcol | cut -b2`
      yessort=`x` echo $desiredcol | cut -b3`
      desiredcol=`echo $desiredcol | cut -b1`
      desiredcol=`echo $tabcols | cut -d' ' -
f$desiredcol`
      if expr $yestype = xt > /dev/null; then
        firstcol=`echo $desiredcol | cut -f1 -d.`
        desiredcol=`$firstcol` echo $desiredcol | cut -
f2 -d.`
      else
        desiredcol=`echo $desiredcol | cut -f1 -d.`
      fi
      if expr $yessort = xt > /dev/null; then
        desiredcol=`$desiredcol SORT`
      fi
      echo $addcomma `echo $desiredcol | cut -f1 -
d.`
      addcomma=,
# on some systems, we have "lastcol" instead of
"lastcol-" like we want due to cut's behavior.
# we can make it work on these systems by
making sure the - is in.
      colslft=`echo $colslft | grep -.`
      colslft=`echo $colslft | cut -f2- -d.`
      done
    fi
  }
#colize- get columns desired for table $1, print
with commas between.
#second argument is a string like tttfff,
representing
#which columns are to be displayed.
# $3 true if only name should be printed,
otherwise name and value are printed
# $4 has all columns that need to have SORT
after them (for queue)
# $5 has all the columns that need to have
compress after them (for hash tables)
colize(){

```

```

if expr $2 = all > /dev/null; then
  allcols=t
else
  enabledcols=$2
fi
sortcols=$4
compresscols=$5
tabcols=`tp $1 cols`
addcomma=
for curcol in $tabcols ; do
  if expr `x` echo $sortcols | cut -b1` = xt >
/dev/null; then
    sortsql= SORT
  else
    sortsql=
  fi
  if expr `x` echo $compresscols | cut -b1` = xt >
/dev/null; then
    compresssql=compress
  else
    compresssql=
  fi
  if expr `x` echo $enabledcols | cut -b1` = xt \||
x$allcols = xt > /dev/null; then
    if expr x$3 = xt > /dev/null; then
      echo $addcomma `echo $curcol | cut -f1 -d.`
    else
      name=`echo $curcol | cut -f1 -d.`
      type=`echo $curcol | cut -f2 -d.`
      if expr "x$tpcc_ieee_number" = "xt"
>/dev/null; then
        if expr "$name" = "c_balance" \|| "$name" =
"d_ytd" \|| "$name" = "s_ytd" \|| "$name" = "w_ytd"
> /dev/null ; then
          type="binary_double"
        fi
        if expr "$name" = "c_ytd_payment" \||
"$name" = "c_payment_cnt" \|| "$name" =
"c_delivery_cnt" \|| "$name" = "h_amount" \||
"$name" = "i_price" \|| "$name" = "ol_amount" \||
"$name" = "ol_quantity" \|| "$name" = "s_quantity"
\|| "$name" = "s_order_cnt" \|| "$name" =
"s_remote_cnt" > /dev/null ; then
          type="binary_float"
        fi
      fi
      if expr x$allcols = xt > /dev/null; then
#      echo $addcomma $name $type `echo
$curcol | cut -f3 -d.` | tr . ' ' $sortsql $compresssql
echo $addcomma $name $type `echo
$curcol | cut -f3 -d.` | tr . ' ' $sortsql
      else
#      echo $addcomma $name $type $sortsql
$compresssql
echo $addcomma $name $type $sortsql
      fi
    fi
  fi
  addcomma=,
  enabledcols=`echo $enabledcols | cut -b2-`
  sortcols=`echo $sortcols | cut -b2-`
  compresscols=`echo $compresscols | cut -b2-`
done
}

#createpart- create the sql to generate partitions
based on warehouse.
createpart(){
  warecol=`tp $1 warecol`

```

```

#check if unavailable for partitioning
if test -z "$warecol"; then
  exit 1
fi
if expr $warecol = none > /dev/null; then
  exit 1
fi

echo "partition by range( $warecol )" (

partcount=0
waredivide=0
wareinc=`expr $tpcc_scale / $tpcc_np`
wareextra=`expr $tpcc_scale \% $tpcc_np`
addcomma=
while expr $partcount \< $tpcc_np > /dev/null; do
  waredivide=`expr $wareinc + $waredivide +
\($wareextra / $tpcc_np)`
  partvalue=`expr 1 + $waredivide`
  if test $partcount = `expr $tpcc_np - 1` ; then
    partvalue="maxvalue"
  fi
  echo $addcomma partition ${1}_$partcount
values less than \($partvalue\) tablespace
${1}_$partcount
  addcomma=,
  wareextra=`expr $wareextra + 1`
  partcount=`expr $partcount + 1`
done
echo `)`
}

#createindexpart- create the sql to generate
partitions based on warehouse,
# this time for indices (so no calculation is
required.)
createindexpart(){
  echo `local `
  partcount=0
  addcomma=
  while expr $partcount \< $tpcc_np > /dev/null; do
    echo $addcomma partition ${1}_$partcount
tablespace ${1}_$partcount
    addcomma=,
    partcount=`expr $partcount + 1`
  done
  echo `)`
}

createindexpart_rac_iordr2() {
  echo `global partition by range (o_w_id) `
  partcount=0
  warepernode=`$tpcc_bcexpr $tpcc_scale /
$tpcc_np`
  addcomma=
  warenum=1
  while expr $partcount \< $tpcc_np > /dev/null; do
    warenum=`expr $warenum + $warepernode`
    echo $addcomma partition iordr2_${partcount}
values less than `(` $warenum `)` tablespace
iordr2_${partcount}
    addcomma=,
    partcount=`expr $partcount + 1`
  done
  echo `)`
}

tpcc_ware_tsfileinc=$tpcc_np
tpcc_dist_tsfileinc=$tpcc_np
tpcc_nord_tsfileinc=$tpcc_np
tpcc_iware_tsfileinc=$tpcc_np

```

```

tpcc_idist_tsfileinc=$tpcc_np
tpcc_inord_tsfileinc=$tpcc_np
tpcc_item_tsfileinc=1
tpcc_iitem_tsfileinc=1
if test $tpcc_np -eq 1 ; then
  tpcc_iordr2_indices=4-3-2-1-
fi
:
:
tkvcin.sql
:
:
-- The initnew package for storing variables used
in the
-- New Order anonymous block
CREATE OR REPLACE PACKAGE inittpcc
AS
  TYPE intarray IS TABLE OF INTEGER INDEX
  BY BINARY_INTEGER;
  TYPE distarray IS TABLE OF VARCHAR(24)
  INDEX BY BINARY_INTEGER;
  nulldate DATE;
  TYPE rowidarray IS TABLE OF ROWID INDEX
  BY PLS_INTEGER;
  s_dist          distarray;
  idx1arr         intarray;
  s_remote        intarray;
  dist            intarray;
  row_id          rowidarray;
  cust_rowid      rowid;
  dist_name       VARCHAR2(11);
  ware_name       VARCHAR2(11);
  c_num          PLS_INTEGER;
  PROCEDURE init_no(idxarr intarray);
  PROCEDURE init_del;
  PROCEDURE init_pay;
END 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
:
:
tokilobytes.sh
:
:
#!/bin/sh
# convert k, m, g, t into kilobytes, echo result
# no units means kilobytes is assumed.
amount=`echo $1 | sed -e's/^.*(.)\|1/'

```

```

unit=`echo $1 | sed -e's/^.*(.)\|1/'
if expr $unit = k \| $unit = K> /dev/null; then
  result=$amount
elif expr $unit = m \| $unit = M> /dev/null; then
  result=`$tpcc_bcexpr $amount \` 1024`
elif expr $unit = g \| $unit = G> /dev/null; then
  result=`$tpcc_bcexpr $amount \` 1024 \` 1024`
elif expr $unit = t \| $unit = T> /dev/null; then
  result=`$tpcc_bcexpr $amount \` 1024 \` 1024 \`
1024`
else
  #assume it's already correct.
  result=$result
fi
if test -n "$result"; then
  echo $result
else
  #check if it's still valid
  echo $1
  expr $1 + 0 > /dev/null
  exit $?
fi
:
:
tpccload.c
:
:
#ifdef RCSID
static char *RCSid =
  "$Header: tpccload.c 7030100.1 96/05/13
16:20:36 plai Generic<base> $ Copyr (c) 1993
Oracle";
#endif /* RCSID */

/*=====
=====+
| Copyright (c) 1994 Oracle Corp, Redwood
Shores, CA |
| OPEN SYSTEMS PERFORMANCE
GROUP |
| All Rights Reserved
|
+=====
=====+
| FILENAME
| tpccload.c
| DESCRIPTION
| Load or generate TPC-C database tables.
| Usage: tpccload -M <# of wares> [options]
| options: -A load all tables
|           -w load ware table
|           -d load dist table
|           -c load cust table
|           -i load item table
|           -s load stok table (cluster around
s_w_id)
|           -S load stok table (cluster around
s_i_id)
|           -h load hist table
|           -n load new-order table
|           -o <oline file> load order and
order-line table
|           -b <ware#> beginning ware
number
|           -e <ware#> ending ware number
|           -j <item#> beginning item number
(with -S)
|           -k <item#> ending item number
(with -S)

```

```

| -g generate rows to standard
output

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

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys/types.h>
#include "tpcc.h"

#ifdef ORA_NT
#undef boolean
#include <process.h>
#include "dpcore.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 SQLXTW "INSERT INTO ware (w_id,
w_ytd, w_tax, w_name, w_street_1, w_street_2,
w_city, w_state, w_zip) VALUES (:w_id,
30000000, :w_tax, :w_name, :w_street_1, \
:w_street_2, :w_city, :w_state, :w_zip)"
#define SQLXTD "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, 30000000, :d_tax, \

3001, :d_name, :d_street_1, :d_street_2, :d_city, :
d_state, :d_zip)"

#define SQLXTCQUERY "select 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 SQLTTC "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 SQLTTH "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 SQLTTSQUERY "select count(*) from
stok where s_w_id = :s_s_w_id and s_i_id
= :s_s_i_id"
#define SQLTTS "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 SQLTIT "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 SQLTTO1 "INSERT INTO ordr (O_ID,
O_D_ID, O_W_ID, O_C_ID, O_ENTRY_D, O_CAR
RIER_ID, O_OL_CNT, O_ALL_LOCAL) \
VALUES (:o_id, :o_d_id, :o_w_id, :o_c_id, \
SYSDATE, :o_carrier_id, :o_ol_cnt, 1)"
#define SQLTTO2 "INSERT INTO ordr (O_ID,
O_D_ID, O_W_ID, O_C_ID, O_ENTRY_D, O_CAR
RIER_ID, O_OL_CNT, O_ALL_LOCAL) \
VALUES (:o_id, :o_d_id, :o_w_id, :o_c_id, \
SYSDATE, 11, :o_ol_cnt, 1)"
#define SQLTTO11 "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 SQLTTO12 "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 SQLTTO13 "INSERT INTO nord
(no_o_id, no_d_id, no_w_id) VALUES
(no_o_id, no_d_id, no_w_id)"
#define SQLTTO14 "alter session set
'_enable_hash_overflow'=true"
#define SQLTTO15 "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 *tpcsusr;
OCISmt *curw;
OCISmt *curd;
OCISmt *curc;
OCISmt *curcs;
OCISmt *curh;
OCISmt *curi;
OCISmt *curss;
OCISmt *curj;
OCISmt *curk;
OCISmt *curo1;
OCISmt *curo2;
OCISmt *curo11;
OCISmt *curo12;
OCISmt *cumo;
OCISmt *w_id_bp = (OCISmt *) 0;
OCISmt *w_name_bp = (OCISmt *) 0;
OCISmt *w_street1_bp = (OCISmt *) 0;
OCISmt *w_street2_bp = (OCISmt *) 0;
OCISmt *w_city_bp = (OCISmt *) 0;
OCISmt *w_state_bp = (OCISmt *) 0;
OCISmt *w_zip_bp = (OCISmt *) 0;
OCISmt *w_tax_bp = (OCISmt *) 0;

```

```

OCISmt *d_id_bp = (OCISmt *) 0;
OCISmt *d_w_id_bp = (OCISmt *) 0;
OCISmt *d_name_bp = (OCISmt *) 0;
OCISmt *d_street1_bp = (OCISmt *) 0;
OCISmt *d_street2_bp = (OCISmt *) 0;
OCISmt *d_city_bp = (OCISmt *) 0;
OCISmt *d_state_bp = (OCISmt *) 0;
OCISmt *d_zip_bp = (OCISmt *) 0;
OCISmt *d_tax_bp = (OCISmt *) 0;
OCISmt *s_c_ret_bp = (OCISmt *) 0;
OCISmt *s_c_id_bp = (OCISmt *) 0;
OCISmt *s_c_d_id_bp = (OCISmt *) 0;
OCISmt *s_c_w_id_bp = (OCISmt *) 0;
OCISmt *c_id_bp = (OCISmt *) 0;
OCISmt *c_d_id_bp = (OCISmt *) 0;
OCISmt *c_w_id_bp = (OCISmt *) 0;
OCISmt *c_first_bp = (OCISmt *) 0;
OCISmt *c_last_bp = (OCISmt *) 0;
OCISmt *c_street1_bp = (OCISmt *) 0;
OCISmt *c_street2_bp = (OCISmt *) 0;
OCISmt *c_city_bp = (OCISmt *) 0;
OCISmt *c_state_bp = (OCISmt *) 0;
OCISmt *c_zip_bp = (OCISmt *) 0;
OCISmt *c_phone_bp = (OCISmt *) 0;
OCISmt *c_discount_bp = (OCISmt *) 0;
OCISmt *c_credit_bp = (OCISmt *) 0;
OCISmt *c_data_bp = (OCISmt *) 0;
OCISmt *i_id_bp = (OCISmt *) 0;
OCISmt *i_im_id_bp = (OCISmt *) 0;
OCISmt *i_name_bp = (OCISmt *) 0;
OCISmt *i_price_bp = (OCISmt *) 0;
OCISmt *i_data_bp = (OCISmt *) 0;
OCISmt *s_s_ret_bp = (OCISmt *) 0;
OCISmt *s_s_i_id_bp = (OCISmt *) 0;
OCISmt *s_s_w_id_bp = (OCISmt *) 0;
OCISmt *s_i_id_bp = (OCISmt *) 0;
OCISmt *s_w_id_bp = (OCISmt *) 0;
OCISmt *s_quantity_bp = (OCISmt *) 0;
OCISmt *s_dist_01_bp = (OCISmt *) 0;
OCISmt *s_dist_02_bp = (OCISmt *) 0;
OCISmt *s_dist_03_bp = (OCISmt *) 0;
OCISmt *s_dist_04_bp = (OCISmt *) 0;
OCISmt *s_dist_05_bp = (OCISmt *) 0;
OCISmt *s_dist_06_bp = (OCISmt *) 0;
OCISmt *s_dist_07_bp = (OCISmt *) 0;
OCISmt *s_dist_08_bp = (OCISmt *) 0;
OCISmt *s_dist_09_bp = (OCISmt *) 0;
OCISmt *s_dist_10_bp = (OCISmt *) 0;
OCISmt *s_data_bp = (OCISmt *) 0;
OCISmt *h_c_id_bp = (OCISmt *) 0;
OCISmt *h_c_d_id_bp = (OCISmt *) 0;
OCISmt *h_c_w_id_bp = (OCISmt *) 0;
OCISmt *h_d_id_bp = (OCISmt *) 0;
OCISmt *h_w_id_bp = (OCISmt *) 0;
OCISmt *h_data_bp = (OCISmt *) 0;
OCISmt *ol_o_id_bp = (OCISmt *) 0;
OCISmt *ol_d_id_bp = (OCISmt *) 0;
OCISmt *ol_w_id_bp = (OCISmt *) 0;
OCISmt *ol_i_id_bp = (OCISmt *) 0;
OCISmt *ol_number_bp = (OCISmt *) 0;
OCISmt *ol_supply_w_id_bp = (OCISmt *) 0;
OCISmt *ol_dist_info_bp = (OCISmt *) 0;
OCISmt *ol_amount_bp = (OCISmt *) 0;
OCISmt *o_id_bp = (OCISmt *) 0;
OCISmt *o_d_id_bp = (OCISmt *) 0;
OCISmt *o_w_id_bp = (OCISmt *) 0;
OCISmt *o_c_id_bp = (OCISmt *) 0;
OCISmt *o_carrier_id_bp = (OCISmt *) 0;
OCISmt *o_ol_cnt_bp = (OCISmt *) 0;

```

```

OCIBind *o_ocnt_bp = (OCIBind *) 0;
OCIBind *o_oclnt_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:\t\ttpccload -M <multiplier>
[options]\n");
    fprintf(stderr, "options:\n");
    fprintf(stderr, "\t-A :tload all tables\n");
    fprintf(stderr, "\t-w :tload ware table\n");
    fprintf(stderr, "\t-d :tload dist table\n");
    fprintf(stderr, "\t-c :tload cust table\n");
    fprintf(stderr, "\t-i :tload item table\n");
    fprintf(stderr, "\t-s :tload stok table (cluster
around s_w_id)\n");
    fprintf(stderr, "\t-S :tload stok table (cluster
around s_i_id)\n");
    fprintf(stderr, "\t-h :tload hist table\n");
    fprintf(stderr, "\t-n :tload new-order table\n");
    fprintf(stderr, "\t-o <oline file> :tload order and
order-line table\n");
    fprintf(stderr, "\t-b <ware#> :tbeginning ware
number\n");
    fprintf(stderr, "\t-e <ware#> :tending ware
number\n");
    fprintf(stderr, "\t-j <item#> :tbeginning item
number (with -S)\n");
    fprintf(stderr, "\t-k <item#> :tending item
number (with -S)\n");
    fprintf(stderr, "\t-g :tgenerate rows to standard
output\n");
    fprintf(stderr, "\t $tpcc_bench must be set to the
location of the kit\n");
    fprintf(stderr, "\n");
    exit(1);
}
int sqlfile(fnam, linebuf)
char *fnam;
text *linebuf;
{
    FILE *fd;
    int nulpt = 0;
    char realfile[512];
    sprintf(realfile, "%s", fnam);
    fd = fopen(realfile, "r");
    if (!fd)
    {
        return (0);
    }
    while (fgets((char *)linebuf+nulpt,
SQL_BUF_SIZE, fd))
    {
        nulpt = strlen((char *)linebuf);
    }
    return(nulpt);
}
void quit()
{
    OCIERROR(errhp, OCI_SessionEnd
(tpcsvc, errhp, tpcusr, OCI_DEFAULT));
    OCIERROR(errhp, OCI_ServerDetach ( tpcsrv,
errhp, OCI_DEFAULT));
    OCIHandleFree((dvoid *)tpcusr,
OCI_HTYPE_SESSION);
    OCIHandleFree((dvoid *)tpcsvc,
OCI_HTYPE_SVCCTX);

```

```

    OCIHandleFree((dvoid *)errhp,
OCI_HTYPE_ERROR);
    OCIHandleFree((dvoid *)tpcusr,
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_lm_id[100];
    int i_price[100];
    char i_name[100][25];
    char i_data[100][51];
    int s_s_count;
    int s_s_i_id;
    int s_s_w_id;

```

```

int s_i_id[100];
int s_w_id[100];
int s_quantity[100];
char s_dist_01[100][24];
char s_dist_02[100][24];
char s_dist_03[100][24];
char s_dist_04[100][24];
char s_dist_05[100][24];
char s_dist_06[100][24];
char s_dist_07[100][24];
char s_dist_08[100][24];
char s_dist_09[100][24];
char s_dist_10[100][24];
char s_data[100][51];
int h_w_id[100];
int h_d_id[100];
int h_c_id[100];
char h_data[100][25];
int o_id[100];
int o_d_id[100];
int o_w_id[100];
int o_c_id[100];
int o_carrier_id[100];
int o_ol_cnt[100];
int ol_o_id[1500];
int ol_d_id[1500];
int ol_w_id[1500];
int ol_number[1500];
int ol_i_id[1500];
int ol_supply_w_id[1500];
int ol_amount[1500];
char ol_dist_info[1500][24];
int o_cnt;
int ol_cnt;
ub2 ol_o_id_len[1500];
ub2 ol_d_id_len[1500];
ub2 ol_w_id_len[1500];
ub2 ol_number_len[1500];
ub2 ol_i_id_len[1500];
ub2 ol_supply_w_id_len[1500];
ub2 ol_dist_info_len[1500];
ub2 ol_amount_len[1500];
ub4 ol_o_id_clen;
ub4 ol_d_id_clen;
ub4 ol_w_id_clen;
ub4 ol_number_clen;
ub4 ol_i_id_clen;
ub4 ol_supply_w_id_clen;
ub4 ol_dist_info_clen;
ub4 ol_amount_clen;
ub2 o_id_len[100];
ub2 o_d_id_len[100];
ub2 o_w_id_len[100];
ub2 o_c_id_len[100];
ub2 o_carrier_id_len[100];
ub2 o_ol_cnt_len[100];
ub4 o_id_clen;
ub4 o_d_id_clen;
ub4 o_w_id_clen;
ub4 o_c_id_clen;
ub4 o_carrier_id_clen;
ub4 o_ol_cnt_clen;
text stmbuf[16*1024];
int no_o_id[100];
int no_d_id[100];
int no_w_id[100];
char sdate[30];
#ifdef ORA_NT
clock_t begin_time, end_time;

```

```

clock_t begin_cpu, end_cpu;
char *arg_ptr, **end_args;
#else
double begin_time, end_time;
double begin_cpu, end_cpu;
double gettime(), getcpu();
extern int getopt();
extern char *optarg;
extern int optind, opterr;
int opt;
#endif
char *argstr="M:AwdcisShno:b:ej:k:g";
int do_A=0;
int do_w=0;
int do_d=0;
int do_i=0;
int do_c=0;
int do_s=0;
int do_S=0;
int do_h=0;
int do_o=0;
int do_n=0;
int gen=0;
int bware=1;
int eware=0;
int bitem=1;
int eitem=0;
FILE *olfp=NULL;
char olfname[100];
char *basename;
int status;
#ifdef ORA_NT
char fname[100];
FILE *logfile;
#endif /* ORA_NT */
/*-----+
| Parse command line -- look for scale factor.
|
+-----*/
if (argc == 1) {
    myusage ();
}
#ifdef ORA_NT
end_args = argv + argc;
for (++argv; argv < end_args; )
{
    arg_ptr = *argv++;
    if (*arg_ptr != '-')
    {
        myusage ();
    } else
    {
        switch (arg_ptr[1]) {
        case '?': myusage ();
            break;
        case 'M': scale = atoi (*argv++);
            break;
        case 'A': do_A = 1;
            break;
        case 'w': do_w = 1;
            break;
        case 'd': do_d = 1;
            break;
        case 'c': do_c = 1;
            break;
        case 'i': do_i = 1;
            break;
        case 's': do_s = 1;
            break;
        case 'S': do_S = 1;
            break;
        case 'h': do_h = 1;
            break;
        case 'n': do_n = 1;
            break;
        case 'o': do_o = 1;
            strcpy (olfname, 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 'S': do_S = 1;
    break;
case 'h': do_h = 1;
    break;
case 'n': do_n = 1;
    break;
case 'o': do_o = 1;
    strcpy (olfname, *argv++);
    break;
case 'b': bware = atoi (*argv++);
    break;
case 'e': eware = atoi (*argv++);
    break;
case 'j': bitem = atoi (*argv++);
    break;
case 'k': eitem = atoi (*argv++);
    break;
case 'g': gen = 1;
    strcpy (fname, *argv++);
    break;
case 'l': logfile=fopen(*argv+,"w");
    break;
default: fprintf (stderr, "THIS SHOULD
NEVER HAPPEN!!!\n");
    fprintf (stderr, "(reached default case in
getopt ())\n");
    myusage ();
}
}
}

#else
while ((opt = getopt (argc, argv, argstr)) != -1) {
    switch (opt) {
    case '?': myusage ();
        break;
    case 'M': scale = atoi (optarg);
        break;
    case 'A': do_A = 1;
        break;
    case 'w': do_w = 1;
        break;
    case 'd': do_d = 1;
        break;
    case 'c': do_c = 1;
        break;
    case 'i': do_i = 1;
        break;
    case 's': do_s = 1;
        break;
    case 'S': do_S = 1;
        break;
    case 'h': do_h = 1;
        break;
    case 'n': do_n = 1;
        break;
    case 'o': do_o = 1;
        strcpy (olfname, optarg);
        break;
    case 'b': bware = atoi (optarg);
        break;
    case 'e': eware = atoi (optarg);
        break;
    case 'j': bitem = atoi (optarg);
        break;
    case 'k': eitem = atoi (optarg);
        break;
    }
}

```

```

case 'g': gen = 1;
    break;
default: fprintf (stderr, "THIS SHOULD
NEVER HAPPEN!!!\n");
    fprintf (stderr, "(reached default case in
getopt ())\n");
    myusage ();
}
}
}
#endif /* ORA_NT */
/*-----*]
| Rudimentary error checking
|
+-----*/
if (scale < 1) {
    fprintf (stderr, "Invalid scale factor: '%d'\n",
scale);
    myusage ();
}
if (!(do_A || do_w || do_d || do_c || do_i || do_s ||
do_S || do_h || do_o ||
do_n)) {
    fprintf (stderr, "What should I load???\n");
    myusage ();
}
if (gen && (do_A || (do_w + do_d + do_c + do_i
+ do_s + do_S + do_h + do_o +
do_n > 1))) {
    fprintf (stderr, "Can only generate table one at
a time\n");
    myusage ();
}
if (do_S && (do_A || do_s)) {
    fprintf (stderr, "Cluster stock table around
s_w_id or s_i_id?\n");
    myusage ();
}
if (eware <= 0)
    eware = scale;
if (eitem <= 0)
    eitem = STOCFAC;
if (do_S) {
    if ((bitem < 1) || (bitem > STOCFAC)) {
        fprintf (stderr, "Invalid beginning item
number: '%d'\n", bitem);
        myusage ();
    }
    if ((eitem < bitem) || (eitem > STOCFAC)) {
        fprintf (stderr, "Invalid ending item number:
'%d'\n", eitem);
        myusage ();
    }
}
if (do_o) {
    if ((basename = getenv ("tpcc_bench")) ==
NULL)
    {
        fprintf (stderr, "$tpcc_bench is not set");
        myusage ();
    }
}
if ((bware < 1) || (bware > scale)) {
    fprintf (stderr, "Invalid beginning warehouse
number: '%d'\n", bware);
    myusage ();
}
if ((eware < bware) || (eware > scale)) {
    fprintf (stderr, "Invalid ending warehouse
number: '%d'\n", eware);
}

```

```

mysusage ();
}
if (gen && do_o) {
    if ((olfp = fopen(olfname, "w")) == NULL) {
        fprintf(stderr, "Can't open '%s' for writing
order lines\n", olfname);
        mysusage ();
    }
}

/*-----+
| 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(tpcsrv,
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,OCISmtPrepare(curw,
errhp, (text *)SQLXTW,
            strlen((char *)SQLXTW), (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,OCISmtPrepare(curd,
errhp, (text *)SQLXTD,
            strlen((char *)SQLXTD), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

    if (do_A || do_c) {

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

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid
***)&curcs), OCI_HTYPE_STMT, 0, (dvoid***)0);
    OCIERROR(errhp,OCISmtPrepare(curcs,
errhp, (text *)SQLXTCQUERY,
            strlen((char *)SQLXTCQUERY),
(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,OCISmtPrepare(curh,
errhp, (text *)SQLXTH,
            strlen((char *)SQLXTH), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }
    if (do_A || do_s || do_s) {

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

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid
***)&curss), OCI_HTYPE_STMT, 0, (dvoid***)0);
    OCIERROR(errhp,OCISmtPrepare(curss,
errhp, (text *)SQLXTSQQUERY,
            strlen((char *)SQLXTSQQUERY),
(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,OCISmtPrepare(curi,
errhp, (text *)SQLXTI,
            strlen((char *)SQLXTI), (ub4)
OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }
    if (do_A || do_o) {
        int stat;
        char fname[160];

OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid
***)&curot), 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,OCISmtPrepare(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,OCISmtPrepare(curno,
errhp, (text *)SQLXTNO,
            strlen((char *)SQLXTNO), (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) {
OCIERROR(errhp, OCIBindByName(curcs,
&s_c_id_bp, errhp, (text *)":s_c_id",
strlen(":s_c_id"), (ub1 *)&s_c_id,
sizeof(int), SQLT_INT,
(dvoid *) 0, (ub2 *)0, (ub2 *)0,
(ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
OCIERROR(errhp, OCIBindByName(curcs,
&s_c_w_id_bp, errhp, (text *)":s_c_w_id",
strlen(":s_c_w_id"), (ub1
*)&s_c_w_id, sizeof(int), SQLT_INT,
(dvoid *) 0, (ub2 *)0, (ub2 *)0,
(ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
OCIERROR(errhp, OCIBindByName(curcs,
&s_c_d_id_bp, errhp, (text *)":s_c_d_id",
strlen(":s_c_d_id"), (ub1 *)&s_c_d_id,
sizeof(int), SQLT_INT,
(dvoid *) 0, (ub2 *)0, (ub2 *)0,
(ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
OCIERROR(errhp, OCIBindByName(curcs,
&s_c_ret_bp, errhp, 1, &s_c
_count, sizeof(int), SQLT_INT, \
0, 0, 0, OCI_DEFAULT));
OCIERROR(errhp, OCIBindByName(curd,
&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(curd,
&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(curd,
&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(curd,
&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),
SQL_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), SOLT_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,
SQL_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), SOLT_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,
SQL_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), SOLT_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), SOLT_INT,
        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
        (ub4) 0, (ub4 *) 0, (ub4)
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), SOLT_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), SOLT_INT,
        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), SOLT_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), SOLT_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), SOLT_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,
24, SOLT_CHR,
        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
        (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
        OCIERROR(errhp, OCIBindByName(curs,
&s_dist_02_bp, errhp, (text *)"s_dist_02",
        strlen(":s_dist_02"), (ub1 *)s_dist_02,
24, SOLT_CHR,
        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
        (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
        OCIERROR(errhp, OCIBindByName(curs,
&s_dist_03_bp, errhp, (text *)"s_dist_03",
        strlen(":s_dist_03"), (ub1 *)s_dist_03,
24, SOLT_CHR,
        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
        (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
        OCIERROR(errhp, OCIBindByName(curs,
&s_dist_04_bp, errhp, (text *)"s_dist_04",
        strlen(":s_dist_04"), (ub1 *)s_dist_04,
24, SOLT_CHR,
        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
        (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
        OCIERROR(errhp, OCIBindByName(curs,
&s_dist_05_bp, errhp, (text *)"s_dist_05",
        strlen(":s_dist_05"), (ub1 *)s_dist_05,
24, SOLT_CHR,
        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
        (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
        OCIERROR(errhp, OCIBindByName(curs,
&s_dist_06_bp, errhp, (text *)"s_dist_06",
        strlen(":s_dist_06"), (ub1 *)s_dist_06,
24, SOLT_CHR,
        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
        (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
        OCIERROR(errhp, OCIBindByName(curs,
&s_dist_07_bp, errhp, (text *)"s_dist_07",
        strlen(":s_dist_07"), (ub1 *)s_dist_07,
24, SOLT_CHR,
        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
        (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
        OCIERROR(errhp, OCIBindByName(curs,
&s_dist_08_bp, errhp, (text *)"s_dist_08",
        strlen(":s_dist_08"), (ub1 *)s_dist_08,
24, SOLT_CHR,
        (dvoid *) 0, (ub2 *)0, (ub2 *)0,

```

```

        (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
        OCIERROR(errhp, OCIBindByName(curs,
&s_dist_09_bp, errhp, (text *)"s_dist_09",
        strlen(":s_dist_09"), (ub1 *)s_dist_09,
24, SOLT_CHR,
        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
        (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
        OCIERROR(errhp, OCIBindByName(curs,
&s_dist_10_bp, errhp, (text *)"s_dist_10",
        strlen(":s_dist_10"), (ub1 *)s_dist_10,
24, SOLT_CHR,
        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
        (ub4) 0, (ub4 *) 0, (ub4)
OCI_DEFAULT));
        OCIERROR(errhp, OCIBindByName(curs,
&s_data_bp, errhp, (text *)"s_data",
        strlen(":s_data"), (ub1 *)s_data, 51,
SQL_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), SOLT_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), SOLT_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), SOLT_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), SOLT_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), SOLT_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,
SQL_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,
    &o_ol_id_bp, errhp, (text *)"o_ol_id",
    strlen("o_ol_id"), (ub1 *)o_ol_id,
    sizeof(int), SQLT_INT,
    (dvoid *) 0, (ub2 *)o_ol_id_len, (ub2
    *)0,
    (ub4) 15*ORDEARR, (ub4
    *)&o_ol_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) 15*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) 15*ORDEARR, (ub4 *)
    &o_w_id_clen, (ub4) OCI_DEFAULT));
  OCIERROR(errhp, OCIBindByName(curo1,
    &o_l_number_bp, errhp, (text *)"o_l_number",
    strlen("o_l_number"), (ub1
    *)o_l_number, sizeof(int), SQLT_INT,
    (dvoid *) 0, (ub2 *)o_l_number_len,
    (ub2 *)0,
    (ub4) 15*ORDEARR, (ub4 *)
    &o_l_number_clen, (ub4) OCI_DEFAULT));
  OCIERROR(errhp, OCIBindByName(curo1,
    &o_i_id_bp, errhp, (text *)"o_i_id",
    strlen("o_i_id"), (ub1 *)o_i_id,
    sizeof(int), SQLT_INT,
    (dvoid *) 0, (ub2 *)o_i_id_len, (ub2
    *)0,
    (ub4) 15*ORDEARR, (ub4 *)
    &o_i_id_clen, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,
    &o_l_supply_w_id_bp, errhp, (text
    *)"o_l_supply_w_id",
    strlen("o_l_supply_w_id"), (ub1
    *)o_l_supply_w_id, sizeof(int), SQLT_INT,
    (dvoid *) 0, (ub2
    *)o_l_supply_w_id_len, (ub2 *)0,
    (ub4) 15*ORDEARR, (ub4 *)
    &o_l_supply_w_id_clen, (ub4) OCI_DEFAULT));
  OCIERROR(errhp, OCIBindByName(curo1,
    &o_l_dist_info_bp, errhp, (text *)"o_l_dist_info",
    strlen("o_l_dist_info"), (ub1
    *)o_l_dist_info, 24, SQLT_CHR,
    (dvoid *) 0, (ub2 *)o_l_dist_info_len,
    (ub2 *)0,
    (ub4) 15*ORDEARR, (ub4 *)
    &o_l_dist_info_clen, (ub4) OCI_DEFAULT));

```

```

OCIERROR(errhp, OCIBindByName(curo1,
    &o_l_amount_bp, errhp, (text *)"o_l_amount",
    strlen("o_l_amount"), (ub1
    *)o_l_amount, sizeof(int), SQLT_INT,
    (dvoid *) 0, (ub2 *)o_l_amount_len,
    (ub2 *)0,
    (ub4) 15*ORDEARR, (ub4 *)
    &o_l_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);
#ifdef ORANT
  srand48 (SEED);
#endif
  initperm ();
  /*-----+
  | Load the WAREHOUSE table.
  |
  +-----*/
  if (do_A || do_w) {
    nrows = aware - bware + 1;
    fprintf (stderr, "Loading/generating warehouse:
    w%d - w%d (%d rows)\n",
    bware, aware, nrows);
    begin_time = gettime ();
    begin_cpu = getcpu ();
    for (loop = bware; loop <= aware; loop++) {
      w_tax = (float) ((rand48 () % 2001) *
    0.0001);
      randstr (w_name, 6, 10);
      randstr (w_street_1, 10, 20);
      randstr (w_street_2, 10, 20);
      randstr (w_city, 10, 20);
      randstr (str2, 2, 2);
      randnum (num9, 9);
      num9[4] = num9[5] = num9[6] = num9[7] =
    num9[8] = '1';
      if (gen) {
        printf ("%d
    30000000 %6.4f %s %s %s %s %s %s\n", loop,
    w_tax,
    w_name, w_street_1, w_street_2,
    w_city, str2, num9);
        fflush (stdout);
      }
      else {

```

```

w_id = loop;
strncpy (w_state, str2, 2);
strncpy (w_zip, num9, 9);
status = OCISmtExecute(tpscvc, curw,
errhp, (ub4) 1, (ub4) 0,
(CONST OCI_Snapshot*) 0,
(OCI_Snapshot*) 0,
(ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
if (status != OCI_SUCCESS) {
fprintf (stderr, "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) ((rand48 () % 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 = OCISmtExecute(tpscvc, curd,
errhp, (ub4) DISTARR, (ub4) 0,
(CONST OCI_Snapshot*) 0,
(OCI_Snapshot*) 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);
OCISmtPrepare(cur, errhp, stmbuf,
strlen((char *)stmbuf),
OCI_NT_V_SYNTAX,
OCI_DEFAULT);
OCIERROR(errhp, OCISmtExecute(tpscvc,
cur, errhp, 1, 0, 0, OCI_DEFAULT));
OCIHandleFree(cur, 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 = OCISmtExecute(tpscvc, curcs,
errhp, (ub4) 1, (ub4) 0,
(CONST OCI_Snapshot*) 0,
(OCI_Snapshot*) 0,
(ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
if (status != OCI_SUCCESS) {
OCIERROR(errhp, status);
}
}
}
quit ();
exit (1);
}
}
}

```

```

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 = OCISmtExecute(tpscvc, curcs,
errhp, (ub4) 1, (ub4) 0,
(CONST OCI_Snapshot*) 0,
(OCI_Snapshot*) 0,
(ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
if (status != OCI_SUCCESS) {
fprintf (stderr, "Select failed\n");
OCIERROR(errhp, status);
quit ();
exit (1);
}
}
if (s_c_count == 0) {
s_c_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 = OCISmtExecute(tpscvc, curcs,
errhp, (ub4) 1, (ub4) 0,
(CONST OCI_Snapshot*) 0,
(OCI_Snapshot*) 0,
(ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
if (status != OCI_SUCCESS) {
OCIERROR(errhp, status);
quit ();
exit (1);
}
}
if (s_c_count == 0) break;
else s_c_id++;
}
}
}
if (s_c_id > CUSTFAC) {
s_c_d_id=1;
s_c_w_id++;
s_c_id=1;
}
fprintf (stderr, "start at wid: %d, did: %d,
cid: %d\n", s_c_w_id, s_c_d_id, s_c_id);
cid = s_c_id - 1;
cdid = s_c_d_id;
cwid = s_c_w_id;
nrows = (eware - s_c_w_id + 1) * DISTFAC *
CUSTFAC - (s_c_d_id - 1) * CUSTFAC - s_c_id +
1;
loopcount = 0;
for (row = 0; row < nrows; ) {
for (i = 0; i < CUSTARR; i++, row++) {
cid++;
if (cid > CUSTFAC) { /* cycle cust id */

```

```

cid = 1;          /* cheap mod */
cdid++;         /* shift dist cycle */
if (cdid > DISTFAC) {
    cdid = 1;
    cwdid++;    /* shift ware cycle */
}
}
c_id[i] = cid;
c_d_id[i] = cdid;
c_w_id[i] = cwdid;
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, cwdid, 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 = OCISmtExecute(tpcsvc, curc,
errhp, (ub4) CUSTARR, (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);
    OCISmtPrepare(curi, errhp, stmbuf,
strlen((char *)stmbuf),
        OCI_NTV_SYNTAX,
OCI_DEFAULT);
    OCIERROR(errhp, OCISmtExecute(tpcsvc,
curi, errhp, 1, 0, 0, OCI_DEFAULT));
    OCIHandleFree(curi, OCI_HTYPE_STMT);
}
}
/*-----+
| Load the ITEM table.
+-----*/
if (do_A || do_i) {
    nrows = ITEMFAC;
    fprintf (stderr, "Loading/generating item: (%d
rows)\n ", nrows);
    begin_time = gettime ();
    begin_cpu = getcpu ();
    loopcount = 0;
    for (row = 0; row < nrows; ) {
        for (i = 0; i < ITEMARR; i++, row++) {
            i_im_id[i] = (lrand48 () % 10000) + 1;
            i_price[i] = ((lrand48 () % 9901) + 100);
            randstr (i_name[i], 14, 24);
            randdatastr (i_data[i], 26, 50);
            if (gen) {
                printf ("%d %d %s %d %s\n", row + 1,
i_im_id[i], i_name[i],
                    i_price[i], i_data[i]);
            }
            else {
                i_id[i] = row + 1;
            }
        }
        if (gen) {
            fflush (stdout);
        }
        else {
            status = OCISmtExecute(tpcsvc, curi,
errhp, (ub4) ITEMARR, (ub4) 0,
(CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
(ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
if (status != OCI_SUCCESS) {
            fprintf (stderr, "Aborted at i_id %d\n",
i_id[0]);
            OCIERROR(errhp, status);
            quit ();
            exit (1);
        }
    }
}

```

```

if ((++loopcount) % 50)
    fprintf (stderr, ".");
else
    fprintf (stderr, "%d rows committed\n ",
row);
}
end_time = gettime ();
end_cpu = getcpu ();
fprintf (stderr, "Done. %d rows
loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
        nrows, end_time - begin_time, end_cpu -
begin_cpu);
}
}
/*-----+
| Load the STOCK table.
+-----*/
if (do_A || do_s) {
    nrows = (eware - bware + 1) * STOCFAC;
    fprintf (stderr, "Loading/generating stock: w%d
-w%d (%d rows)\n ",
        bware, aware, nrows);
    begin_time = gettime ();
    begin_cpu = getcpu ();

    s_s_i_id = 1;
    s_s_w_id = bware;

    while (s_s_w_id <= aware) {
        status = OCISmtExecute(tpcsvc, curss,
errhp, (ub4) 1, (ub4) 0,
(CONST OCISnapshot*) 0,
(OCISnapshot*) 0,
(ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
        if (status != OCI_SUCCESS) {
            OCIERROR(errhp, status);
            quit ();
            exit (1);
        }
        if (s_s_count == 0) {
            s_s_w_id--;
            break;
        }
        else s_s_w_id++;
    }
    if (s_s_w_id < bware) s_s_w_id = bware;
    else {
        if (s_s_w_id > aware) s_s_w_id = aware;
        while (s_s_i_id <= STOCFAC) {
            status = OCISmtExecute(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++;
    }
}

```



```

        fflush (stdout);
    }
    else {
        status = OCISmtExecute(tpcsvc, curs,
errhp, (ub4) STOCARR, (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, aware, nrows);
    begin_time = gettime ();
    begin_cpu = getcpu ();
    cid = 0;
    cdid = 1;
    cwid = bware;
    loopcount = 0;
    for (row = 0; row < nrows; ) {
        for (i = 0; i < 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 = OCISmtExecute(tpcsvc, curh,
errhp, (ub4) HISTARR, (ub4) 0,
        (CONST OCISnapshot*) 0,
        (OCISnapshot*) 0,
        (ub4) OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);
        if (status != OCI_SUCCESS) {
            fprintf (stderr, "Aborted at w_id %d,
d_id %d, c_id %d\n",
                h_w_id[0], h_d_id[0], h_c_id[0]);
            OCIERROR(errhp, status);
            quit ();
            exit (1);
        }
    }
}
if ((++loopcount) % 50)
    fprintf (stderr, ".");
else
    fprintf (stderr, " %d rows committed\n ",
row);
}
end_time = gettime ();
end_cpu = getcpu ();
fprintf (stderr, "Done. %d rows
loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
        nrows, end_time - begin_time, end_cpu -
begin_cpu);
}
/*-----+
| Load the ORDERS and ORDER-LINE table.
|-----*/
+-----*/

if (do_A || do_o) {
    int batch_olcnt;
    nrows = (eware - bware + 1) * ORDEFAC *
DISTFAC;
    fprintf (stderr, "Loading/generating orders and
order-line: w%d - w%d (%d ord, ~%d ord)\n ",
        bware, aware, nrows, nrows * 10);
    begin_time = gettime ();
    begin_cpu = getcpu ();
    cid = 0;
    cdid = 1;
    cwid = bware;
    loopcount = 0;
    for (row = 0; row < nrows; ) {
        batch_olcnt = 0;
        for (i = 0; i < ORDEARR; i++, row++) {
            cid++;
            if (cid > ORDEFAC) { /* cycle cust id */
                cid = 1; /* cheap mod */
                cdid++; /* shift district cycle */
                if (cdid > DISTFAC) {
                    cdid = 1;
                    cwid++; /* shift warehouse
cycle */
                }
            }
            o_carrier_id[i] = irand48 () % 10 + 1;
            o_ol_cnt[i] = olcnt = irand48 () % 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 = irand48 () %
100000 + 1;
        if (cid < 2101)
            ol_amount[batch_olcnt] = 0;
        else
            ol_amount[batch_olcnt] = (irand48 () %
999999 + 1);
        randstr (str24[j], 24, 24);

        if (gen) {
            if (cid < 2101) {
                printf (olfp,
"%d %d %d %d %s %d %d %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(tpscvc,
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 = gettimeofday ();
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:
%d - w%d (%d rows)\n ",
bware, eware, nrows);
    begin_time = gettimeofday ();

```

```

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 (stderr);
        }
        else {
            status = OCIStmtExecute(tpscvc, 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 = gettimeofday ();
    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[j] = (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);
}
.....
updateordrordl.sh
.....
#!/bin/sh
$!pcc_sqlplus $!pcc_user_pass <<!
set serveroutput on size 80000
declare
    wid number;
    did number;
    oid number;
begin
    for wid in ${5}..${7} loop
        for did in 1..10 loop
            for oid in 1..2100 loop
                update ord set o_entry_d=sysdate
                where o_w_id = wid and o_d_id =
                did and o_id = oid;
                update ordl set ol_delivery_d = sysdate
                where ol_w_id = wid and ol_d_id =
                did and ol_o_id = oid;
                commit;
            end loop;
        end loop;
        dbms_output.put_line(wid || 'warehouses
        finished');
    end loop;
end;
/

```

```

exit 0
!
.....
views.sql
.....
connect tpcc/tpcc;
set echo on;
create or replace view wh_cust
(w_id, w_tax, c_id, c_d_id, c_w_id, c_discount,
c_last, c_credit)
as select w.w_id, w.w_tax,
        c.c_id, c.c_d_id, c.c_w_id, c.c_discount,
        c.c_last, c.c_credit
    from cust c, ware w
    where w.w_id = c.c_w_id;

create or replace view wh_dist
(w_id, d_id, d_tax, d_next_o_id, w_tax)
as select w.w_id, d.d_id, d.d_tax, d.d_next_o_id,
w.w_tax
    from dist d, ware w
    where w.w_id = d.d_w_id;

create or replace view stock_item
(i_id, s_w_id, i_price, i_name, i_data, s_data,
s_quantity,
s_order_cnt, s_ytd, s_remote_cnt,
s_dist_01, s_dist_02, s_dist_03, s_dist_04,
s_dist_05,
s_dist_06, s_dist_07, s_dist_08, s_dist_09,
s_dist_10)
as
select i.i_id, s.w_id, i.i_price, i.i_name, i.i_data,
s_data, s_quantity,
s_order_cnt, s_ytd, s_remote_cnt,
s_dist_01, s_dist_02, s_dist_03, s_dist_04,
s_dist_05,
s_dist_06, s_dist_07, s_dist_08, s_dist_09,
s_dist_10
    from stok s, item i
    where i.i_id = s.s_i_id;

set echo off;

```

Appendix F: 60 Day Space Calculation

TPM	595,702.31					
Warehouses	48,000					
SEGMENT	TYPE	TSPACE	BLOCKS	FIVE_PCT	DAILY_GROW	TOTAL
CUSTCLUSTER	CLUSTER	CUST	194,259,352	9,712,968	0	203,972,320
DCLUSTER	CLUSTER	WARE	638,384	31,919	0	670,303
HIST	TABLE	HIST	30,896,208	0	1,031,451	31,927,659
ICUST1	INDEX	ICUST1	2,091,583	104,579	0	2,196,162
ICUST2	INDEX	ICUST2	4,587,059	229,353	0	4,816,412
IDIST	INDEX	WARE	2,720	136	0	2,856
IITEM	INDEX	ITEMS	490	25	0	515
IORDR2	INDEX	IORD2	6,465,473	323,274	0	6,788,747
ISTOK	INDEX	ISTK	6,296,643	314,832	0	6,611,475
ITEMCLUSTER	CLUSTER	ITEMS	7,616	381	0	7,997
IWARE	INDEX	WARE	280	14	0	294
NORDCLUSTER	CLUSTER	NORD	11,381,850	569,093	0	11,950,943
ORDR	CLUSTER	ORD	95,257,026	0	18,914,943	114,171,969
STOCKCLUSTER	CLUSTER	STOCKS	392,765,808	19,638,290	0	412,404,098
SYSTEM	SYS	SYSTEM	8,448,000	0	0	8,448,000
WCLUSTER	CLUSTER	WARE	129,472	6,474	0	135,946
Total			753,227,964	30,931,337	19,946,394	804,105,695
Dynamic space		126,153,234				
Static space		658,006,067				
Free space		19,946,394				
Daily growth		19,946,394				
Daily spread		0	Oracle may be configured such that daily spread is 0			
180-day space (blk.)		1,854,789,735				
Block size (bytes)		2,048				
60-day (GB)		3,537.73				
Log block size		512				
Log blocks/tpmC		9.84	Number of log blocks used per New-Order			
8-hour log (GB)		1,341.64				
		Capacity	NumLun		Total GB	
Database Disks	ETERNUS3000 (RAID0)	32	312		49,920.00	
		Capacity	NumLun			
8-Hr Log Disks	ETERNUS3000 (RAID5)	32	48		7,680.00	
		Capacity	Disks			
Interl		18.03	2		36.06	
		Capacity	NumLun			
Total		32	360		57,636.06	

Appendix G: Price Quotes

From: MaryBeth Pierantoni <mary.beth.pierantoni@oracle.com>
 To: skr@jp.fujitsu.com <skr@jp.fujitsu.com>
 Subject: Oracle pricing for 64 processors
 Date: Thu, 18 Sep 2003 13:02

Dear Mr. Kurogi:

To follow is the Oracle pricing information for the Fujitsu/Oracle 64 processor TPC-C benchmark.

Product	Price	Quantity	Extended Price
Oracle Database 10g Enterprise Edition for 3 years, Unlimited Users	\$20,000	64	\$1,280,000
Database Server Support Package	\$2,000	1	\$6,000
Oracle Mandatory E-Business Discount			<\$321,500>
TOTAL			\$964,500

Contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 650-506-2118

VERITAS

Sales Quotation 222052

VERITAS Software

Creation Date 10/14/2003

350 Bix Street
P.O. Box 7011
Mountain View CA 94043-2237
800-327-2232

To:
Name RUBY CERVELLI
Company FUJITSU TECHNOLOGY
Phone (408) 746-7926
Fax
Email RUBY_CERVELLI@FTSI

Account Representative:
Name JULIAN SUAREZ
Company VERITAS Software
Address 400 International Parkway/Heathrow, FL 32746
Phone +1 (303) 383-4021
Fax
Email julian.suarez@veritas.com

Inside Sales Rep:
MICHAEL MAWSON
VERITAS Software
800 International Parkway/Heathrow, FL 32746
(407) 357-5193
(407) 357-7770
mike.mawson@veritas.com

I am pleased to accommodate your request for quotation. If you have any questions related to the products listed or require more information, please contact me at my direct telephone listed above. Thank you for your interest in VERITAS Software and I look forward to serving you in the future.

Ordering Information

(All prices are quoted in US Dollars)

Qty	Part Number	Description	List Price	Est. Price
1	3 v08974F-M00212	Volume Manager,Solaris,v3.5 Tier 4C Extended Support, 1 Yr 24x7	13,845.00	35,304.75
		Comments:Support for line 4.		
2	1 N09665F	Storage Solutions,Solaris,v3.5 Maintenance Pack 1,English,Media Kit	100.00	65.00
3	1 N08836F	Volume Manager,Solaris Administrator's Guide,v3.5,English,Manual	50.00	32.50
4	1 A08974F-M00000	Volume Manager,Solaris,v3.5,License Tier 4C	60,195.00	39,126.75
Subtotal Information		License 39,126.75	Total	74,529.00
		Support 35,304.75		
		Finished Good 97.50		

Comments

Please note that all prices quoted are in US Dollars and are good for 90 days from the date shown above.

Any extension of this sales quotation will only be valid when given in writing by VERITAS Software ("VERITAS").

This quotation is made subject to VERITAS standard Software License Terms and Conditions including FOB VERITAS facilities, freight prepaid and charge, and payment terms of Net 30 days. Changes to such Terms and Conditions must be accepted in writing by VERITAS. Customers shall pay VERITAS a license fee for each software product as quoted. Customer agrees that these fees do not include any maintenance charge, any rights to additional software, enhancements, future updates and upgrades or other software products or deliverables which may be available from VERITAS. Fees for such items will be identified in the then current VERITAS price list. Any orders placed pursuant to this quotation, and any and all use of VERITAS products or services, are subject to the terms and conditions of the applicable VERITAS agreement(s) governing the particular type of order transaction involved.

Consulting

Unless otherwise specified, all VERITAS Consulting Services prices are exclusive of travel, lodging and meal expenses which will be billed at actual cost.

Tax Information

All orders are subject to sales tax. If you are requesting exemption, please provide a copy of your Sales & Use Tax Exemption Certificate.

Address Information

Bill to (For purchase orders only)

Ship to

End User

FUJITSU TECHNOLOGY SOLUTIONS,
250 EAST CARIBBEAN DR.,
SUNNYVALE, CA 94089
United States
Attn: RUBY CERVELLI

FUJITSU TECHNOLOGY SOLUTIONS, INC.,
250 EAST CARIBBEAN DR.,
SUNNYVALE, CA 94089
United States
Attn: RUBY CERVELLI

FUJITSU TECHNOLOGY SOLUTIONS, INC.,
250 EAST CARIBBEAN DR.,
SUNNYVALE, CA 94089
United States
RUBY CERVELLI

Extreme Networks 15501 SUMMIT48I 48PT 10/100SWCH 4PT GBIC W/ BASIC L3 1P/ LAN Switches (Stackable) - Mozilla Firebird

File Edit View Go Bookmarks Tools Help

http://www.dreamretail.com/cgi-win/dream55.exe/~p015000380819

Google LONGMAN Web Dictiona...

Dream Retail . com


[Searching](#) [Ordering](#) [Paying](#) [Shipping](#) [Tracking](#) [Privacy](#) [Security](#) [Terms](#) [Returns](#) [ContactUs](#)

Select search options, sort criteria, sort order, then hit Go!
[CLICK HERE FOR Order Status](#)

Wild Plurals In-Stock "OR" Model View keys **HELP!**

Sort Order?

Mfg Name Starting: **A B C D E F G H I J K L M N O P Q R S T U V W X Y Z #**



Extreme Networks

**15501 SUMMIT48I 48PT 10/100SWCH 4PT GBIC W/ BASIC L3
1P/ LAN Switches (Stackable)**

Normally \$8,995.00 - Now Only **\$8,240.04**

SAVE \$754.96 - over 8% - Our# 015000380819 , Wt=28.9

[Specifications](#)

Must custom order, expect some delay
 E-mail me at

No signup account needed to order, just enter **Quantity** then **Calculate Price**.
 Add **Quantity** to **Order#** **Password**
 Enter **NEW** in **Order#** to force a new Order#

**WE VALUE YOUR OPINION. YOU CAN REVIEW ANY PRODUCT!
NO ONE HAS REVIEWED THIS PRODUCT. WOULD YOU LIKE TO?**
[CLICK HERE to submit your own product review](#)

Want to look at different SWITCHES, MODULES?

[FROM EXTREME NETWORKS](#) [OTHER VENDORS](#)
ON SALE BEST BUYS

Want to see other EXTREME NETWORKS Products?

[THEIR OTHER PRODUCTS](#)

Choose a different COMPUTER HARDWARE subcategory

No image available

Computer Hardware

Computer Software

Office Supplies

Consumer Electronics

Appliances

Automotive Supplies

Hardware, Building Supply

Furniture, Furnishings

Housewares, Pet Supplies

Kitchenware

Lawn, Gardern, Outdoors

Safety and Security

Tools, Moving Equipment

Travel, Entertainment

Health and Beauty

Jewelry and Watches

Photography, Cameras

Sporting Goods

http://www.dreamretail.com/STHARD.html

Extreme Networks 10011 SX GBIC MOD GETH 1PT SC 1000BSX GBIC XCVR MMF Switching LAN/Uplink Modules - Mozilla Firebird

File Edit View Go Bookmarks Tools Help

http://www.dreamretail.com/cgi-win/dream55.exe/?p015000069291

Google LONGMAN Web Dictiona...

Dream Retail . com

[Searching](#) [Ordering](#) [Paying](#) [Shipping](#) [Tracking](#) [Privacy](#) [Security](#) [Terms](#) [Returns](#) [ContactUs](#)

Computer Hardware

Computer Software

Office Supplies

Consumer Electronics

Appliances

Automotive Supplies

Hardware, Building Supply

Furniture, Furnishings

Housewares, Pet Supplies

Kitchenware

Lawn, Gardern, Outdoors

Safety and Security

Tools, Moving Equipment

Travel, Entertainment

Health and Beauty

Jewelry and Watches

Photography, Cameras

Sporting Goods

Select search options, sort criteria, sort order, then hit Go!

Wild Plurals In-Stock "OR" Model View keys [HELP!](#)

Sort Order?

Mfg Name Starting: [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#) #

Extreme Networks

10011 SX GBIC MOD GETH 1PT SC 1000BSX GBIC XCVR
MMF Switching LAN/Uplink Modules

Normally \$495.00 - Now Only **\$461.57**

SAVE \$33.43 - over 6% - Our# 015000069291 , Wt=0.4

172 in stock

No signup account needed to order, just enter **Quantity** then **Calculate Price**.

Add **Quantity** to **Order#** **Password**

Enter **NEW** in **Order#** to force a new Order#:

WE VALUE YOUR OPINION. YOU CAN REVIEW ANY PRODUCT!
NO ONE HAS REVIEWED THIS PRODUCT, WOULD YOU LIKE TO?

[CLICK HERE](#) to submit your own product review

Want to look at different SWITCHES, MODULES?

[FROM EXTREME NETWORKS](#) [OTHER VENDORS](#)

[ON SALE](#) [BEST BUYS](#)

Want to see other EXTREME NETWORKS Products?

[THEIR OTHER PRODUCTS](#)

Choose a different COMPUTER HARDWARE subcategory

No image available

Specifications

http://www.dreamretail.com/STHARD.html

Extreme Networks 97008-48I EXTREMEWORKS 4HR ONSITE 24X7X365 TAC SUMMIT48I Hardware Maintenance Agreement - Mozil...

File Edit View Go Bookmarks Tools Help

http://www.dreamretail.com/cgi-win/dream55.exe/?p015000231673

Google LONGMAN Web Dictiona...

Dream Retail . com

Searching Ordering Paying Shipping Tracking Privacy Security Terms Returns ContactUs

Select search options, sort criteria, sort order, then hit Go!

Wild Plurals In-Stock "OR" Model View keys HELP!

Sort Order?

Mfg Name Starting: **A B C D E F G H I J K L M N O P Q R S T U V W X Y Z #**

Extreme Networks

97008-48I EXTREMEWORKS 4HR ONSITE 24X7X365 TAC
SUMMIT48I Hardware Maintenance Agreement
Your price **\$2,043.85** - Our# 015000231673 , Wt=1.0

Specifications: Must custom order, expect some delay
E-mail me at

No signup account needed to order, just enter **Quantity** then **Calculate Price**.
Add **Quantity** to **Order#** **Password**
Enter **NEW** in **Order#** to force a new Order#.

**WE VALUE YOUR OPINION. YOU CAN REVIEW ANY PRODUCT!
NO ONE HAS REVIEWED THIS PRODUCT, WOULD YOU LIKE TO?**
[CLICK HERE to submit your own product review](#)

Want to look at different SERVICE AGREEMENTS?
[FROM EXTREME NETWORKS](#) [OTHER VENDORS](#)

Want to see other EXTREME NETWORKS Products?
[THEIR OTHER PRODUCTS](#)

[Choose a different COMPUTER HARDWARE subcategory](#)

Thank you for browsing DreamRetail - [CLICK HERE to E-mail suggestions](#).

http://www.dreamretail.com/STEELECHtml

Appendix H: Auditor's attestation letter



Sponsors:	Shin'ichi Kurogi Manager Server Business Development Center Fujitsu Limited Daiichi-Seimei Bldg. 2-7-1 Nishi-Shinjuku, Tokyo, Japan	Harald Eri Manager, OLTP Performance Oracle Server Technologies Oracle Corporation 400 Oracle Parkway M/S 40P11 Redwood Shores, CA 94065
-----------	--	---

October 30, 2003

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

Platform:	Fujitsu PRIMEPOWER 2500 e/s
Operating system:	Solaris 8
Database Manager:	Oracle Database 10g Enterprise Edition
Transaction Manager:	Microsoft COM+

The results were:

CPU's Speed	Memory	Disks	NewOrder Response Time - 90%	tpmC
Server: Fujitsu PRIMEPOWER 2500				
64 x SPARC64 V (1.3 GHz)	512 GB (2 MB L2 cache/cpu)	2 x 36.4GB internal 1800 x 36GB 15Krpm	0.87 Sec.	595,702.31
Fifty-two (52) Clients: Fujitsu PRIMERGY TX200f (Specification for each)				
2 x Intel Xeon (2.4 GHz)	1 GB (512 KB L2 cache/cpu)	2 x 36 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC requirements for Revision 5.1 of the benchmark.

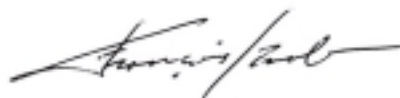
The following verification items were given special attention:

- The transactions were correctly implemented.
- The database records were the proper size.
- The database was properly scaled and populated.
- The ACID properties were met.
- Input data was generated according to the specified percentages.
- The transaction cycle times included the required keying and think times.
- The reported response times were correctly measured.
- At least 90% of all delivery transactions met the 80 Second completion time limit.
- All 90% response times were under the specified maximums.
- The measurement interval was representative of steady state conditions.
- The reported measurement interval was 2 hours.
- Four checkpoints were executed 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 measured configuration included a mix of PRIMERGY C200 (2xPenitum III 1.26GHz, 512KB L2 cache) and F250 (2xXeon 1.8GHz, 512KB L2 cache) clients. For commercial availability reasons, these clients were replaced by PRIMERGY TX200 (2xXeon 2.4BGHz, 512KB L2 cache) in the priced configuration. In my opinion, this change would have no material negative effect on the reported results and complies with the requirements set buy the TPC for such substitutions.

Respectfully Yours,



François Raab, President